



UNIVERSIDAD
DE PIURA

REPOSITORIO INSTITUCIONAL
PIRHUA

ANÁLISIS Y DISEÑO DE UN SOFTWARE PARA CONTROL Y MONITOREO DE VARIABLES DEL PROCESO DE CRIANZA DE LANGOSTINO

Francisco Donayre-Monteza

Piura, junio de 2017

FACULTAD DE INGENIERÍA

Área Departamental de Ingeniería Industrial y de Sistemas

Donayre, F. (2017). *Análisis y diseño de un software para control y monitoreo de variables del proceso de crianza de langostino* (Tesis de licenciatura en Ingeniería Industrial y de Sistemas). Universidad de Piura. Facultad de Ingeniería. Programa Académico de Ingeniería Industrial y de Sistemas. Piura, Perú.



Esta obra está bajo una [licencia Creative Commons Atribución-NoComercial-SinDerivadas 2.5 Perú](#)

[Repositorio institucional PIRHUA – Universidad de Piura](#)

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA



Análisis y diseño de un software para control y monitoreo de variables del proceso de crianza de langostino.

Tesis para optar el Título de Ingeniero Industrial y de Sistemas

Francisco Alonso Donayre Monteza

Asesor: Dr. Ing. Erick Alexander Miñan Ubillús.
Co – Asesor: Ing. Carlos David Zacarías Vélez.

Piura, Junio 2017

A Dios y a María por todo su amor; a mi esposa, Yessica y a nuestros niños: Elenita y Juancito, por su cariño de cada día; a mi mamá Elnita y a mi papá Panchito, por su amorosa dedicación en los años y momentos más importantes de mi vida; a mi amada mamá Vicky y a mi hermana Belén por su apoyo incondicional a todas mis causas; a mi tía Ceci, mi tío Guillermo y mi tío Enrique por tenerme por hijo y llenarme de cariño y valor día a día; a todos aquellos que me han brindado su amistad, su ánimo y su confianza en este camino..

Prólogo

La Universidad de Piura, como parte de su compromiso con el desarrollo de la región y del país, promueve –entre otras actividades- la ejecución proyectos de investigación, desarrollo e innovación en colaboración con entidades públicas y privadas. En este contexto, un equipo de investigadores de la facultad de ingeniería, especialistas en análisis estadístico y matemático, estableció una relación de cooperación con empresas dedicadas a la producción de langostino blanco en agua dulce, con el objetivo de realizar un modelo matemático que permita el control de parámetros que son considerados importantes en la operación.

Dentro del proceso de recojo de información y estudio de la operación de producción del langostino blanco en agua dulce, se encontró que el registro de los datos es llevado en hojas electrónicas de cálculo (Microsoft Excel) y, en otros casos, dicho registro se llevaba de forma manual; estos hechos hacen que los datos se encuentren dispersos, exista incongruencia e inconsistencia de los mismos y, por tanto, el trabajo de generación del modelo matemático, objeto de la cooperación, tomará más tiempo y esfuerzo.

Ante lo mencionado, surgió la visión de proponer un proyecto cuyo producto sea un sistema de información que sirva de punto de partida para empresas langostineras que buscan una solución que les permita automatizar y optimizar los procesos de registro y almacenamiento de datos de la operación de producción del langostino blanco en agua dulce, obteniendo de manera inmediata datos ordenados, consistentes y confiables, los cuales son la base para la generación de información a partir de cartas de control de los parámetros principales del proceso productivo. Además, la arquitectura física y lógica del sistema propuesto permite la interacción con otros sistemas especializados en análisis de datos, esto -de forma puntual- agiliza la actual labor de los profesionales de la Universidad de Piura que tienen por encargo el modelamiento matemático del control de variables de producción.

El proyecto del sistema de información ha sido concebido en dos etapas: análisis y diseño e implementación. El presente trabajo de tesis se encarga de la primera etapa mencionada para la cual, quien suscribe, cuenta con la dirección y asesoría de los especialistas de la Facultad de Ingeniería de la Universidad de Piura y el financiamiento del Programa Nacional de Innovación para la Competitividad y Productividad (Innovate Perú).

Deseo brindar un agradecimiento muy especial a mi asesor, el Ing. Carlos Zacarías Vélez por su amistad, confianza y constante guía, del mismo modo, agradecer al equipo del proyecto de modelamiento, los ingenieros: Susana Vegas Chiyón, Valeria Quevedo Candela, e Ismael Sánchez Rodríguez-Morcillo por su confianza y la oportunidad de colaborar con sus objetivos. Para terminar, agradecer también al Fondo para la Innovación, la Ciencia y la Tecnología (FINCyT) por el impulso y financiamiento que brinda a este y otros proyectos que, en común, buscan el desarrollo de nuestro país habilitándolo a través del desarrollo de la ciencia y la tecnología.

Resumen

Las empresas dedicadas a la producción de langostino blanco (en agua dulce) de la región Piura cuentan con un proceso de producción que tiene por objetivo cosechar el mayor número de langostinos con el peso y tamaño óptimos, requeridos por el mercado. Dentro del desarrollo del proceso mencionado, se registran un conjunto de datos a partir de mediciones de parámetros que son utilizados para controlar la evolución de las larvas de langostino en las diferentes etapas de su cultivo. Estos datos, en la mayoría de casos, están siendo registrados en hojas de cálculo electrónicas, e incluso, de forma manual en cuadernos de notas; esta situación genera que gran cantidad de datos se pierdan, sean inconsistentes y poco fiables, ello dificulta en gran medida el procesamiento de los datos para obtener información valiosa del proceso y, por tanto, impide la generación de mayor conocimiento en la industria.

Esta situación ha sido palpada por los investigadores de la facultad de ingeniería de la Universidad de Piura dentro del marco de un proyecto de colaboración para el análisis de datos generados en la producción del langostino blanco en agua dulce. Ante esta realidad, surge la visión de implementar un sistema informático que registre y almacene de forma ordenada los datos que nacen del proceso en sus diferentes etapas.

Este trabajo de tesis se encarga de realizar el análisis y el diseño del sistema informático para el registro y control de variables de producción del langostino blanco en agua dulce, así, el presente documento contiene un análisis detallado de los requisitos (funcionales y no funcionales) del sistema informático, así como también la definición de las especificaciones técnicas de diseño del mismo y un apartado de conclusiones y recomendaciones que nacen de la experiencia adquirida en la realización de este trabajo.

Como en todo proyecto de ingeniería de software, se realizó el análisis de factibilidad técnica, económica y operativa, el cual, demuestra que la implementación y puesta en marcha del sistema informático propuesto es viable y que puede ser llevado a cabo con dos analistas programadores, dedicando su jornada, de forma exclusiva, por un periodo de 7.31 meses y, finalmente, el costo total estimado para el proyecto es de S/. 72,655.57.

Índice

Capítulo 1	3
Marco conceptual	3
1.1. SISTEMAS DE INFORMACIÓN	3
1.2. ARQUITECTURA CLIENTE/SERVIDOR	3
1.3. SERVIDOR WEB	4
1.4. SERVICIOS WEB	5
1.5. TIPOS DE APLICACIONES	6
1.5.1. Aplicaciones de cliente enriquecido	6
1.5.2. Aplicación Web	7
1.5.3. Aplicación Móvil	7
1.5.3.1. Arquitectura de las aplicaciones móviles	7
1.6. MODELO VISTA CONTROLADOR	9
1.7. LENGUAJE UNIFICADO DE MODELAMIENTO	10
1.7.1. Casos de uso	10
1.7.2. Diagramas de clase	10
1.8. MODELO ENTIDAD RELACIÓN	10
1.9. BASES DE DATOS	10
1.9.1. Administradores de bases de datos	11
1.10. GLOSARIO DE TÉRMINOS	11
Capítulo 2	13
Aspectos generales	13
2.1 PLANTEAMIENTO DE LA HIPÓTESIS	13
2.2 ALCANCE	13
2.2.1 Alcance del proyecto	14
2.2.2 Alcance del producto	14
2.3 FACTIBILIDAD	15
2.3.1 Factibilidad técnica	15
2.3.2 Factibilidad económica	16
2.3.3 Factibilidad operativa	20
2.4 APLICACIONES SIMILARES	20
2.4.1 POND Onshore Aquiculture Modeling	20
2.5 NECESIDAD DE INFORMACIÓN	23
Capítulo 3	25
Análisis del Sistema de Información	25
3.1 DETERMINACIÓN DE REQUISITOS FUNCIONALES	25
3.2 DETERMINACIÓN DE REQUISITOS NO FUNCIONALES	29
3.3 MATRIZ DE TRAZABILIDAD	32
3.4 DIAGRAMAS DE CASOS DE USO	33
3.4.1 Diagrama de casos de uso del administrador	33
3.4.2 Diagrama de casos de uso del analista	42
3.4.3 Diagrama de casos de uso del operario	45
3.5 DIAGRAMA DE CLASES	48
Capítulo 4	51
Diseño del Sistema de Información	51
4.1 INTRODUCCIÓN	51
4.2 ARQUITECTURA	51
4.2.1 Arquitectura lógica	52
4.2.2 Arquitectura física	54
4.3 SOFTWARE Y LENGUAJE DE PROGRAMACIÓN	55
4.4 DISEÑO DE PANTALLAS	55
4.4.1 Diseño de pantallas para la interfaz web	56
4.4.2 Diseño de pantallas para la interfaz móvil	61

4.5	DISEÑO DE BASE DE DATOS.	63
4.5.1	Modelo Entidad Relación.	63
4.5.2	Diccionario de datos.	66
Capítulo 5	73
Conclusiones y recomendaciones	73
5.1	CONCLUSIONES.	73
5.2	RECOMENDACIONES.	74
Bibliografía	75

Introducción

En la región Piura existen empresas dedicadas a la producción, en agua dulce, de langostino blanco (*Litopenaeus Vannamei*). Esta actividad, define un proceso productivo el cual, -entre otras cosas-, cuenta con la definición, monitoreo y control de parámetros útiles para la operación dentro de sus diversas etapas, así, del correcto análisis de los mencionados parámetros, se pueden definir indicadores clave que permitan, además de la optimización de los recursos de la empresa en esta actividad, la generación de conocimiento valioso para el proceso productivo.

La Facultad de Ingeniería de la Universidad de Piura, en colaboración con empresas que producen langostino blanco en agua dulce, encontró que, -en unos casos-, el registro de datos del proceso se hace en hojas de cálculo de Microsoft Excel y, -en otros casos-, dicho registro se hace en uno o varios cuadernos. Esta situación, hace que el inicio de un trabajo de análisis estadístico y matemático de los valores de los parámetros, en busca de optimizar los recursos utilizados, cuente con una complejidad alta debido a la muy probable inconsistencia, incongruencia y redundancia de los datos registrados.

Dado lo anterior, surge la propuesta de iniciar un proyecto de creación de un sistema de información el cual automatice y ordene en gran medida los procesos de toma de datos y el registro de los mismos, logrando así contar con datos confiables y actualizados del proceso productivo, hecho que, -entre otros beneficios-, facilita las tareas de análisis de datos en procura de la mejora continua y adquisición de conocimiento del proceso de producción del langostino en agua dulce.

Este trabajo de tesis se enmarca en la necesidad previamente planteada, con lo cual, pretende definir los documentos de requisitos y las especificaciones técnicas de un sistema de información que, para las empresas que desarrollan esta actividad y llevan su registro de forma manual o en hojas de cálculo electrónicas, sirva como un primer sistema informático de soporte que permita el registro, control y monitoreo de los parámetros del proceso productivo del langostino blanco en agua dulce.

Capítulo 1

Marco conceptual

1.1. Sistemas de Información.¹

Un sistema de información es un conjunto integrado de componentes usados para recolectar, almacenar y procesar datos y, a la vez, entregar información, conocimiento y productos digitales.

Debemos precisar dentro de este concepto que las necesidades de información son diferentes en cada uno de los niveles de la organización. Así, la información puede ser categorizada como: información estratégica, información para la gestión e información operativa.

La figura 1.1 muestra la relación de los sistemas de información con los niveles de la organización.



Figura 1.1 Relación de los sistemas de información con los niveles de la organización.

1.2. Arquitectura cliente/servidor.²

¹ Traducido de los apuntes del curso "Information Systems" (capítulos 1 y 2) del Dr. Kailash Joshi, profesor la facultad de administración de empresas de la Universidad de Missouri, sede de St. Louis. (<http://www.umsl.edu/~joshik/msis480>)

² Texto elaborado en base a la traducción propia del "Chapter 3: Architectural Patterns and Styles", del libro web "Microsoft Architecture Application Guide, 2nd Edition" (<https://msdn.microsoft.com/en-us/library/ff650706.aspx>)

El patrón arquitectónico³ cliente-servidor describe sistemas distribuidos⁴ que implican un sistema cliente y servidor separado y una red de conexión (ver figura 1.2).

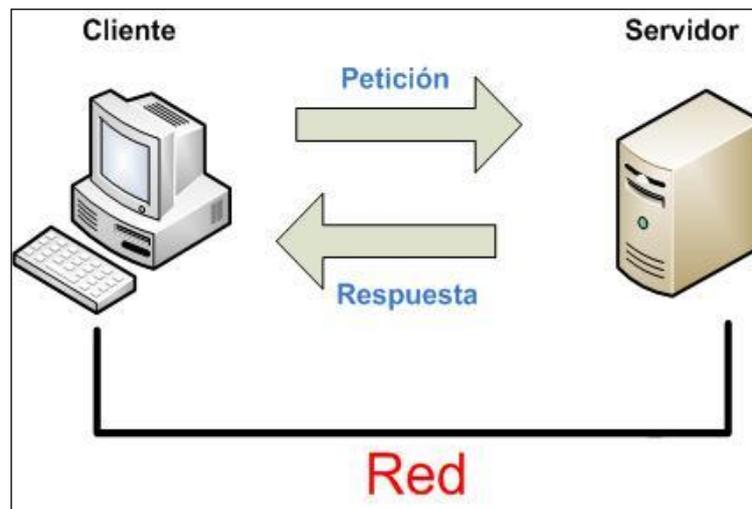


Figura 1.2. Arquitectura Cliente-Servidor⁵

En general, la arquitectura cliente-servidor describe la relación entre un cliente y uno o más servidores, donde dicho cliente inicia uno o más pedidos (posiblemente utilizando una interfaz gráfica), espera por respuestas, las recibe y las procesa. Por el lado del servidor, normalmente autoriza al usuario para luego llevar a cabo el procesamiento requerido y así, generar los resultados. De igual modo, el servidor puede enviar respuestas usando un variado rango de protocolos y formatos de datos para comunicar la información al cliente.

Se debe considerar el uso de la arquitectura cliente-servidor si su aplicación está basada en el servidor y soportará muchos clientes, si está creando aplicaciones basadas en la web para exponerlas a través de un navegador, si está implementando procesos de negocios que serán usados por gente en toda la organización, si está creando servicios para el consumo de otras aplicaciones.

Los fabricantes de hardware llaman servidor a un computador con características especiales que normalmente forma parte de una red y está diseñado para permanecer encendido las 24 horas del día. En adelante, el término servidor será utilizado para referirse al software que brinda algún tipo de servicio, esté éste instalado en cualquier tipo de computador.

1.3. Servidor Web.⁶

El servidor web es un software diseñado para permitir la interacción entre computadores. Suele funcionar permaneciendo a la espera de peticiones, cuando las recibe, responde a

³ Los patrones arquitectónicos expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos, por tanto, se puede afirmar que ayudan a especificar la estructura fundamental de una aplicación.

⁴ Sistema Distribuido, es un Sistema informático compuesto por un conjunto de nodos de procesamiento comunicados y coordinados mediante una red que permite el intercambio de mensajes entre los mismos.

⁵ Imagen extraída del capítulo I, apartado 1.3.1 de la tesis para obtener el grado de ingeniero industrial y de sistemas por la Universidad de Piura, del ingeniero Carlos Zacarías Vélez.

⁶ El contenido de este apartado ha sido tomado de los apuntes del profesor Alex Sánchez Pla para el curso: "Introducción a la biocomputación" (Departamento de Estadística, Universidad de Barcelona, España)

ellas transfiriendo documentos de tipo hipertexto, para ello, implementa el protocolo HTTP (HyperText Transfer Protocol). Vale señalar que, tal como se mencionó en el punto anterior, el término también se emplea para referirse al computador que ejecuta el programa.

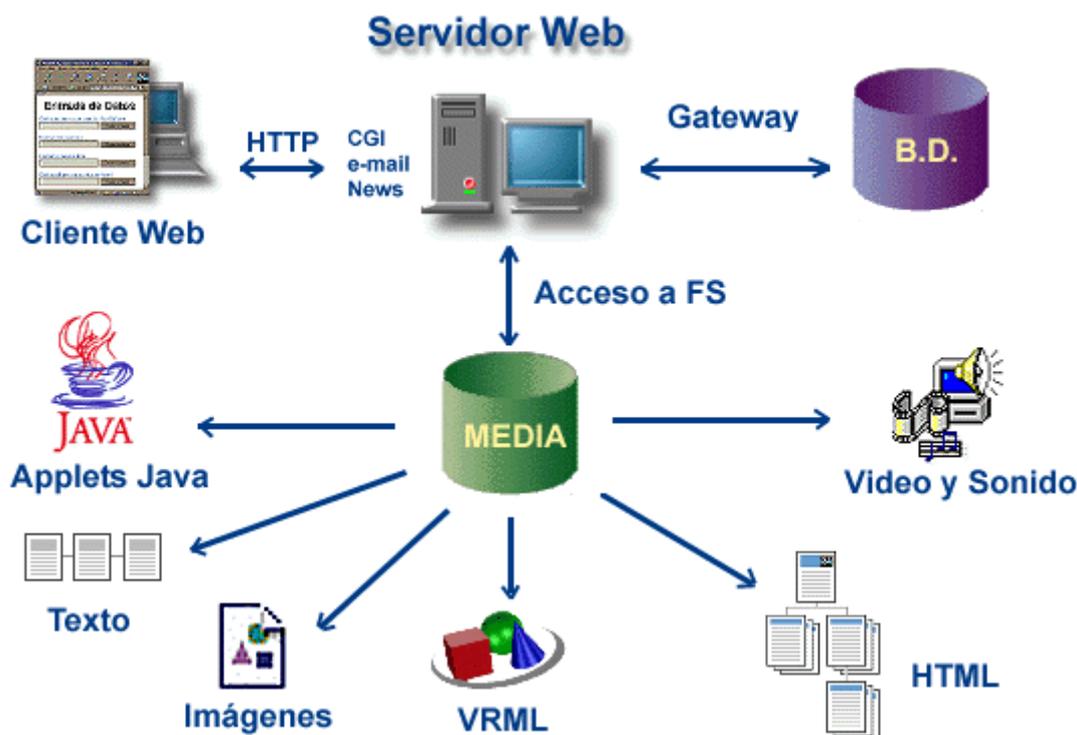


Figura 1.3 Servidor Web

Por su parte, el cliente realiza las peticiones, al recibir una respuesta, interpreta el código HTML que recibe y ejecuta scripts “del lado del cliente” como JavaScript y controla aspectos del formato del documento.

1.4. Servicios Web.⁷

Los servicios web son aplicaciones de software que hacen uso de protocolos y estándares para que se puedan comunicar diferentes aplicaciones. Éstos pueden o no estar desarrolladas en el mismo lenguaje de programación, y pueden ser ejecutadas sobre sistemas operativos diversos.

Los servicios web integran de forma estandarizada aplicaciones web mediante el uso de tecnologías como: XML, SOAP, WSDL y UDDI sobre los protocolos de Internet. XML es usado para describir los datos, SOAP se ocupa para la transferencia de los datos, WSDL se emplea para describir los servicios disponibles y UDDI se ocupa para conocer cuáles son los servicios disponibles.

Las organizaciones OASIS⁸ y W3C⁹ son los responsables de la arquitectura y reglamentación de los servicios web.

⁷ Texto e imagen extraídos del capítulo I, apartado 1.3.4 de la tesis para obtener el grado de ingeniero industrial y de sistemas por la Universidad de Piura, del ingeniero Carlos Zacarías Vélez.

Para mejorar la interoperabilidad entre distintas implementaciones de servicios web se ha creado el organismo WS-I (Web Services Interoperability Organization), encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

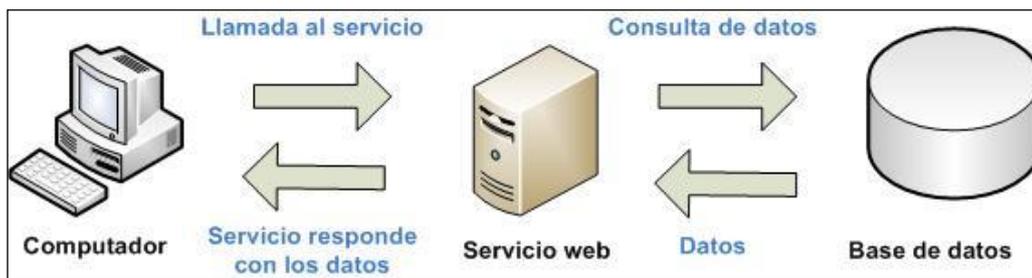


Figura 1.4 Funcionamiento de un servicio web.

1.5. Tipos de aplicaciones.¹⁰

A continuación, presentamos tres de los tipos de aplicaciones más comunes entre los que se podría decidir implementar un proyecto de construcción de un sistema de información:

1.5.1. Aplicaciones de cliente enriquecido.

Las interfaces de usuario de cliente enriquecido pueden brindar una experiencia de alta capacidad de respuesta, interactiva y enriquecida para aplicaciones que deben operar de forma independiente, conectada, ocasionalmente conectada y en escenarios desconectados. Una aplicación de cliente enriquecido estará normalmente estructurada como una aplicación multicapa¹¹.

A su vez, estas aplicaciones podrían usar datos almacenados ya sea en un servidor remoto, o de forma local o como una combinación de ambos. Dichas aplicaciones pueden consumir -también- servicios expuestos por otras aplicaciones, incluyendo servicios web.

Se debe considerar el uso de aplicaciones de cliente enriquecido si:

- Su aplicación debe soportar escenarios desconectados u ocasionalmente conectados.
- Su aplicación será desplegada en computadores personales.
- Su aplicación debe ser altamente interactiva y tener gran capacidad de respuesta.

⁸ OASIS (Organization for the Advancement of Structured Information Standards) es un consorcio internacional sin fines de lucro que orienta el desarrollo, la convergencia y la adopción de los estándares de comercio electrónico y servicios web.

⁹ W3C (WWW Consortium) es un consorcio internacional que produce recomendaciones para la WWW.

¹⁰ Para la elaboración de este apartado se ha tomado como referencia el capítulo 20 y siguientes de "Microsoft Application Architecture Guide 2nd edition" (<https://msdn.microsoft.com/en-us/library/ff650706.aspx>)

¹¹ Las aplicaciones multicapa son aquellas que separan el procesamiento en capas discretas que están distribuidas entre el cliente y el servidor. (Extraído del artículo: Microsoft N-Tier Data Applications Overview - <https://msdn.microsoft.com/en-us/library/bb384398.aspx>)

- La interfaz de usuario de su aplicación debe brindar funcionalidad enriquecida e interactividad pero no requiere las avanzadas capacidades gráficas o de contenido multimedia de una aplicación de internet enriquecida.
- Su aplicación debe utilizar los recursos del computador personal del cliente.

1.5.2. Aplicación Web.

Una aplicación web se basa y extiende a un sistema web para agregar funcionalidad de negocio. Así, del modo más sencillo, una aplicación web es un sistema web que permite a sus usuarios ejecutar lógica de negocio por medio de un navegador web.

Existe una sutil diferencia entre una aplicación web y un sitio web. La aplicación web es un sitio web donde el ingreso de datos, a través de la navegación en el sitio, afecta el estado del negocio. Así, en esencia, una aplicación web usa a un sitio web como interfaz de una aplicación empresarial.

1.5.3. Aplicación Móvil.

Una aplicación móvil es un programa de computador diseñado para ejecutarse en dispositivos móviles tales como teléfonos celulares inteligentes (smartphones) y tabletas.

Muchas aplicaciones de este tipo vienen preinstaladas en los dispositivos móviles antes mencionados, ejemplo de esto son: navegadores de internet, clientes de correo electrónico, calendarios, software de orientación en base a mapas, aplicaciones para compra de elementos multimedia etcétera. Aquellas aplicaciones que no están preinstaladas, usualmente están disponibles a través de plataformas de distribución de aplicaciones las cuales están típicamente operadas por los propietarios de los sistemas operativos móviles.

Debido a la demanda de uso de los dispositivos móviles y a la evolución y actual facilidad del acceso a los kits de desarrollo de aplicaciones móviles, los proyectos de software empresarial, -que usualmente eran aplicados a computadores de escritorio-, incluyen un análisis del impacto del concepto de movilidad en el proyecto, lo cual lleva al desarrollo de una versión para dispositivos móviles en procura del aumento de la calidad y de la productividad del negocio.

Se debe considerar usar aplicaciones móviles si:

- Si los usuarios dependen de dispositivos portátiles.
- La aplicación a desarrollar soporta una interfaz de usuario simple que es adecuada para mostrarse en pantallas pequeñas.
- Si la aplicación debe atender escenarios desconectados u ocasionalmente conectados. En este caso, una aplicación móvil de cliente enriquecido es usualmente la más apropiada.

1.5.3.1. Arquitectura de las aplicaciones móviles.

Una aplicación móvil estará normalmente estructurada como una aplicación multicapa consistente en capas de presentación, de negocios y de datos.

Cuando se desarrolla una aplicación móvil, se debe elegir entre desarrollar clientes ligeros basados en la web o desarrollar clientes enriquecidos.

En una aplicación de cliente enriquecido la capa lógica y la capa de servicios de datos estarán ubicadas en el mismo dispositivo.

Por otro lado, si se está construyendo una aplicación móvil de cliente ligero, todas las capas estarán ubicadas en el servidor.

La figura 1.5 ilustra la arquitectura más usada para aplicaciones de cliente enriquecido con componentes agrupados en áreas de interés.

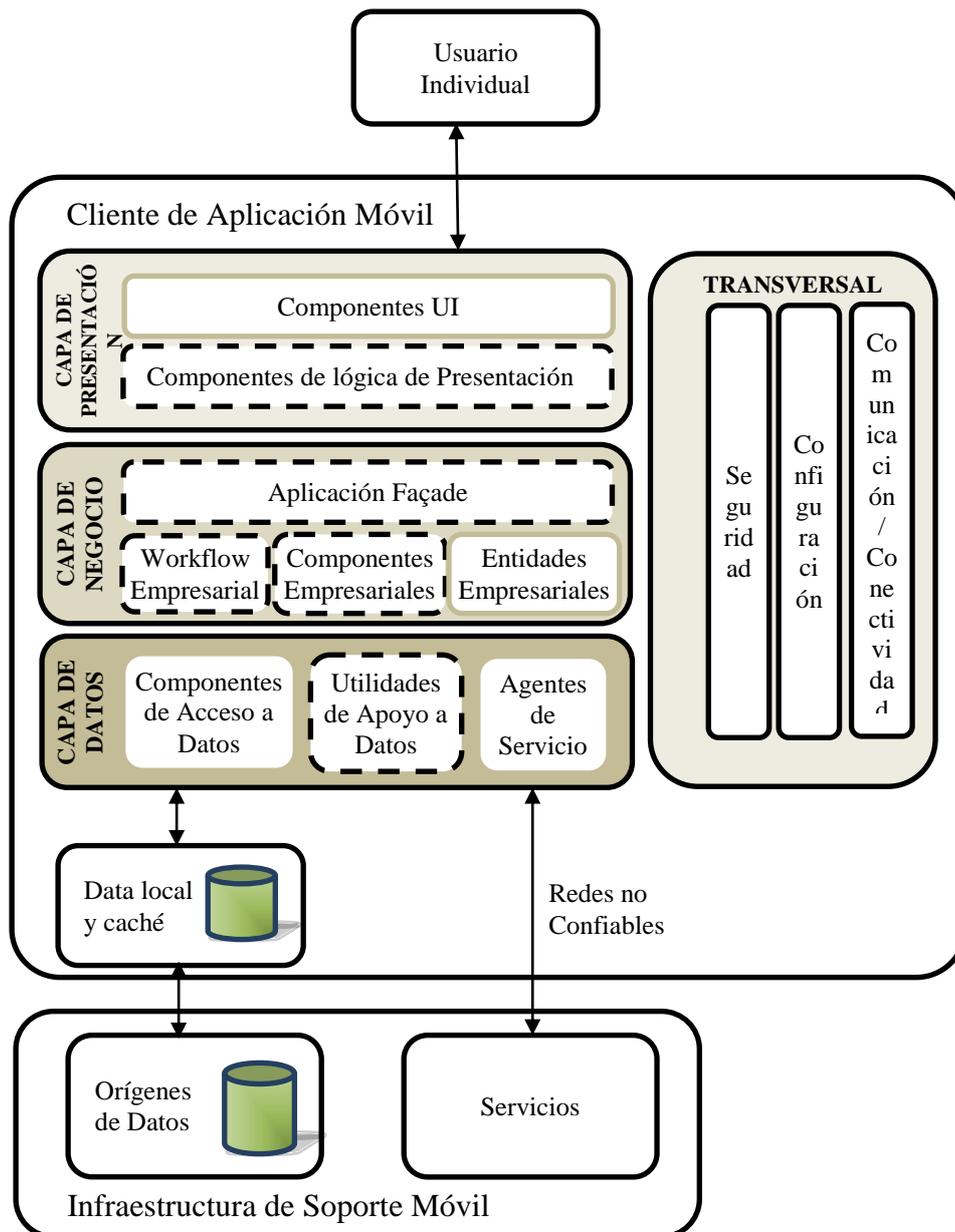


Figura 1.5 Estructura típica de una aplicación móvil.

Pasaremos a continuación a explicar lo modelado en la figura 1.5.

Una aplicación móvil generalmente contiene componentes de interfaz de usuario en la capa de presentación y, posiblemente, podría incluir componentes de lógica de presentación.

La capa de negocios, si existe, usualmente contendrá componentes de lógica de negocio, flujos de trabajo del negocio y componentes de entidad de negocio que son requeridos por la aplicación y, opcionalmente, un façade¹².

La capa de datos a menudo incluirá los accesos a datos y los componentes de agente de servicio.

1.6. Modelo Vista Controlador¹³

El patrón Modelo-Vista-Controlador (MVC) separa el modelamiento del dominio, la presentación y las acciones basadas en las entradas de usuario, en tres clases separadas:

- **Modelo:** gestiona el comportamiento y datos del dominio de la aplicación, responde a requerimientos de información acerca de su propio estado (usualmente desde la Vista) y responde a las instrucciones para cambiar estado (usualmente desde el Controlador)
- **Vista:** gestiona la presentación de la información.
- **Controlador:** interpreta las entradas del ratón y del teclado hechas por el usuario, informando al modelo y/o a la vista el cambio más apropiado.

La figura 1.6 muestra la relación estructural en el modelo.

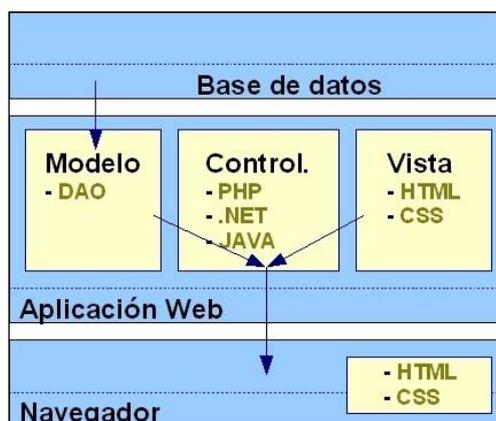


Figura 1.6 Modelo, Vista, Controlador¹⁴

Es importante notar que, tanto la vista como el controlador, dependen del modelo. Sin embargo, el modelo no depende ni de la vista ni del controlador. Esto es uno de los

¹² El patrón facade (o façade) es un patrón de diseño de software comúnmente usado en la programación orientada a objetos. Así, un facade es un objeto que provee una interfaz simplificada para un código extenso tal como lo es una librería de clases.

¹³ El contenido ha sido extraído y traducido del artículo “Microsoft Model-View-Controller”, del portal “Microsoft Developer Network” (<https://msdn.microsoft.com/en-us/library/ff649643.aspx>).

¹⁴ Imagen extraída del capítulo I, apartado 1.6 de la tesis para obtener el grado de ingeniero industrial y de sistemas por la Universidad de Piura, del ingeniero Carlos Zacarías Vélez.

beneficios clave de esta separación, la cual, permite al modelo ser construido y probado sin depender de la presentación visual.

1.7. Lenguaje Unificado de Modelamiento.¹⁵

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

1.7.1. Casos de uso.

El diagrama de casos de uso representa la forma en como un cliente (actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan.

Un diagrama de casos de uso consta de los siguientes elementos: actor, casos de uso y relaciones de uso, herencia y comunicación.

1.7.2. Diagramas de clase.

Un diagrama de clases es una herramienta que explica los conceptos más significativos en un dominio del problema, identificando los atributos y las asociaciones de éstos. Se debe tomar en cuenta que son diagramas de estructura estática donde no se define ninguna operación y representan entidades del mundo real, no componentes del software. Un diagrama de clases está compuesto por los siguientes elementos: clases y relaciones.

1.8. Modelo Entidad Relación.¹⁶

El modelo entidad relación (ER) proporciona una herramienta para representar información del mundo real a nivel conceptual. Permite describir las entidades involucradas en una base de datos, así como las relaciones y restricciones de ellas. Una entidad es un elemento (real o abstracto) con existencia independiente de otros elementos.

1.9. Bases de datos.¹⁷

Una base de datos es un conjunto ordenado de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Con los avances tecnológicos de la informática y de la electrónica, las bases de datos se encuentran de forma digital, ofreciendo un amplio rango de soluciones al problema de almacenar datos.

¹⁵ Extraído de "Tutorial de UML" de Patricio Salinas y Nancy Histchfeld, departamento de Ciencias de la Computación, Universidad de Chile (<http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>).

¹⁶ Tomado de la separata "Modelo Entidad Relación" de Amparo López Gaona, docente del Posgrado en Ciencia e Ingeniería de la Computación, Facultad de Ciencias, Universidad Nacional Autónoma de México.

¹⁷ Extraído del capítulo I, apartado 1.7 de la tesis para obtener el grado de ingeniero industrial y de sistemas por la Universidad de Piura, del ingeniero Carlos Zacarías Vélez.

1.9.1. Administradores de bases de datos

Los administradores de base de datos o también llamados Sistemas de Gestión de Base de Datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los SGBD es el de manejar de manera clara, sencilla, ordenada y segura un conjunto de datos que posteriormente se convertirán en información relevante para un usuario u organización.

1.10. Glosario de términos.

A continuación presentamos un glosario de términos que serán utilizados en este trabajo de tesis y que son de uso común en el proceso de producción de langostinos.

Lote: es un paquete de larvas que una empresa compra para iniciar un proceso de producción de langostino blanco en agua dulce.

Campaña: es el periodo de tiempo en el que, un conjunto de lotes ingresan, de forma sucesiva, a un proceso de producción de langostino. Este periodo termina cuando todos los lotes que han ingresado ya han sido cosechados.

Proceso de producción de langostino blanco: es el conjunto de actividades necesarias para que los lotes de larvas que son adquiridos por la empresa, puedan cosecharse como langostino blanco luego de pasar por unas etapas de cultivo en agua dulce.

Etapas: son los estadios de un proceso de producción de langostino blanco en agua dulce. Para efectos de este trabajo de tesis, se toman 3 etapas: aclimatación de larvas, pre cría y engorde.

Fases: periodo de tiempo en el que una etapa del proceso de producción se está llevando a cabo en una determinada piscina.

Piscina, espacio físico limitado que se utiliza para el cultivo de los langostinos en sus distintas etapas de crecimiento. Las piscinas tienen diferentes características para cada etapa del proceso de producción.

Parámetros: son las variables medibles en cada una de las etapas del proceso de producción de langostino blanco en agua dulce.

Capítulo 2

Aspectos generales

2.1 Planteamiento de la hipótesis.

- Según la información recibida en el transcurso del desarrollo del presente trabajo¹⁸, muchas de las empresas de la Región Piura que tienen actividades de producción del langostino blanco en agua dulce, realizan el registro de sus variables por medio de anotaciones en cuadernos u hojas de cálculo (Microsoft Excel), por ello, existe un alto grado de complejidad al momento de hacer un análisis de los datos de modo que permita llevar a cabo un control y monitoreo de las variables registradas; a esto se une el hecho de la alta probabilidad de pérdida de los datos, de contar con datos inconsistentes y redundantes que limiten la precisión de los trabajos de análisis estadístico.
- Dado el contexto explicado, ejecutar un proyecto de desarrollo de un sistema de información, permitirá a los interesados contar con datos ordenados, integrados y libres de redundancias e inconsistencias. Además, será la plataforma idónea para futuros proyectos y trabajos de investigación de la operación que permitan generar conocimiento valioso para la misma.
- El proyecto de desarrollo del sistema de información podría ser llevado a cabo, -principalmente-, por las empresas dedicadas a esta actividad dado que cuentan con acceso a recursos económicos o financiamiento, e incluso algunas poseen la plataforma tecnológica, el personal calificado y el capital necesario para culminar con éxito el mencionado proyecto.

2.2 Alcance.

En cuanto al alcance geográfico del proyecto este se contextualiza en el entorno físico de realización tanto de las operaciones de producción de langostino blanco, como de las oficinas administrativas que acogen al personal involucrado en la actividad de producción.

¹⁸ Hasta el momento de presentación de este trabajo de tesis, no se ha verificado la existencia de estadísticas que incluyan directamente a empresas langostineras de la región Piura, tanto en entidades públicas de orden regional y nacional, como en instituciones privadas.

El proyecto que se plantea para la creación de un sistema de información para el control y monitoreo de las variables de producción de langostino blanco.

El presente trabajo de tesis se refiere íntegramente al análisis y diseño del sistema de información.

2.2.1 Alcance del proyecto.

Definiremos a continuación el alcance del proyecto:

- Llevar a cabo, -por medio de reuniones-, el levantamiento de información sobre el proceso productivo del langostino blanco en la Región Piura.
- Definir los requerimientos funcionales y no funcionales del sistema.
- Redactar el marco conceptual necesario para el entendimiento de la terminología técnica utilizada en el presente trabajo.
- Elaborar el estudio de factibilidad del proyecto para determinar, -desde el punto de vista técnico, operativo y económico-, la usabilidad del aplicativo a desarrollar así como también la posibilidad de implementación del proyecto.
- Llevar a cabo la investigación necesaria para determinar si en el mercado informático existen aplicaciones que gestionen el control y monitoreo de variables del proceso de producción del langostino blanco en agua dulce.
- Elaborar la documentación del análisis y diseño del sistema de información. Este entregable contará con los diagramas de caso de uso, los diagramas de clases, el modelo de datos y su diccionario, el diseño de pantallas principales.

2.2.2 Alcance del producto.

Se elaborará un documento conteniendo:

- Análisis de factibilidad.
- Modelamiento conceptual del sistema de información conteniendo los diagramas de casos de uso y el diagrama de clases.
- Modelamiento de los datos de forma detallada.
- Diseño de pantallas principales del sistema de información.
- Descripción de la arquitectura lógica y física del sistema de información a implementar.

2.3 Factibilidad.

El análisis de factibilidad que a continuación se presenta, ha sido realizado tomando en cuenta el proyecto de sistema de información a implementar, logrando así facilitar el desarrollo de ambos sub proyectos descritos previamente en este capítulo.

Este análisis incluye el detalle de factibilidad técnica, económica y operativa, los cuales serán descritos a continuación.

2.3.1 Factibilidad técnica.

Vamos a considerar en este punto el hardware y software necesario para la implementación del sistema de información

Para el Analista – Programador

Hardware

Tabla 2.1 Especificaciones técnicas de hardware recomendadas para el analista – programador

Equipo	Características
Computador de escritorio	Procesador Intel Core I5 4460 de 3.4 GHz Quad Core
	Memoria RAM de 4GB DDR3
	Disco Duro 500 GB
	Lector Grabador CD/DVD

Software

Tipo	Software recomendado
Sistema operativo	Ubuntu Linux 15 (o superior) / Microsoft Windows 7 profesional (o superior)
Ofimática	Microsoft Office 2007 o superior
Servidor de base de datos	MySQL 5.7 o superior
IDÉ de base de datos	MySQL Workbench 6.3 o superior
Lenguaje de programación	PHP 5.6.18 o superior
Servidor Web	Apache 2.2.31 o superior

Tabla 2.2 Especificaciones técnicas de software recomendadas para el analista – programador

Cabe señalar que, en cuanto al software, este dependerá principalmente del entorno propio de la empresa que ejecutará el proyecto de sistema de información, sin embargo, para efectos de este estudio, propondremos el software elemental para la programación del proyecto.

Del servidor**Hardware**

Equipo	Características
Computador de escritorio	Procesador Intel Core I5 4460 de 3.4 GHz Quad Core
	Memoria RAM de 4GB DDR3
	Disco Duro 500 GB
	Lector Grabador CD/DVD

Tabla 2.3 Especificaciones técnicas de hardware recomendadas para el servidor

Software

Tipo	Software recomendado
Sistema operativo	Ubuntu Linux 15 (o superior)
Servidor de base de datos	MySQL 5.7 o superior
IDE de base de datos	MySQL Workbench 6.3 o superior
Lenguaje de programación	PHP 5.6.18 o superior
Servidor Web	Apache 2.2.31 o superior

Tabla 2.4 Especificaciones técnicas de software recomendadas para el servidor

Dadas las especificaciones mínimas indicadas previamente, se nota que el software requerido es de libre acceso y que, -en sus versiones básicas-, la mayoría de ellos no implica pago por licenciamiento. En el caso del hardware, los requerimientos indicados para los computadores son accesibles de forma rápida y directa en el mercado local.

Por lo antes expuesto podemos considerar que esta ejecución, -a nivel técnico-, es factible para el proyecto.

2.3.2 Factibilidad económica.

Dentro de este apartado, tomaremos en consideración los costos de hardware y software descritos en la factibilidad técnica y, dado que se trata de un proyecto de desarrollo de software, utilizaremos el modelo básico de costos constructivos (COCOMO¹⁹) por ser una herramienta muy utilizada para estos efectos.

Costos de hardware y software

Mostramos el resumen de los costos unitarios de hardware y de software según lo indicado en la factibilidad técnica.

¹⁹ COCOMO® es un modelo que permite estimar costo, esfuerzo y programación cuando se planifica una nueva actividad de desarrollo de software. Tiene tres formas: básica, intermedia y detallada. Cada una ofrece mayor detalle y precisión cuando más avanzado se está en la planificación del proyecto. (Centro para la Ingeniería de Sistemas y Software de la Universidad del Sur de California)

Ítem	Unidad
Computadores de escritorio	1,500.00
Licencia de Ubuntu Linux 15	0.00
Licencia de Microsoft Windows 7 (OEM ²⁰)	0.00
Licencia de Microsoft Office 2016 Home	370.00
Licencia de MySQL 5.7	0.00
Licencia de MySQL Workbench 6.3	0.00
Licencia de PHP 5.6.18	0.00
Licencia de Apache 2.2.31	0.00

Tabla 2.5 Tabla de costos de hardware y software según lo indicado en la factibilidad técnica.

Obsérvese que, en cuanto al software, el costo de licenciamiento es nulo, esto debido a la utilización de software que se distribuye gratuitamente a través de internet.

COCOMO

Usaremos las siguientes ecuaciones del modelo COCOMO:

- Miles de líneas de código: $MF = C * RS$.
- Esfuerzo: $E = a(MF)^b$, personas necesarias para llevar a cabo el proyecto su unidad es persona-mes.
- Tiempo de desarrollo: $Tdev = c(E)^d$, se estima en meses.
- Personal necesario: $P = E/Tdev$, personas necesarias para el desarrollo del proyecto, su unidad son personas.
- Costo total: $C = E * CHM$, costo total del proyecto.

Donde

- **C** es una constante la cual, por experiencia transmitida, se ha establecido el valor de 400.
- **RS** son las responsabilidades del sistema.
- **a, b, c y d** son constantes con valores definidos en una tabla, según cada submodelo.
- **MF** es la cantidad de líneas de código, en miles.

²⁰ La Licencia para ensambladores de equipos OEM de Microsoft está concebida para facilitar a los ensambladores la adquisición y distribución de software Microsoft original dentro de los equipos que ensamblan y comercializan. Esta licencia es un contrato entre usted, el ensamblador de equipos y Microsoft. (Microsoft – Licencia de Ensambladores de Equipos)

- **CHM** es el costo por hombre mes. (Lo que se gasta en desarrollo de software cuando trabaja un mes en un proyecto, por seguridad asumimos el doble del salario promedio de un analista - programador).

COCOMO tiene tres modelos de representar el tipo de proyecto y pueden ser:

- Modo orgánico: un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar.
- Modo rígido: el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.
- Modo semilibre: corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

Asumiendo un proyecto de tipo orgánico: pocas personas y alta confianza entre ellas, se intentará realizar una aproximación rápida del esfuerzo escogiendo el modelo básico, el cual usa la tabla que se muestra a continuación.

Modelo básico				
Constante \ Modo	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Rígido	3.60	1.20	2.50	0.32
Semilibre	3.00	1.12	2.50	0.35

Tabla 2.6 Modelo básico (COCOMO)

A continuación procederemos a aplicar las fórmulas del modelo COCOMO tomando en cuenta que usaremos el modelo básico y de tipo orgánico.

- **RS: Responsabilidades del Sistema**
 - Gestión de unidades de medida.
 - Gestión de proveedores.
 - Gestión de usuarios y su acceso al sistema informático.
 - Gestión de sectores de piscinas.
 - Gestión de campañas.
 - Gestión de lotes.
 - Gestión de piscinas.
 - Gestión de fases.
 - Asignación de piscinas a fases.
 - Asignación de límites a fases.
 - Generación de alertas automáticas ante límites trasgredidos.
 - Carga de piscinas (ingresar lotes a piscinas).

- Gestión de parámetros.
- Gestión de valores de parámetros por fases.
- Análisis histórico de datos de parámetros.
- Generación de gráficas de control.

Nota: el término “Gestión” implica la inserción, edición, borrado, búsqueda y muestra de elementos.

- Miles de líneas de código: $MF = C * RS$.

$$\text{Entonces: } MF = 400 * 16 = 6400$$

- Esfuerzo: $E = a(MF)^b$

$$\text{Entonces: } E = 2.40(6.4)^{1.05} = 16.85 \text{ personas-mes.}$$

- Tiempo de desarrollo: $Tdev = c(E)^d$

$$\text{Entonces } Tdev = 2.5(16.85)^{0.38} = 7.31 \text{ meses}$$

- Personal necesario: $P = E/Tdev$

$$\text{Entonces: } P = 16.85 / 7.31 = 2.30 \text{ (2 personas)}$$

- Costo total: $C = E * CHM$,

Hallamos CHM (doble del costo promedio de un analista – programador)

$$CHM = 2(2000) = 4000$$

$$\text{Entonces: } C = 16.85 * 2 * 2000 = \mathbf{67,415.57 \text{ nuevos soles.}}$$

Contando ya con los costos, procedemos a hacer una tabla resumen de los mismos:

Objeto de costo	Ítem	Costo
1 Servidor	Hardware	1,500.00
	Software	0.00
2 Analistas - programadores	Hardware	3,000.00
	Software	740.00
	Salario	67,415.57
TOTAL		72,655.57

Tabla 2.7 Consolidación detallada de costos

Si la empresa que decida desarrollar este sistema de información cuenta con el hardware indicado y con las licencias de software especificadas (estén, o no, sujetas a pago), del costo total estimado se retirarían los costos de hardware y software con el objetivo de

presupuestar únicamente el salario de los analistas programadores por el tiempo estimado del proyecto de desarrollo.

2.3.3 Factibilidad operativa.

El sistema de información que esta tesis analiza, diseña y plantea desarrollar, será una herramienta que permitirá a las empresas productoras del langostino blanco, la automatización de la toma de datos y, con ello, el ordenamiento congruente y consistente de los mismos para la generación oportuna de información confiable a través de cartas de control.

Este software será una herramienta que podrá ser usada como un primer modelo para aquellas empresas que en la actualidad, cuentan con procesos manuales o mínimamente informatizados (hojas de cálculo) puesto que, a nivel de usuario, la tecnología a implementar es de actual cotidianeidad: aplicaciones web y aplicaciones móviles, lo que hace que los procesos de capacitación preliminar en la operación del software, junto con los procesos de aprendizaje y adopción del sistema, sean más rápidos, tanto por la sencillez en el uso del entorno gráfico del software como por el conocimiento y experiencia que los usuarios ya tienen con la terminología y procesos de toma de datos de la producción.

Del mismo modo, los especialistas encargados de analizar los datos almacenados, -además de tener las gráficas de control y la visualización de los datos de la operación-, contarán con una base de datos segura, ordenada, confiable y accesible para el desarrollo de sus operaciones y procedimientos de orden matemático y estadístico así como también, para la interacción con software especializado en la materia de su especialidad. Esto permitirá generar información útil que podrá ser replicada de forma segura en el sistema a desarrollar.

2.4 Aplicaciones similares.

2.4.1 POND Onshore Aquiculture Modeling.²¹

POND es un programa para el sistema operativo de Microsoft Windows utilizado, fabricado por Longline Enviroment (Reino Unido) y hecho para administrar granjas de langostinos. POND calcula el crecimiento y la producción de langostino a través del tiempo, tomando en cuenta el diseño de la granja, costo de la alimentación, tecnología de la granja (especies, densidad de siembra, periodo de crecimiento) y las características del estanque (temperatura del agua, salinidad, clorofila y oxígeno disuelto). POND analiza todos los datos ingresados por el usuario y luego muestra reportes de biomasa cosechable, análisis de producción, efectos de la calidad del agua, análisis de balance de masa entre otros. Las especies que cubre POND son:

- Langostino blanco del Pacífico (*Penaeus vannamei*)
- Langostino de la India (*Penaeus indicus*)
- Gamba tigre gigante (*Penaeus monodon*)

²¹ Lo descrito en este apartado del presente trabajo, está basada en la información brindada por Longline Enviroment Ltd en un documento promocional su producto POND, publicado en la siguiente página web: http://www.longline.co.uk/site/aboutus/publications/pond_brochure.pdf

Cabe señalar en este apartado que, pese a diversos intentos, no ha sido posible obtener una copia de este software para poder revisar y comprobar tanto la veracidad de la información, como las funcionalidades que podrían ser similares a las propuestas en este trabajo de tesis.

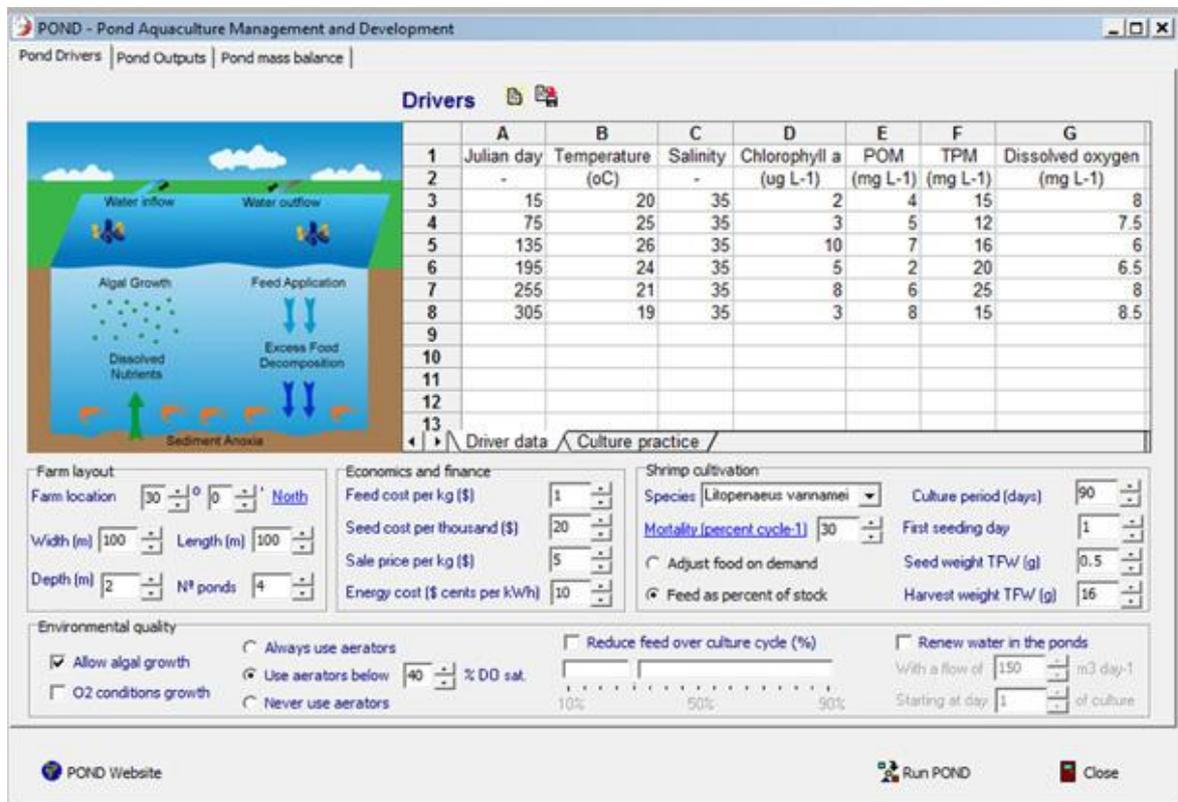


Figura 2.1 Pantalla de controladores de POND

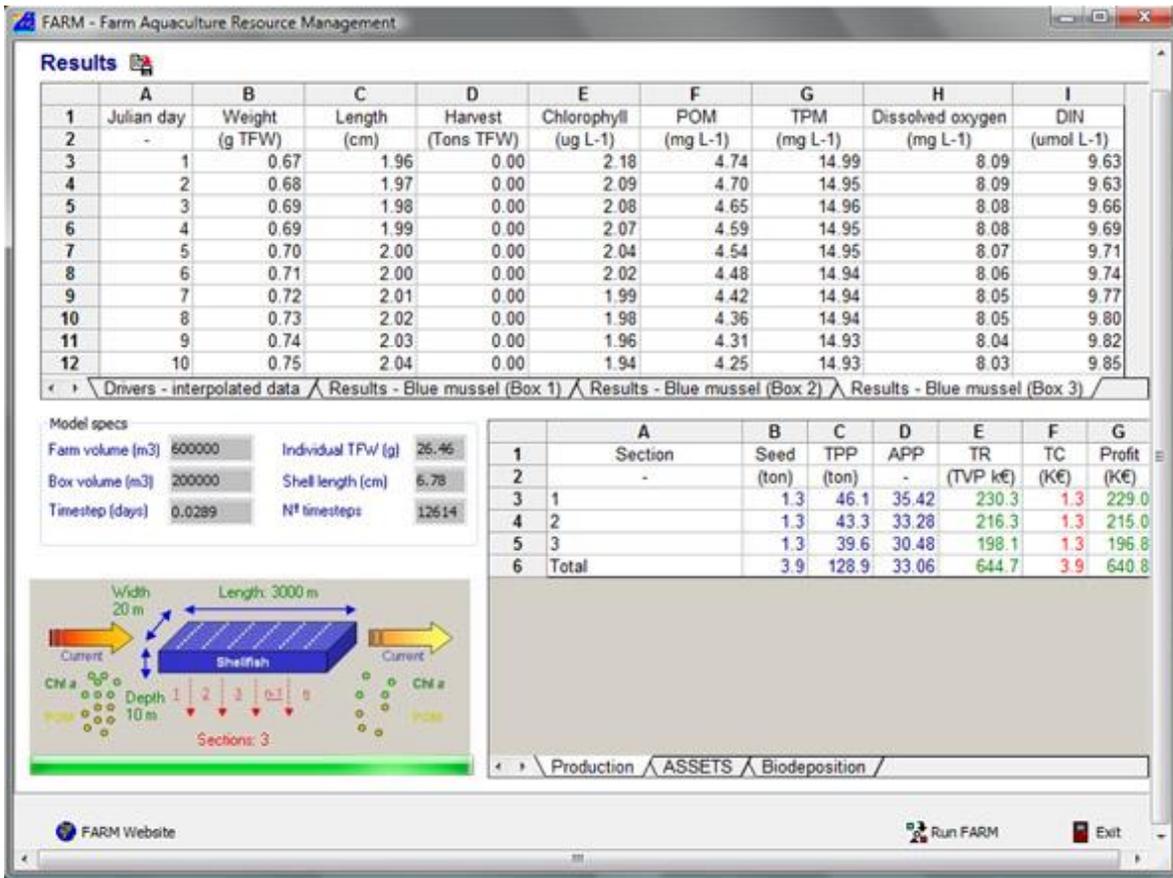


Figura 2.2 Pantalla de datos de salida de POND

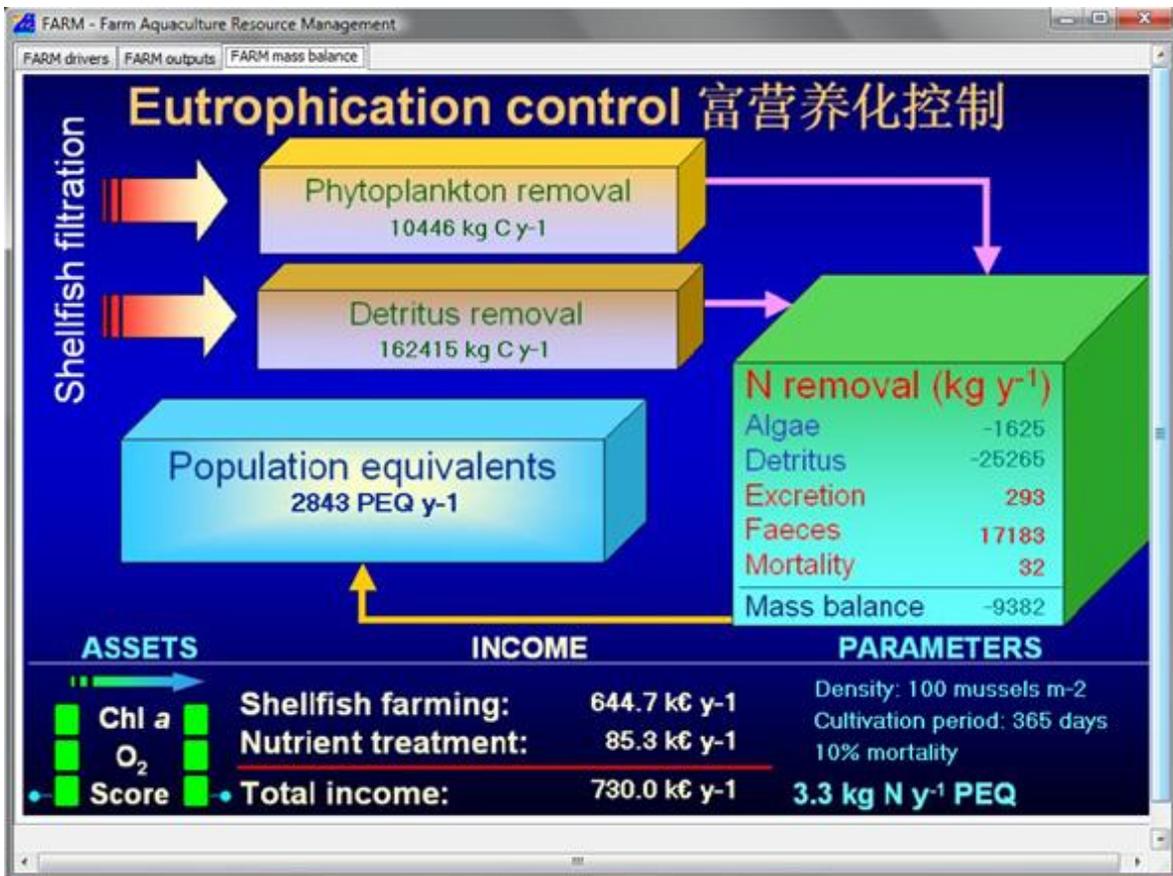


Figura 2.3 Pantalla que muestra el balance de masa calculado por POND.

2.5 Necesidad de información.

Para el desarrollo de esta tesis, contamos con la información y los requerimientos que brindan los especialistas en gestión y análisis de datos (matemáticos y estadísticos) de la facultad de ingeniería de la Universidad de Piura, quienes inicialmente han concebido la visión de esta necesidad y, por tanto, son los principales interesados en la sistematización de los datos de la actividad productiva para el estudio matemático y estadístico de las variables del proceso en afán de optimizar los recursos existentes, materia de este estudio; además, dichos profesionales cuentan con la experiencia y conocimientos necesarios en materia de procesos de producción dada su especialidad y el desarrollo de diversos proyectos de colaboración interinstitucional con diversas empresas del medio..

Para determinar la necesidad de información del proyecto, se llevarán a cabo las siguientes actividades: realización de entrevistas personales y recabo de documentación.

Realización de entrevistas personales.

Esta actividad consistió en realizar reuniones con el personal especializado en análisis y gestión de datos (matemáticos y estadísticos) de la Universidad de Piura, quienes cuentan con el conocimiento del proceso productivo, con el fin de tener detalles sobre lo siguiente:

- Detalle de los procesos y sus procedimientos de trabajo a nivel de la operación y de la gestión de la misma.
- Información interna y externa, -distinta o ajena al proceso en si-, que influya directamente en la actividad productiva.

Recabo de documentación.

Esta actividad consiste en obtener copias físicas o digitales de documentación utilizada en la actividad productiva. Aquí se encontró lo siguiente:

- Formatos y formularios que contengan datos del proceso productivo.
- Reportes que se emiten tomando en consideración datos sobre el proceso.

Normativa nacional e internacional que regula el proceso, lo enmarca o da recomendaciones de las mejores prácticas en el desarrollo del proceso de producción.

Capítulo 3

Análisis del Sistema de Información

3.1 Determinación de requisitos funcionales.

A continuación, presentamos el listado de los requisitos funcionales recogidos en la etapa de análisis del sistema de información propuesto.

Identificador	Título
RF-0001	Gestionar unidades de medida.
RF-0002	Gestionar parámetros.
RF-0003	Gestionar usuarios.
RF-0004	Recuperar clave.
RF-0005	Gestionar sectores de piscinas.
RF-0006	Gestionar piscinas.
RF-0007	Gestión de proveedores.
RF-0008	Gestión de campañas.
RF-0009	Gestión de lotes.
RF-0010	Gestión de fases.
RF-0011	Asignar piscinas a fases.
RF-0012	Cargar piscinas.
RF-0013	Registrar límites a una fase.
RF-0014	Registrar valores medidos en una fase.
RF-0015	Generar gráficas de control por parámetro para todas las piscinas.
RF-0016	Generar gráfica de control por parámetro en una piscina.
RF-0017	Generar gráficas de control por parámetro de salida.
RF-0018	Generación de alertas por parámetro fuera de límite.
RF-0019	Guardar errores de sistema en un archivo de texto.

Tabla 3.1 Listado de requisitos funcionales.

Presentamos a continuación el detalle de los requisitos funcionales listados en la tabla 3.1.

IDENTIFICADOR	RF-0001	TÍTULO	Gestionar unidades de medida
DESCRIPCIÓN	El sistema debe permitir agregar y modificar las unidades de		

	medida de valores de parámetros.		
REQUISITOS DE INFORMACIÓN	Los datos para las unidades serán: nombre y abreviatura		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0002	TÍTULO	Gestionar parámetros
DESCRIPCIÓN	El sistema debe permitir crear y modificar los parámetros usados en el proceso de producción del langostino blanco.		
REQUISITOS DE INFORMACIÓN	Los datos de parámetros que el sistema debe administrar son: nombre, abreviatura, unidad de medida, es de entrada/es de salida.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0003	TÍTULO	Gestionar usuarios
DESCRIPCIÓN	El sistema debe permitir crear, modificar y habilitar/deshabilitar usuarios para el acceso al software.		
REQUISITOS DE INFORMACIÓN	Los datos de los usuarios son: nombre, apellidos, correo electrónico, nombre de inicio de sesión, clave de acceso encriptada, activo.		
NECESIDAD	Esencial	PRIORIDAD	Media

IDENTIFICADOR	RF-0004	TÍTULO	Recuperar clave
DESCRIPCIÓN	El sistema debe permitir al administrador poder hacer el cambio de clave del usuario que lo solicite.		
REQUISITOS DE INFORMACIÓN	Los datos necesarios para esta funcionalidad son: código del usuario solicitante, nueva clave de acceso al sistema.		
NECESIDAD	Deseable	PRIORIDAD	Baja

IDENTIFICADOR	RF-0005	TÍTULO	Gestionar sectores de piscinas
DESCRIPCIÓN	El sistema debe permitir agregar y modificar los sectores de piscinas que pueden existir en la planta de producción.		
REQUISITOS DE INFORMACIÓN	Los datos requeridos son: código, estado (activo o no).		
NECESIDAD	Deseable	PRIORIDAD	Baja

IDENTIFICADOR	RF-0006	TÍTULO	Gestionar piscinas
DESCRIPCIÓN	El sistema debe permitir agregar y modificar los datos generales de una piscina que es utilizada en el proceso de producción.		
REQUISITOS DE INFORMACIÓN	Los datos requeridos son: código, sector, nombre de la piscina, valor de la superficie de la piscina, habilitado/deshabilitado, descripción.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0007	TÍTULO	Gestión de proveedores
DESCRIPCIÓN	El sistema debe permitir el ingreso y edición de datos generales de los proveedores de lotes de larvas para el proceso.		
REQUISITOS DE INFORMACIÓN	Los datos a almacenar son: nombre, activo.		
NECESIDAD	Deseable	PRIORIDAD	Media

IDENTIFICADOR	RF-0008	TÍTULO	Gestión de campañas
DESCRIPCIÓN	El sistema debe permitir el ingreso de los datos de las campañas de producción que la empresa puede tener.		
REQUISITOS DE INFORMACIÓN	Los datos que se necesitan son: código, descripción, fecha de inicio, fecha de fin.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0009	TÍTULO	Gestión de lotes
DESCRIPCIÓN	El sistema debe permitir el ingreso de datos de los lotes de larvas que son comprados por la empresa para empezar un nuevo proceso de producción.		
REQUISITOS DE INFORMACIÓN	Los datos a almacenar son: código de la campaña, código del proveedor, código del lote, nombre, cantidad, peso, descripción, fecha de llegada real		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0010	TÍTULO	Gestión de fases.
DESCRIPCIÓN	El sistema debe permitir la configuración de las fases de un proceso de crianza de langostinos en una campaña determinada.		
REQUISITOS DE INFORMACIÓN	Datos necesarios son: campaña, etapa, fecha de inicio, fecha de fin.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0011	TÍTULO	Asignar piscinas a fases.
DESCRIPCIÓN	Al momento de configurar una fase, el sistema debe permitir la asignación de las piscinas que serán usadas en la etapa que es motivo del inicio de la fase del proceso.		
REQUISITOS DE INFORMACIÓN	Los datos requeridos son: fase, piscina, fecha de asignación de la piscina a la fase.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0012	TÍTULO	Cargar piscinas.
DESCRIPCIÓN	A lo largo del ciclo productivo de una fase, el sistema debe permitir		

	el registro detallado de lotes que son ingresados a una determinada piscina.		
REQUISITOS DE INFORMACIÓN	Los datos requeridos son: fase, piscina, lote, fecha de ingreso del lote a la piscina, cantidad que ingresa.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0013	TÍTULO	Registrar límites a una fase.
DESCRIPCIÓN	Al momento de configurar una fase, el sistema debe permitir el registro de los límites de cada parámetro que será medido en dicha fase.		
REQUISITOS DE INFORMACIÓN	Los datos requeridos son: fase, número de semana, parámetro, límite superior, límite inferior, valor medio.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0014	TÍTULO	Registrar valores medidos en una fase.
DESCRIPCIÓN	El sistema debe permitir el ingreso de valores (desde dispositivos móviles y computadores de escritorio) para los parámetros medibles en cada fase y en diferentes momentos de la vigencia de la misma.		
REQUISITOS DE INFORMACIÓN	Los datos requeridos para esta funcionalidad son: código de la fase, código del parámetro, código del proveedor de la medida (puede ser una tercera empresa), valor obtenido, comentarios, fecha y hora de toma de la medida		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0015	TÍTULO	Generar gráficas de control por parámetro para todas las piscinas
DESCRIPCIÓN	El sistema debe permitir la generación de una gráfica de control por elección de un parámetro medible en todas las piscinas dentro de una etapa y en una campaña.		
REQUISITOS DE INFORMACIÓN	Los datos necesarios para esta funcionalidad son: código del parámetro, código de la etapa, códigos de las piscinas, valores del parámetro dentro de las fases que corresponden a la etapa y a la campaña seleccionadas.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0016	TÍTULO	Generar gráfica de control por parámetro en una piscina
DESCRIPCIÓN	El sistema debe permitir la generación de una gráfica de control por elección de un parámetro medible en una piscina dentro de una etapa para una campaña.		
REQUISITOS DE INFORMACIÓN	Los datos necesarios son: código del parámetro, código de la etapa, código de la piscina y el conjunto de valores del parámetro dentro de las fases que corresponden a la etapa y a la campaña seleccionada.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0017	TÍTULO	Generar gráficas de control por parámetro de salida.
DESCRIPCIÓN	El sistema debe permitir la creación de gráficas de control para un parámetro de salida a lo largo de distintas etapas por las que pasa un lote en una campaña.		
REQUISITOS DE INFORMACIÓN	Esta funcionalidad requiere los siguientes datos: código del parámetro de salida, código de la etapa dentro de una campaña, valores registrados para el parámetro en los procesos seleccionados.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RF-0018	TÍTULO	Generación de alertas por parámetro fuera de límite.
DESCRIPCIÓN	El sistema debe permitir la generación de una alerta que indique que, en cierta fase, un parámetro está fuera del límite proyectado en una semana específica.		
REQUISITOS DE INFORMACIÓN	Los datos necesarios son: código del valor ingresado, código del límite trasgredido, visualizado o no.		
NECESIDAD	Esencial	PRIORIDAD	Media

IDENTIFICADOR	RF-0019	TÍTULO	Guardar errores de sistema en un archivo de texto
DESCRIPCIÓN	El sistema deberá guardar en un archivo de texto todos los errores de sistema, es decir, aquellos que no involucran a la operación en sí.		
REQUISITOS DE INFORMACIÓN	Los datos a incluir en cada evento son: fecha, hora, descripción del error, origen del error.		
NECESIDAD	Esencial	PRIORIDAD	Baja

3.2 Determinación de requisitos no funcionales.

Ahora, pasamos a definir los requisitos no funcionales definidos en el presente análisis.

Identificador	Título
RNF-0001	Tiempo de transacciones en operaciones comunes del sistema.
RNF-0002	Tiempo de transacciones en operaciones especiales del sistema.
RNF-0003	Facilitar la experiencia de envío de datos desde dispositivos móviles.
RNF-0004	Resaltar campos obligatorios no ingresados.
RNF-0005	Mostrar mensajes de resultados de la operación.
RNF-0006	Cifrado de claves de acceso.
RNF-0007	Cantidad de decimales para valores numéricos.
RNF-0008	Tematización para la aplicación móvil.
RNF-0009	Resaltar líneas en gráficas de control.
RNF-0010	Permisos sobre fases cerradas.
RNF-0011	Resaltado de fases finalizadas.
RNF-0012	Registro histórico de datos de auditoría.

Identificador	Título
RNF-0013	Establecer foco en cada formulario.

Tabla 3.2 Listado de requisitos no funcionales.

Ahora presentamos el detalle de los requisitos no funcionales listados en la tabla 3.2.

IDENTIFICADOR	RNF-0001	TÍTULO	Tiempo de transacciones en operaciones comunes del sistema.
DESCRIPCIÓN	El tiempo de procesamiento de las transacciones (requerimiento y respuesta) de aquellas operaciones del sistema que no involucren generación de gráficas de control ni cálculo de valores óptimos, no debe exceder de 05 segundos.		
REQUISITOS FUNCIONALES	RF-0001 al RF-0014, RF-0018, RF-0019		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RNF-0002	TÍTULO	Tiempo de transacciones en operaciones especiales del sistema
DESCRIPCIÓN	El tiempo de procesamiento de las operaciones que involucran la generación de gráficas de control, así como del cálculo de valores óptimos del sistema, no debe exceder los 10 segundos.		
REQUISITOS FUNCIONALES	RF-0015 al RF-0017		
NECESIDAD	Deseable	PRIORIDAD	Media

IDENTIFICADOR	RNF-0003	TÍTULO	Facilitar la experiencia de envío de datos desde dispositivos móviles
DESCRIPCIÓN	Dentro de la aplicación móvil, el usuario deberá contar con un botón para importar datos y otro para exportarlos, esto con el fin de sincronizar los datos guardados en el dispositivo con el gestor de datos principal del sistema.		
REQUISITOS FUNCIONALES	RF-0014		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RNF-0004	TÍTULO	Resaltar campos obligatorios no ingresados
DESCRIPCIÓN	El sistema deberá resaltar con color rojo, todos aquellos campos obligatorios que el usuario no ha ingresado luego de enviar la solicitud de transacción al sistema. Además, mostrará un mensaje explicando esta ocurrencia.		
REQUISITOS FUNCIONALES	RF-0001 al RF-0017.		
NECESIDAD	Esencial	PRIORIDAD	Baja

IDENTIFICADOR	RNF-0005	TÍTULO	Mostrar mensajes de resultados de la operación
DESCRIPCIÓN	El sistema deberá mostrar, -dentro de la misma pantalla-, un mensaje explicativo de los resultados de la transacción solicitada. Cada tipo de mensaje contará con un icono representativo.		
REQUISITOS FUNCIONALES	Todos		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RNF-0006	TÍTULO	Cifrado de claves de acceso
DESCRIPCIÓN	El sistema deberá tener cifradas todas las claves de acceso al software usando el algoritmo MD5. Asimismo, el envío de las claves al momento de ingresar al software deberá estar cifrado.		
REQUISITOS FUNCIONALES	RF-0003 y RF-0004.		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RNF-0007	TÍTULO	Cantidad de decimales para valores numéricos
DESCRIPCIÓN	El sistema deberá permitir, para valores numéricos, el ingreso y la muestra de hasta 4 valores decimales.		
REQUISITOS FUNCIONALES	RF-0006, RF-0009, RF-0012 al RF-0017		
NECESIDAD	Esencial	PRIORIDAD	Media

IDENTIFICADOR	RNF-0008	TÍTULO	Tematización para la aplicación móvil.
DESCRIPCIÓN	La aplicación móvil para el ingreso de valores debe tener una combinación de colores en un contraste tal, que facilite el ingreso de datos dentro de un entorno a plena luz del día (campo). Dentro de este requisito también consideraremos el tamaño, tipo y grosor de los textos.		
REQUISITOS FUNCIONALES	RF-0014		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RNF-0009	TÍTULO	Resaltar líneas en gráficas de control
DESCRIPCIÓN	Al generar las gráficas de control, el sistema deberá tener un color diferente para cada línea perteneciente al dibujo. Es de especial atención que el sistema deberá dibujar con color rojo todos aquellos segmentos de la gráfica que cumplan los requisitos de “proceso fuera de control”.		
REQUISITOS	RF-0015 al RF-0017		

FUNCIONALES			
NECESIDAD	Deseable	PRIORIDAD	Media

IDENTIFICADOR	RNF-0010	TÍTULO	Permisos sobre fases cerradas.
DESCRIPCIÓN	Cuando una fase ya está cerrada, el sistema debe permitir únicamente la visualización de todos los datos almacenados.		
REQUISITOS FUNCIONALES	RF-0010 al RF-0014		
NECESIDAD	Esencial	PRIORIDAD	Alta

IDENTIFICADOR	RNF-0011	TÍTULO	Resaltado de fases finalizadas
DESCRIPCIÓN	Cuando una fase ya ha finalizado, el sistema debe mostrar tanto textos como valores de un color diferente, de una tonalidad más baja que cuando se podían modificar los datos dentro de la misma fase.		
REQUISITOS FUNCIONALES	RF-0010 al RF-0014		
NECESIDAD	Opcional	PRIORIDAD	Baja

IDENTIFICADOR	RF-0012	TÍTULO	Registro histórico de datos de auditoría
DESCRIPCIÓN	El sistema, por cada operación de cálculo, modificación o inserción realizada, debe mantener un registro histórico de cambios en base a datos de auditoría (código de usuario que lo realizó, fecha y hora de la operación)		
REQUISITOS FUNCIONALES	RF-0009, RF-0010, RF-0012, RF0014		
NECESIDAD	Deseable	PRIORIDAD	Media

IDENTIFICADOR	RNF-0013	TÍTULO	Establecer foco en cada formulario.
DESCRIPCIÓN	Al momento de mostrar una pantalla que contenga un formulario de ingreso de datos, el sistema deberá establecer el foco en el primer campo de ingresos de datos, sin importar que este sea o no obligatorio.		
REQUISITOS FUNCIONALES	Todos		
NECESIDAD	Deseable	PRIORIDAD	Baja

3.3 Matriz de trazabilidad.

La matriz de trazabilidad de requisitos es un cuadro que vincula los requisitos funcionales y no funcionales. Esta matriz ayuda a asegurar que los requerimientos tengan consistencia.

		Requisitos no funcionales													
		1	2	3	4	5	6	7	8	9	10	11	12	13	
Requisitos funcionales	1	x			x	x									x
	2	x			x	x									x
	3	x			x	x	x								x
	4	x			x	x	x								x
	5	x			x	x									x
	6	x			x	x		x							x
	7	x			x	x									x
	8	x			x	x									x
	9	x			x	x		x						x	x
	10	x			x	x					x	x	x	x	x
	11	x			x	x					x	x			x
	12	x			x	x		x			x	x	x	x	x
	13	x			x	x		x			x	x			x
	14	x		x	x	x		x	x		x	x	x	x	x
	15		x		x	x		x		x					x
	16		x		x	x		x		x					x
	17		x		x	x		x		x					x
	18	x				x									x
	19	x				x									x

Tabla 3.3 Matriz de trazabilidad

3.4 Diagramas de casos de uso.

Para los casos de uso del sistema de información propuesto, hemos identificado a los siguientes actores:

- Administrador: es el usuario que tiene acceso a todas las funcionalidades del sistema de información sin restricción.
- Analista: es el usuario que, principalmente, crea las cartas de control en el sistema y observa la evolución de los datos del proceso para extraerlos, analizarlos y generar información útil a la empresa.
- Operario: es el usuario que ingresa los datos medidos en campo y que están relacionados al proceso productivo.

3.4.1 Diagrama de casos de uso del administrador.

A continuación, mostramos el diagrama de casos de uso del administrador (figura 3.1) y luego damos detalle de cada uno de ellos.



Figura 3.1 Diagrama de casos de uso del administrador.

NOMBRE	Gestionar parámetros.
ACTORES	Administrador.
OBJETIVO	La gestión de parámetros define la búsqueda, inserción y modificación de datos de los parámetros que son medidos y controlados en el proceso de producción del langostino blanco.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	
ESCENARIO BÁSICO (Búsqueda)	<ol style="list-style-type: none"> 1. Ingresar nombre (parcial o completo) del parámetro. 2. Mostrar lista de resultados con datos de los parámetros. 3. Terminar.
ESCENARIO ALTERNO (Insertar)	<ol style="list-style-type: none"> 1. Elegir opción de ingreso de nuevo parámetro. 2. Ingresar los datos del parámetro en el formulario. 3. Guardar los datos ingresados. 4. Terminar.

ESCENARIO ALTERNO (Editar)	<ol style="list-style-type: none"> 1. Buscar parámetro (definido en el escenario básico). 2. Elegir la opción de edición para un parámetro de la lista de resultados. 3. Modificar los datos del parámetro mostrados en el formulario de edición. 4. Guardar los cambios. 5. Terminar.
-----------------------------------	---

NOMBRE	Gestionar proveedores.
ACTORES	Administrador.
OBJETIVO	La gestión de proveedores define la búsqueda, inserción y modificación de datos de los proveedores de productos y servicios relacionados al proceso de producción.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	
ESCENARIO BÁSICO (Búsqueda)	<ol style="list-style-type: none"> 1. Ingresar nombre (parcial o completo) del proveedor. 2. Mostrar lista de resultados con datos de los proveedores. 3. Terminar.
ESCENARIO ALTERNO (Insertar)	<ol style="list-style-type: none"> 1. Elegir opción de ingreso de nuevo proveedor. 2. Ingresar los datos del proveedor en el formulario. 3. Guardar los datos ingresados. 4. Terminar.
ESCENARIO ALTERNO (Editar)	<ol style="list-style-type: none"> 1. Buscar proveedor (definido en el escenario básico). 2. Elegir la opción de edición para un proveedor de la lista de resultados. 3. Modificar los datos del proveedor mostrados en el formulario de edición. 4. Guardar los cambios. 5. Terminar.

NOMBRE	Gestionar usuarios.
ACTORES	Administrador.
OBJETIVO	La gestión de usuarios define la búsqueda, inserción y modificación de datos de los usuarios del sistema, así como también, las operaciones de recuperación de clave de acceso y la deshabilitación o habilitación de un usuario.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	

ESCENARIO BÁSICO (Búsqueda)	<ol style="list-style-type: none"> 1. Ingresar nombre (parcial o completo) del usuario. 2. Mostrar lista de resultados con datos del usuario. 3. Terminar.
ESCENARIO ALTERNO (Insertar)	<ol style="list-style-type: none"> 1. Elegir opción de ingreso de nuevo usuario. 2. Ingresar los datos del usuario en el formulario. 3. Guardar los datos ingresados. 4. Terminar.
ESCENARIO ALTERNO (Editar)	<ol style="list-style-type: none"> 1. Buscar usuario (definido en el escenario básico). 2. Elegir la opción de edición para un usuario de la lista de resultados. 3. Modificar los datos del usuario mostrados en el formulario de edición. 4. Guardar los cambios. 5. Terminar.
ESCENARIO ALTERNO (Recuperar clave)	<ol style="list-style-type: none"> 1. Buscar usuario (definido en el escenario básico). 2. Elegir la opción de cambio de clave. 3. Cambiar la clave en el formulario mostrado. 4. Confirmar la realización de la operación de cambio de clave. 5. Guardar los cambios 6. Terminar.
ESCENARIO ALTERNO (Deshabilitar usuario)	<ol style="list-style-type: none"> 1. Buscar usuario (definido en el escenario básico). 2. Elegir la opción de deshabilitar al usuario para su ingreso y operación del sistema. 3. Confirmar operación de deshabilitación. 4. Guardar los cambios. 5. Terminar.
ESCENARIO ALTERNO (Habilitar usuario)	<ol style="list-style-type: none"> 1. Buscar usuario (definido en el escenario básico). 2. Elegir la opción de habilitar al usuario para su ingreso y operación en el sistema. 3. Confirmar operación de habilitación. 4. Guardar los cambios. 5. Terminar.

NOMBRE	Gestionar unidades de medida.
ACTORES	Administrador.
OBJETIVO	La gestión de unidades de medida define la búsqueda, inserción y modificación de datos de las unidades de medida usadas en el sistema.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	

ESCENARIO BÁSICO (Búsqueda)	<ol style="list-style-type: none"> 1. Ingresar nombre o abreviatura (parcial o completa) de la unidad de medida. 2. Mostrar lista de resultados con datos de las unidades de medida. 3. Terminar.
ESCENARIO ALTERNO (Insertar)	<ol style="list-style-type: none"> 1. Elegir opción de ingreso de nueva unidad de medida. 2. Ingresar los datos de la unidad de medida en el formulario. 3. Guardar los datos ingresados. 4. Terminar.
ESCENARIO ALTERNO (Editar)	<ol style="list-style-type: none"> 1. Buscar unidad de medida (definido en el escenario básico). 2. Elegir la opción de edición para una unidad de medida de la lista de resultados. 3. Modificar los datos de la unidad de medida mostrados en el formulario de edición. 4. Guardar los cambios. 5. Terminar.

NOMBRE	Gestionar sectores de piscinas.
ACTORES	Administrador.
OBJETIVO	La gestión de parámetros define la búsqueda, inserción y modificación de datos de los sectores de piscinas que puede tener una empresa en su zona de producción.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	
ESCENARIO BÁSICO (Búsqueda)	<ol style="list-style-type: none"> 1. Ingresar nombre (parcial o completo) del sector que se desea buscar. 2. Mostrar lista de resultados con datos de los sectores de piscinas. 3. Terminar.
ESCENARIO ALTERNO (Insertar)	<ol style="list-style-type: none"> 1. Elegir opción de ingreso de nuevo sector de piscinas. 2. Ingresar los datos del nuevo sector de piscinas en el formulario. 3. Agregar las piscinas que pertenecen al nuevo sector. 4. Guardar los datos ingresados. 5. Terminar.
ESCENARIO ALTERNO (Editar)	<ol style="list-style-type: none"> 1. Buscar sector de piscinas (definido en el escenario básico). 2. Elegir la opción de edición para un sector de piscinas de la lista de resultados. 3. Modificar los datos del sector de piscinas mostrados en el formulario de edición. 4. Guardar los cambios.

	5. Terminar.
--	--------------

NOMBRE	Gestionar piscinas.
ACTORES	Administrador.
OBJETIVO	La gestión de piscinas define la búsqueda, inserción y modificación de datos de las piscinas que pueden existir en la zona de producción de una empresa que cultiva langostino blanco.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	
ESCENARIO BÁSICO (Búsqueda)	<ol style="list-style-type: none"> 1. Ingresar nombre o código (parcial o completo) de la piscina. 2. Mostrar lista de resultados con datos de las piscinas. 3. Terminar.
ESCENARIO ALTERNO (Insertar)	<ol style="list-style-type: none"> 1. Elegir opción de ingreso de nueva piscina. 2. Ingresar los datos de la nueva piscina en el formulario. 3. Guardar los datos ingresados. 4. Terminar.
ESCENARIO ALTERNO (Editar)	<ol style="list-style-type: none"> 1. Buscar piscina (definido en el escenario básico). 2. Elegir la opción de edición para una piscina de la lista de resultados. 3. Modificar los datos de la piscina mostrados en el formulario de edición. 4. Guardar los cambios. 5. Terminar.

NOMBRE	Gestionar lotes.
ACTORES	Administrador.
OBJETIVO	La gestión de lotes define la búsqueda, inserción y modificación de datos de los lotes de larvas que una empresa de producción de langostino blanco puede adquirir para iniciar un proceso productivo.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Ingresar código (parcial o completo) del lote a buscar. 2. Mostrar lista de resultados con datos de los lotes.

(Búsqueda)	3. Terminar.
ESCENARIO ALTERNO (Insertar)	1. Elegir opción de ingreso de nuevo lote. 2. Ingresar los datos del lote en el formulario. 3. Guardar los datos ingresados. 4. Terminar.
ESCENARIO ALTERNO (Editar)	1. Buscar lote (definido en el escenario básico). 2. Elegir la opción de edición para un lote de la lista de resultados. 3. Modificar los datos del lote mostrados en el formulario de edición. 4. Guardar los cambios. 5. Terminar.

NOMBRE	Gestionar campaña
ACTORES	Administrador.
OBJETIVO	La gestión de campañas de siembra y cosecha del langostino blanco, define la búsqueda, inserción, modificación, inicio y término de una campaña en el sistema.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	
ESCENARIO BÁSICO (Búsqueda)	1. Ingresar código o descripción (parcial o completa) de la campaña a buscar. 2. Mostrar lista de resultados con datos de las campañas. 3. Terminar.
ESCENARIO ALTERNO (Insertar)	1. Elegir opción de inserción de nueva campaña. 2. Ingresar los datos de la campaña en el formulario. 3. Guardar los datos ingresados. 4. Terminar.
ESCENARIO ALTERNO (Editar)	1. Buscar campaña (definido en el escenario básico). 2. Elegir la opción de edición para una campaña de la lista de resultados. 3. Modificar los datos de la campaña mostrados en el formulario de edición. 4. Guardar los cambios. 5. Terminar.
ESCENARIO ALTERNO (Iniciar campaña)	1. Buscar campaña (definido en el escenario básico) 2. Elegir la opción de iniciar de una campaña de la lista de resultados. 3. Ingresar fecha de inicio de la campaña. 4. Guardar los cambios. 5. Terminar.

ESCENARIO ALTERNO (Cerrar campaña)	<ol style="list-style-type: none"> 1. Buscar campaña (definido en el escenario básico). 2. Elegir la opción de cierre de una campaña de la lista de resultados. 3. Ingresar fecha de cierre de la campaña. 4. Cambiar el estado de todas las fases de la campaña a “cerrada”. 5. Terminar.
---	---

NOMBRE	Configurar fase.
ACTORES	Administrador.
OBJETIVO	La configuración de una fase tiene por objetivo ingresar y aperturar una nueva fase de crianza del langostino blanco.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	- La fase creada se encuentra iniciada.
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Ingresar a la opción de creación de una fase. 2. Ingresar datos de la fase en el formulario mostrado. 3. Guardar los datos ingresados. 4. Terminar.
ESCENARIO ALTERNO	<p>Paso 3: No se ingresaron los datos obligatorios.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje explicando el error. 2. Ir al paso 2.

NOMBRE	Agregar piscinas a fase activa.
ACTORES	Administrador.
OBJETIVO	Este caso de uso tiene por objetivo agregar piscinas a una fase activa para iniciar o continuar el proceso productivo del langostino blanco.
PRECONDICIONES	<ul style="list-style-type: none"> - El usuario debe haber iniciado sesión en el sistema. - Debe existir una fase activa. - Deben existir piscinas registradas.
POSTCONDICIONES	
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Elegir la opción de consultar datos de una fase. 2. Elegir la opción de agregar piscina. 3. Buscar y elegir una o varias piscinas para agregar. 4. Guardar y terminar.
ESCENARIO ALTERNO	<p>Paso 1: La fase elegida no está activa.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje explicativo.

	<p>2. Terminar.</p> <p>Paso 3: No hay piscinas.</p> <p>1. Mostrar mensaje explicativo.</p> <p>2. Terminar.</p>
--	--

NOMBRE	Asignar límite a fase activa.
ACTORES	Administrador.
OBJETIVO	Este caso de uso tiene por objetivo asignar límites a un parámetro que será usado en una fase activa.
PRECONDICIONES	<ul style="list-style-type: none"> - El usuario debe haber iniciado sesión en el sistema. - Debe existir una fase activa. - Deben existir parámetros activos.
POSTCONDICIONES	
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Elegir la opción de consultar datos de una fase. 2. Elegir la opción de asignar límite a fase. 3. Buscar y elegir un parámetro para asignarle límites. 4. Ingresar los valores numéricos de los límites al parámetro elegido. 5. Guardar y terminar.
ESCENARIO ALTERNO	<p>Paso 1: La fase elegida no está activa.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje explicativo. 2. Terminar. <p>Paso 3: No hay parámetros.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje explicativo. 2. Terminar.

NOMBRE	Consultar datos de fase.
ACTORES	Administrador.
OBJETIVO	La consulta de datos de una fase tiene por objetivo mostrar todos los datos ingresados o generados para la fase en cuestión.
PRECONDICIONES	<ul style="list-style-type: none"> - El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Elegir la opción de buscar una fase. 2. Ingresar datos de búsqueda de la fase. 3. Mostrar lista de resultados. 4. Elegir la visualización de datos de una fase de la lista de resultados. 5. Mostrar de forma ordenada, todos los datos ingresados y generados para la fase elegida de la lista de resultados.

	6. Terminar.
ESCENARIO ALTERNO	Paso 3: No hay resultados. 1. Mostrar mensaje explicativo. 2. Ir al paso 2.

3.4.2 Diagrama de casos de uso del analista.

En la figura 3.2 observamos el diagrama de casos de uso del analista, luego, se observa la descripción textual detallada de los mismos.

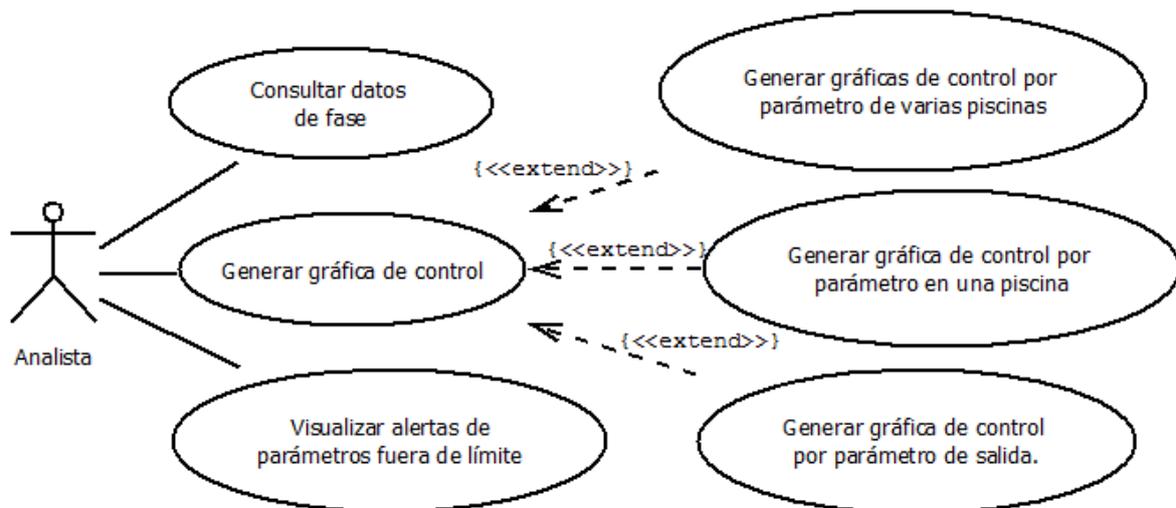


Figura 3.2 Diagrama de casos de uso del analista.

NOMBRE	Consultar datos de fase
ACTORES	Analista, administrador, operario.
OBJETIVO	Mostrar, de forma ordenada, todos los datos de una fase.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema. - Deben existir datos almacenados para las fases.
POSTCONDICIONES	
ESCENARIO BÁSICO	1. Elegir piscina. 2. Elegir etapa. 3. Elegir campaña 4. Ingresar posible fecha de inicio. 5. Ingresar posible fecha de término. 6. Buscar fases. 7. Mostrar resultados coincidentes de fases, señalando su campaña y sus fechas de inicio y fin. 8. Seleccionar una fase. 9. Mostrar pantalla con datos categorizados y ordenados.

	10. Terminar.
ESCENARIO ALTERNO	Paso 7: No hay resultados. 1. Mostrar mensaje explicativo. 2. Ir al paso 1.

NOMBRE	Generar gráfica de control.
ACTORES	Analista.
OBJETIVO	Mostrar una gráfica de control en base a un determinado conjunto de datos.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema. - Deben existir suficiente cantidad de datos para la generación de la gráfica (mínimo 20 puntos)
POSTCONDICIONES	
ESCENARIO BÁSICO	1. Calcular el valor de límite superior. 2. Calcular el valor del límite inferior. 3. Calcular el valor del punto central. 4. Generar conjunto de puntos para los límites superior, inferior y medio. 5. Generar el conjunto de puntos para la gráfica de control. 6. Dentro de los puntos definidos en el paso anterior, hallar aquellos conjuntos de puntos que formarán segmentos que denotarán que el proceso está “fuera de control”. 7. Con todos los conjuntos de datos, dibujar la gráfica de control. 8. Terminar.
ESCENARIO ALTERNO	Pasos 1 al 5: No hay suficiente cantidad de datos. 1. Mostrar mensaje explicativo. 2. Terminar.

NOMBRE	Generar gráfica de control por parámetro en una piscina.
ACTORES	Analista.
OBJETIVO	Mostrar una gráfica de control de un parámetro medido, a través de las distintas etapas de una o varias campañas, en una piscina determinada.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema. - Deben existir datos almacenados para el parámetro y la piscina seleccionados.
POSTCONDICIONES	
ESCENARIO BÁSICO	1. Elegir parámetro. 2. Elegir piscina.

	<ol style="list-style-type: none"> 3. Elegir etapa. 4. Elegir rango de campañas. 5. Hallar conjunto total de valores medidos. 6. Generar gráfica de control. 7. Terminar.
ESCENARIO ALTERNO	<p>Paso 5: No hay datos suficientes.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje explicativo. 2. Ir al paso 1. <p>Paso 6: Error: no cumple condiciones para la generación.7</p> <ol style="list-style-type: none"> 1. Terminar.

NOMBRE	Generar gráfica de control por parámetro en varias piscinas.
ACTORES	Analista.
OBJETIVO	Mostrar una gráfica de control de un parámetro medido, en varias piscinas, en una o varias campañas.
PRECONDICIONES	<ul style="list-style-type: none"> - El usuario debe haber iniciado sesión en el sistema. - Deben existir datos almacenados para el parámetro y para todas las piscinas seleccionadas.
POSTCONDICIONES	
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Elegir parámetro. 2. Elegir piscinas. 3. Elegir rango de campañas. 4. Hallar conjunto total de valores medidos. 5. Generar gráfica de control. 6. Terminar.
ESCENARIO ALTERNO	<p>Paso 4: No hay datos suficientes.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje explicativo. 2. Ir al paso 1. <p>Paso 5: Error: no cumple condiciones para la generación.7</p> <ol style="list-style-type: none"> 1. Terminar.

NOMBRE	Generar gráfica de control por parámetro de salida.
ACTORES	Analista.
OBJETIVO	Mostrar una gráfica de control de un parámetro de salida medido a través del tiempo.
PRECONDICIONES	<ul style="list-style-type: none"> - El usuario debe haber iniciado sesión en el sistema. - Deben existir datos almacenados para el parámetro de salida.
POSTCONDICIONES	
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Elegir parámetro de salida. 2. Elegir rango de tiempo.

	<ol style="list-style-type: none"> 3. Hallar conjunto total de valores medidos. 4. Generar gráfica de control. 5. Terminar.
ESCENARIO ALTERNO	<p>Paso 3: No hay datos suficientes.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje explicativo. 2. Ir al paso 1. <p>Paso 4: Error: no cumple condiciones para la generación.</p> <ol style="list-style-type: none"> 1. Terminar.

NOMBRE	Visualizar alertas de parámetros fuera de límite
ACTORES	Analista, administrador, operario.
OBJETIVO	Mostrar, de forma ordenada, todas las alertas generadas por el ingreso de valores fuera de los límites establecidos para cierto parámetro en una fase.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Mostrar lista de alertas 2. Elegir una alerta 3. Ver los datos de la alerta. 4. Terminar.
ESCENARIO ALTERNO	<p>Paso 4: Ya se vio la alerta.</p> <ol style="list-style-type: none"> 1. Cambiar estado de alerta a “visualizada”. 2. Terminar.

3.4.3 Diagrama de casos de uso del operario.

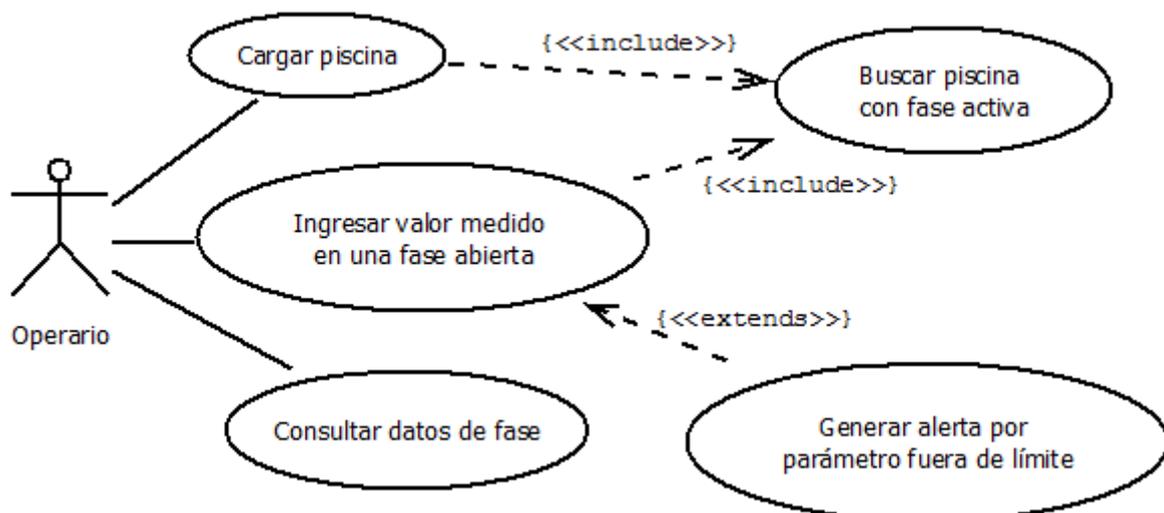


Figura 3.3 Diagrama de casos de uso del operario.

NOMBRE	Buscar piscina con fase activa.
ACTORES	Operario, administrador.

OBJETIVO	Esta búsqueda tiene por objetivo mostrar al usuario una o varias piscinas que, para determinada etapa, cuenta con una fase activa.
PRECONDICIONES	- El usuario debe haber iniciado sesión en el sistema.
POSTCONDICIONES	- Se muestra el listado de resultados con opción a selección única.
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Ingresar nombre o código de piscina. 2. Seleccionar etapa de la fase a buscar. 3. Mostrar lista de resultados, incluyendo únicamente aquellas piscinas que cuentan con una fase activa. 4. Terminar.
ESCENARIO ALTERNO	<p>Paso 3: No hay resultados.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje explicativo. 2. Volver al paso 1.

NOMBRE	Ingresar valor medido en una fase abierta.
ACTORES	Operario, administrador, analista
OBJETIVO	Ingresar el valor de un parámetro medido para una fase activa.
PRECONDICIONES	<ul style="list-style-type: none"> - El usuario debe haber iniciado sesión en el sistema. - Fase en estado: abierta. - Deben existir proveedores registrados.
POSTCONDICIONES	Valor almacenado para el parámetro y fase indicados.
ESCENARIO BÁSICO.	<ol style="list-style-type: none"> 1. Buscar y seleccionar la fase activa de la piscina 2. Elegir que empresa es la que ha hecho la medición 3. Elegir parámetro a medir. 4. Ingresar valor medido. 5. Guardar y terminar.
ESCENARIO ALTERNO	<p>Paso 1: No hay fase activa.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje de error. 2. Terminar. <p>Paso 2: No hay proveedores</p> <ol style="list-style-type: none"> 1. Informar que el proveedor será su misma empresa. <p>Paso 5: El valor guardado está fuera de los límites establecidos para el parámetro.</p> <ol style="list-style-type: none"> 1. Generar alerta de parámetro fuera de límite. <p>Paso 5: Desea ingresar otro valor</p> <ol style="list-style-type: none"> 1. Volver al paso 2

NOMBRE	Cargar piscina.
ACTORES	Operario, administrador.

OBJETIVO	En cierto momento de una fase, a alguna de sus piscinas, puede agregársele una cantidad de larvas provenientes de otro lote.
PRECONDICIONES	<ul style="list-style-type: none"> - El usuario debe haber iniciado sesión en el sistema. - El lote ya debe estar registrado. - La piscina seleccionada debe tener una sola fase activa.
POSTCONDICIONES	<ul style="list-style-type: none"> - La piscina, dentro de su fase activa, debe tener registrado el porcentaje de aporte de masa del lote.
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Buscar y seleccionar una fase activa 2. Buscar y seleccionar una piscina registrada para la fase activa. 3. Buscar y seleccionar el lote aportante. 4. Ingresar datos de masa aportante (cantidad, peso, densidad) 5. Ingresar fecha de registro. 6. Guardar y terminar
ESCENARIO ALTERNO	<p>Paso 1: No hay fase activa.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje de error. 2. Terminar. <p>Paso 2: No hay piscina registrada para la fase activa.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje de error. 2. Terminar. <p>Paso 3: No hay lote disponible.</p> <ol style="list-style-type: none"> 1. Mostrar mensaje de error. 2. Terminar. <p>Paso 6: No se registraron datos</p> <ol style="list-style-type: none"> 1. Mostrar mensaje de error.

NOMBRE	Generar alerta por parámetro fuera de límite.
ACTORES	Operario.
OBJETIVO	Por cada valor ingresado, el sistema debe definir si éste está dentro o fuera del límite establecido para el parámetro, en la semana correspondiente a la fase para la cual se está ingresando el mencionado valor. Si lo ingresado está fuera del límite establecido, el sistema automáticamente generará una alerta.
PRECONDICIONES	<ul style="list-style-type: none"> - El usuario debe haber iniciado sesión en el sistema. - Se debe haber ingresado correctamente un valor para un parámetro en cierta fase activa.
POSTCONDICIONES	<ul style="list-style-type: none"> - Alerta registrada con estado “no visualizado”.
ESCENARIO BÁSICO	<ol style="list-style-type: none"> 1. Ingresar código del valor recientemente medido. 2. Ingresar código del límite trasgredido. 3. Definir a la alerta como “no visualizada” 4. Guardar y terminar

ESCENARIO ALTERNO	Paso 4: No se registró la alerta. 1. Mostrar mensaje explicando el hecho. 2. Terminar.
--------------------------	--

3.5 Diagrama de clases.

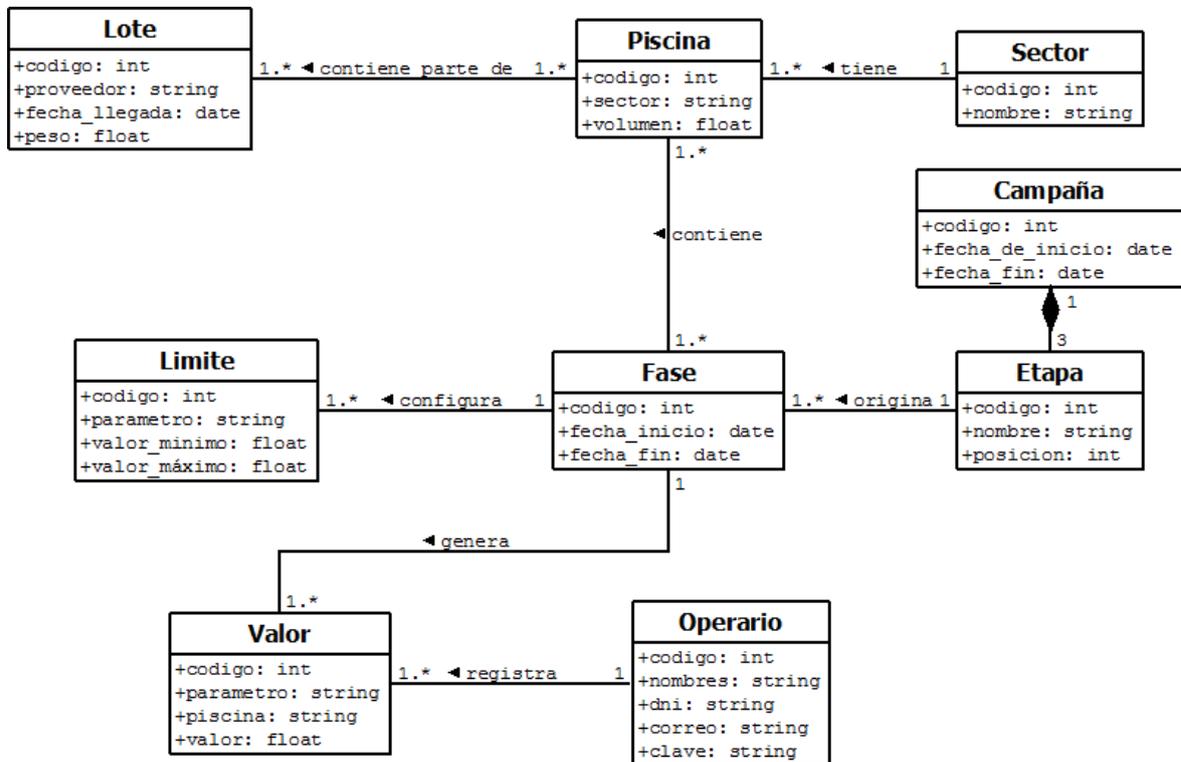


Figura 3.4. Diagrama de clases.

Sabiendo que un diagrama de clases modela un conjunto de conceptos y reglas para representar los aspectos lógicos de la realidad y que estos deben estar contenidos en los requisitos del futuro sistema de información, pasaremos a explicar cual es el dominio del problema que es motivo de este trabajo de tesis.

La empresa inicia una campaña de producción de langostino la cual contiene principalmente 3 etapas: adaptación, pre cría y engorde; vale resaltar que las etapas no existen por si mismas si es que no se ha iniciado una campaña.

Debido a que una campaña tiene muchos ciclos productivos, los cuales están compuestos por etapas, nace el concepto de fases que, en sí, es el desarrollo de una etapa en un periodo de tiempo determinado, dentro de una campaña. Las fases generan el uso de muchas piscinas, organizadas por sectores, y a las cuales se les carga con biomasa de uno o varios lotes de origen de larvas de langostino.

Dentro de todo el ciclo productivo, en todas sus fases, el operario que ejecuta sus labores directamente en el área de producción, registra una serie de valores, resultado de la medición de parámetros de la operación. Cada vez que se realiza el registro del valor de un

parámetro, se debe verificar que, dicho valor, no exceda los límites establecidos para el mismo entro de una fase, por tanto, en una fase se pueden establecer uno o muchos límites a los parámetros.

En la figura 3.4 mostramos el diagrama de clases producto de la definición del dominio del problema previamente descrito.

Capítulo 4

Diseño del Sistema de Información

4.1 Introducción..

El diseño de este sistema de información, toma en cuenta la necesidad de uso del mismo en el contexto físico y funcional de las operaciones de producción de langostino blanco en nuestra región. Así, el mencionado sistema comprende el desarrollo de una aplicación móvil, una aplicación web y, además, debe permitir posteriores interacciones con otros sistemas que necesiten el acceso a sus datos o a la información que se pueda generar.

Tomando en cuenta el limitado o nulo acceso a internet en las áreas de trabajo (campo y oficinas en planta), es que esta tesis sustenta un diseño arquitectónico basado en conexión de red de área local, tal como se explicará con más detalle en la sección de "Arquitectura".

Por lo mencionado, el sistema de información consta de los siguientes aplicativos:

- *Aplicación web*: utilizada por los “administradores” y “analistas” (descritos en el capítulo 3). El acceso a la aplicación será, principalmente, a través de computadores personales dentro de las oficinas de la organización que implemente este sistema.
- *Aplicación móvil*: será utilizada por los “operarios” (descritos en el capítulo 3), es decir, los encargados de ingresar los datos desde la zona misma de las operaciones (en campo) donde, generalmente, no se cuenta con acceso a internet, por ello, para este ingreso, se hará uso de dispositivos móviles y se contará con la funcionalidad de sincronización de datos cuando el aplicativo pueda conectarse de forma estable y segura a la red de área local de la empresa. Finalmente, para lograr una correcta sincronización, se desarrollará un aplicativo intermediario entre la aplicación móvil y la lógica del negocio, esta última, también utilizada por la aplicación web.

4.2 Arquitectura.

La arquitectura del sistema de información consta de dos apartados:

- *Arquitectura lógica*: expresa como se descompone el software en módulos y como estos se relacionan entre sí.

- **Arquitectura física:** expresa como se distribuye el software entre los componentes físicos que participan en nuestra solución, así como la relación entre ellos.

Cabe señalar que para este estudio, se considera que el sistema no accederá a internet, si no, a una red de área local, esto debido a que las zonas de producción del langostino en la región, están ubicadas en zonas alejadas de las ciudades, en las cuales, el acceso a internet es escaso o, en su defecto, de baja calidad.

4.2.1 Arquitectura lógica.

El sistema de información se desarrollará usando el patrón de arquitectura MVC (Modelo Vista Controlador), el cual, define la organización independiente del Modelo (objetos de negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del flujo de trabajo de la aplicación).

De esta forma, se divide el sistema en tres capas donde por un lado está el encapsulamiento de los datos; la interfaz o vista por otro y por último, la lógica interna o controlador.

El patrón de arquitectura MVC está compuesto por:

Modelo: contiene el núcleo de la funcionalidad (dominio) de la aplicación. Además, encapsula el estado de la aplicación y es independiente del controlador y la vista.

Vista: Es la presentación del Modelo y puede acceder a este pero nunca cambiar su estado, además, puede ser notificada cuando hay un cambio de estado en el modelo.

Controlador: reacciona a la petición del cliente, ejecutando la acción adecuada y creando el modelo pertinente.

Para entender cómo funciona este patrón, se debe entender la división a través del conjunto de estos tres elementos y como estos componentes se comunican unos con otros, además de como lo hacen con otras vistas y controladores externos al modelo principal. Para ello, es importante saber que el controlador interpreta las entradas del usuario, enviando el mensaje de acción al modelo y a la vista para que se proceda con las operaciones que se consideren adecuadas.

Los beneficios directos al escoger esta arquitectura lógica para este trabajo de tesis se fundamentan en la mejora de la reutilización, la mejora de la extensibilidad, la flexibilidad y el mantenimiento del desarrollo del sistema de información.

Ahora, se procede a detallar la secuencia de pasos en el flujo de información haciendo uso de la arquitectura MVC:

1. El usuario interactuará con la interfaz gráfica (componente vista) del sistema de información enviando mensajes a la capa de control; por ejemplo: el analista necesita visualizar los valores que tiene un parámetro a lo largo de una fase, para esto indica el

parámetro, la etapa y la campaña en la que desea ubicarse; ingresados estos datos, finaliza la acción presionando el botón "Buscar".

2. El componente controlador recibe el mensaje y se encarga de gestionar la operación solicitada; por ejemplo: el controlador recibe la generación del reporte con los datos de parámetro, etapa y campaña.
3. El controlador accede al componente modelo para realizar alguna transacción (insertar, actualizar o eliminar) o para consultar datos; por ejemplo: la operación de búsqueda de datos para el reporte, sólo consultará datos del modelo.
4. El controlador se encarga de reenviar los datos a los diferentes objetos del componente vista.
5. El componente vista obtiene sus datos del componente modelo, generando la interfaz apropiada para el usuario; por ejemplo: el componente vista mostrará un gráfico de líneas con los valores históricos del parámetro, -para la etapa y campaña deseados-, junto con la fecha de toma del valor.
6. El componente vista esperará nuevas interacciones comenzando el ciclo nuevamente.

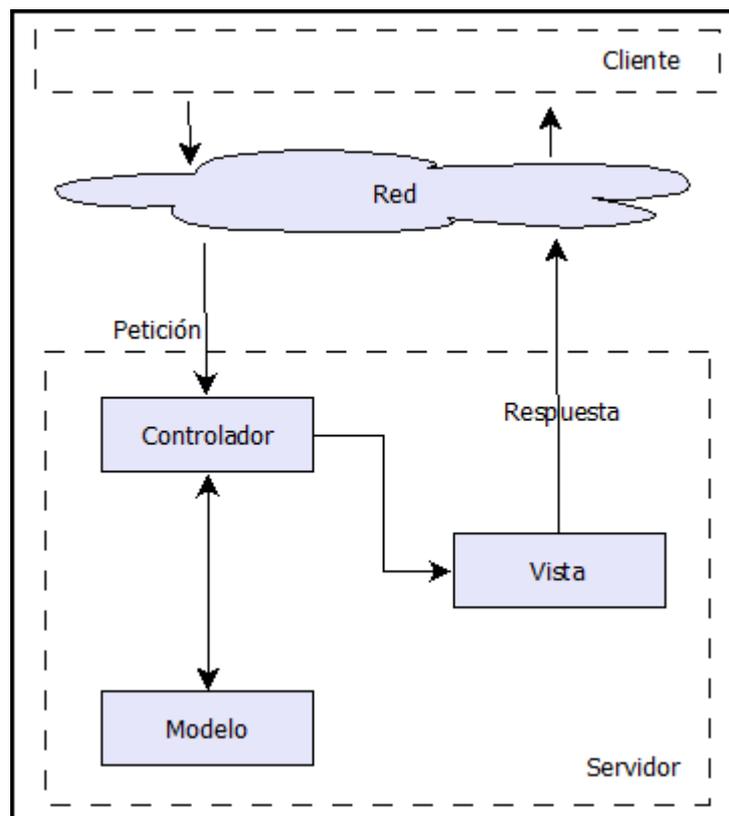


Figura 4.1 Arquitectura de software MVC.

La figura 4.1 presenta de forma gráfica lo antes explicado; obsérvese también que ella no muestra el término "Internet", en su lugar, aparece el término "Red", esto se debe a que este sistema de información está planeado a instalarse dentro de una red de área local.

4.2.2 Arquitectura física.

El sistema de información será desarrollado utilizando herramientas de tecnología web y de base de datos, bajo el entorno de una arquitectura Cliente – Servidor.

Servidor: es el rol que desempeña un equipo ofreciendo un conjunto de servicios a los clientes, tales como manejo de archivos, impresión, páginas web, direccionamiento de correo electrónico, actualización de base de datos y control de acceso.

Cliente: es el rol que desempeña un equipo demandando servicios de los servidores, pero también puede realizar procesamiento local, tales como desplegar páginas web, mostrar ventanas y generar correo electrónico.

El servidor ejecutará sobre sí mismo tres servicios en paralelo: un servidor Web, un servidor de aplicaciones y un sistema de gestión de base de datos.

1. El servidor Web se encargará de mostrar las interfaces gráficas en formato HTML, las cuales son enviadas a los clientes (web) por medio del protocolo HTTP.
2. El servidor de aplicaciones se encargará de generar las interfaces gráficas, ejecutará la lógica del negocio y además se encargará de comunicarse con la base de datos para realizar transacciones o recuperar información de ella.
3. El sistema de gestión de base de datos se encargará de mantener activa la base de datos del sistema de información, la cual contendrá datos relevantes de la operación y la producción.

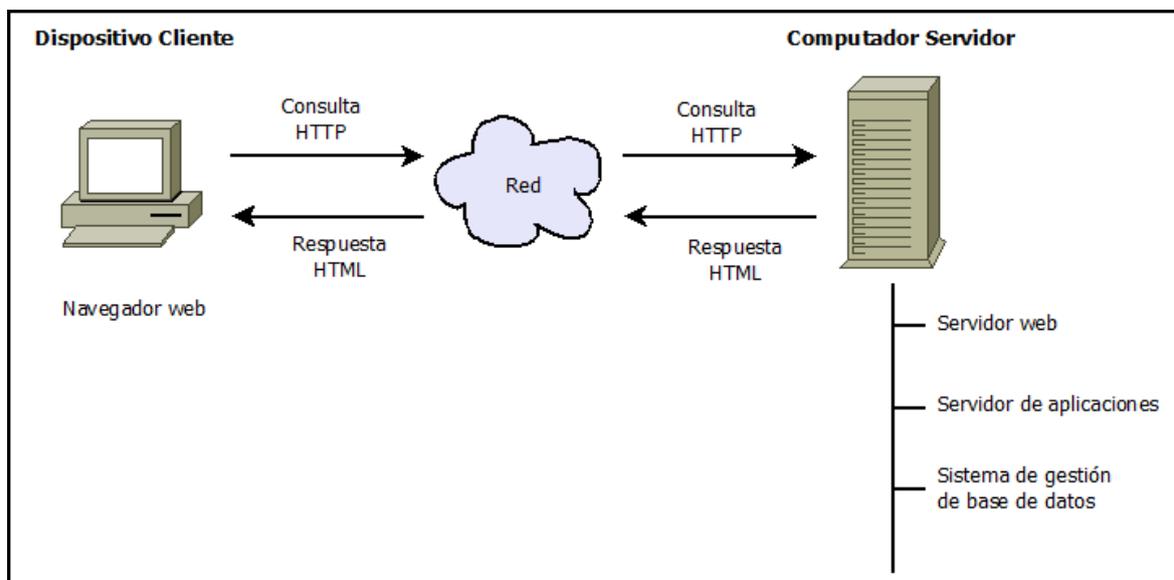


Figura 4.2 Arquitectura física.

Nótese que en la figura 4.2, también se muestra el término “Red” en lugar de “Internet”, además, cuando se menciona al cliente, se hace referencia al cliente web, al cliente móvil, así como también a otras aplicaciones que pudieran interactuar con el sistema.

4.3 Software y lenguaje de programación.

Para la fase de desarrollo y puesta en producción del sistema de información propuesto, se recomienda usar el siguiente software y lenguaje de programación:

Software	Características
Lenguaje de programación	PHP versión 5.6 o superior
Lenguaje de programación (para móviles)	Java Runtime Edition 1.8 o superior
Servidor web	Apache versión 2.2 o superior
Servidor de base de datos	MySQL versión 5.7 o superior
Sistema operativo	Linux Ubuntu versión 15 o superior
Servidor FTP	FileZilla server
Diseñador de bases de datos	MySQL Workbench 6.3 o superior
IDE de manejo de datos	MySQL Workbench 6.3 o superior
Editor de PHP	Sublime Text 3 o superior
Editor Android	Android Studio 2.1.2 o superior

Tabla 4.1 Software y lenguajes de programación recomendados.

El sustento para recomendar el software y lenguajes de programación que figura en la tabla 4.1, se sustenta en dos aspectos: técnicos y económicos.

En cuanto al aspecto económico, el uso del software y lenguajes de programación recomendados, al ser de acceso gratuito en sus versiones básicas, -suficientes para nuestro caso-, permiten un ahorro significativo en adquisición de licencias. Además, estos lenguajes de programación son altamente usados²² en el mercado del desarrollo de sistemas de información, con lo cual, la oferta de analistas y programadores calificados es mayor, lo que repercute en un ahorro en costos de capacitación y de búsqueda de personal para el proyecto. Finalmente, el hardware y dispositivos que se pueden usar cuando el sistema de información esté en su fase de producción, es el de mayor comercialización en el mercado de equipos y, también, esto representa una ventaja en la adquisición de equipos de calidad a bajo costo.

En cuanto a aspectos técnicos, basados en la experiencia propia de quien suscribe, el software y los lenguajes de programación recomendados cumplen con los requisitos de seguridad, escalabilidad, flexibilidad y estabilidad necesarios para el proyecto, tal como lo haría software y lenguajes de programación que requieren licenciamiento.

4.4 Diseño de pantallas.

Para el sistema de información que se está diseñando, se propone el uso de dos interfaces principales: una web y una móvil.

²² Afirmación basada en el artículo “The 2016 Top Programming Languages”, publicado el 26 de julio del 2016 en la revista digital “Spectrum” de la IEEE (<http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>)

En la interfaz web se llevarán a cabo todas las operaciones propias del administrador y del analista y, en la interfaz móvil, se llevarán a cabo las operaciones del operario, para mayor referencia de las características de los roles mencionados, se puede consultar el capítulo 3 de esta tesis.

4.4.1 Diseño de pantallas para la interfaz web.

Se propone que la interfaz web cuente con los siguientes módulos:

- Módulo de gestión de catálogos.
- Módulo de gestión de lotes.
- Módulo de gestión de fases.
- Módulo de reportes.

A continuación se muestran los diseños de pantallas más importantes de cada módulo.

Módulo de gestión de catálogos.

Aquí se realiza la búsqueda, inserción, edición y eliminación de registros base para el funcionamiento del sistema de información. Presentaremos las pantallas más importantes.

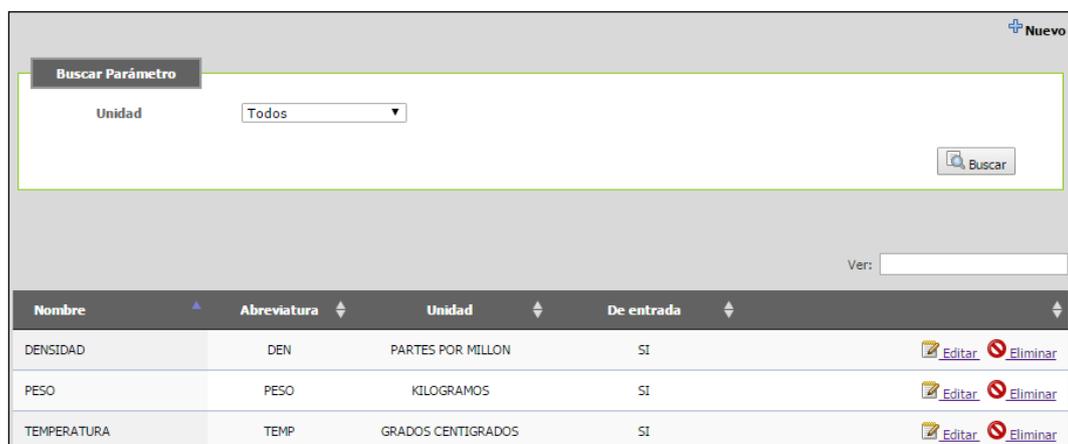


Figura 4.3 Diseño de pantalla de gestión de parámetros.

Gestión de parámetros: aquí se registrarán todos aquellos parámetros de la operación que sean susceptibles a medición como parte del control del proceso productivo. Estos parámetros son la base principal del registro de valores y de los reportes.

Gestión de piscinas: en esta pantalla se gestiona el registro de todas las piscinas que la empresa ha de usar para el proceso de producción del langostino blanco.

Código	Nombre	Sector	Superficie	Activa		
P0001	PISCINA 1	SECTOR 1	10000.00	SI		
P0002	PISCINA 2	SECTOR 1	15000.00	SI		
P0003	PISCINA 3	SECTOR 2	15000.00	SI		
P0004	PISCINA 4	SECTOR 2	20000.00	SI		

Figura 4.4 Diseño de la pantalla de gestión de piscinas.

Gestión de periodos: un periodo (o “campaña”) es un espacio de tiempo en el que la empresa inicia diversos procesos de producción en base a los lotes de larvas adquiridos.

Código	Descripcion	Fecha Inicio	Fecha Fin		
CAMPAÑA 1	PERIODO I	01/10/2016	31/12/2016		
CAMPAÑA 2	PERIODO II	01/01/2017	01/07/2017		

Figura 4.5 Diseño de la pantalla de la gestión de periodos.

Módulo de gestión de lotes.

Proveedor	Nombre	Periodo	Cant.	Peso	Fec. llegada			
PROVEEDOR 1	LOTE 1	CAMPAÑA 1	200000.00	0.090000	01/08/2016 10:00:00			
PROVEEDOR 1	LOTE 2	CAMPAÑA 1	300000.00	0.090000	01/08/2016 10:00:00			

Figura 4.6. Diseño de pantalla de la gestión de lotes.

Contiene las funcionalidades relacionadas a la administración de datos de los lotes de larvas de langostino blanco, comprados a distintos proveedores. La figura 4.6 presenta el diseño de pantalla de vista de datos de un lote.

Módulo de gestión de fases.

Las fases son espacios de tiempo en los que se ejecuta una etapa del proceso de producción del langostino. Las operaciones, de las cuales se visualiza el diseño de pantallas, están definidas en los requisitos del sistema de información (capítulo 3, apartado 1). En la figura 4.7 se visualiza el diseño de pantalla de datos de fases.

Figura 4.7 Diseño de pantalla de datos de fase.

Asignación de piscinas: una fase de producción cuenta con piscinas que la empresa asigna. En la figura 4.8 se ve el diseño de pantalla de esta operación.

Sector	Piscina	Superficie	Eliminar
SECTOR 1	PISCINA 1	10000.00	
SECTOR 1	PISCINA 2	15000.00	
SECTOR 2	PISCINA 3	15000.00	
SECTOR 2	PISCINA 4	20000.00	

Figura 4.8 Diseño de la pantalla de asignación de piscinas a una fase.

Definición de límites de una fase: para toda fase, según sean sus características, estándares nacionales e internacionales, la investigación teórica o la experiencia práctica, se definen límites para aquellos parámetros que necesitan ser seguidos y controlados con el fin de obtener un resultado esperado en diferentes momentos de la producción. Así, la figura 4.9, presenta el diseño de la pantalla de definición de límites para una fase del proceso productivo del langostino blanco.

Registrar/Modificar datos ✖

Etap	ENGORDE	Periodo	CAMPAÑA 1
Fase Inicio	01/09/2016	Fase Fin	31/10/2016
Terminado	NO	Parámetro	PESO ▾
Valor Max.	<input type="text"/>	Valor Min.	<input type="text"/>
Valor Medio.	<input type="text"/>	Semana	SEMANA ▾

Registrar

Parámetro	Val. Max.	Val. Min.	Val. Medio	Semana	Inicio	Fin	
PESO	3.00	1.00	2.00	SEMANA 1	01/09/2016	07/09/2016	Eliminar
PESO	6.00	3.00	4.50	SEMANA 2	08/09/2016	14/09/2016	Eliminar
PESO	9.00	6.00	7.50	SEMANA 3	15/09/2016	21/09/2016	Eliminar

Figura 4.9 Diseño de pantalla de definición de límites en una fase.

Carga de piscinas con lotes: una piscina alberga uno o muchos lotes cuando se encuentra en uso dentro de una fase productiva, el diseño de pantalla de este registro se muestra en la figura 4.10.

Buscar Piscinas

Sector ▾ Todos

Buscar

Ver:

Sector	Piscina	Superficie	
SECTOR 1	PISCINA 1	10000.00	Agregar lotes
SECTOR 1	PISCINA 2	15000.00	Agregar lotes
SECTOR 2	PISCINA 3	15000.00	Agregar lotes
SECTOR 2	PISCINA 4	20000.00	Agregar lotes

Figura 4.10 Diseño de pantalla de la carga de piscinas con lotes.

Módulo de reportes.

En este módulo se muestran los reportes que son utilizados para el control estadístico del proceso productivo.

Reporte por piscina: el sistema de información permite ver, -por piscina-, que parámetro se está desviando de los límites indicados en una semana determinada. Esto lo realiza a través de una carta de control (ver figura 4.11).

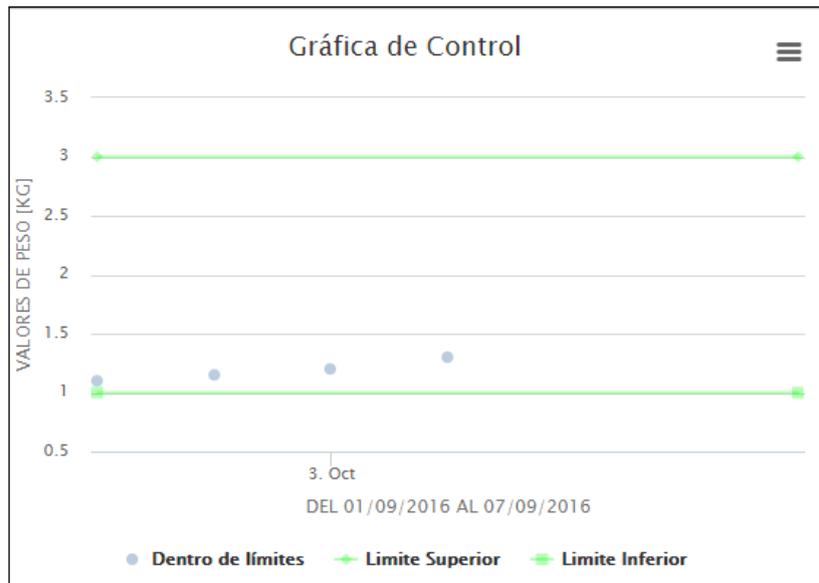


Figura 4.11 Diseño de pantalla para la carta de control por piscina.

Reporte por sector: el sistema de información permite ver, -en un sector de piscinas-, que parámetro se está desviando de los límites indicados en una semana determinada. Esto también se realiza a través de una carta de control.

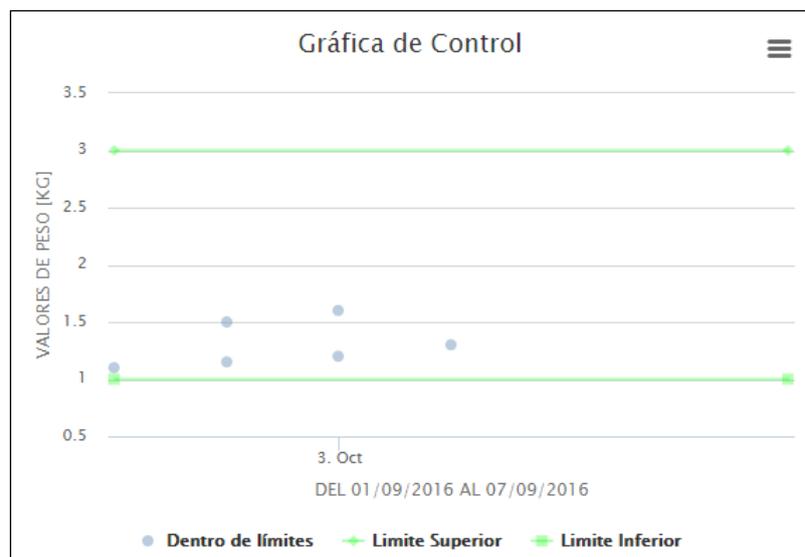


Figura 4.12 Diseño de pantalla para la carta de control por sector de piscinas

Exportar reporte a hoja Excel: el sistema de información propuesto podrá exportar los datos de un reporte a una hoja de cálculo electrónica en formato Excel. En la figura 4.13 se muestra un ejemplo de cómo se vería la hoja electrónica tras una exportación de datos.

	A	B	C	D	E	F	G	H	I
1									
2	Campaña	2016-I	Fechalnicio	01/09/2016					
3	Etapa	ENGORDE	Fecha Fin	31/10/2016					
4									
5	Lote	Sector	Piscina	Parámetro	Unidad	Valor	Fecha Registro	L. Inferior	L. Superior
6	LOTE 1, LOTE 2	SECTOR 1	PISCINA 1	PESO	KILOGRAMOS	1.1	01/09/2016	1	3
7	LOTE 1, LOTE 2	SECTOR 1	PISCINA 1	PESO	KILOGRAMOS	1.15	02/09/2016	1	3
8	LOTE 1, LOTE 2	SECTOR 1	PISCINA 1	PESO	KILOGRAMOS	1.2	03/09/2016	1	3
9	LOTE 1, LOTE 2	SECTOR 1	PISCINA 1	PESO	KILOGRAMOS	1.3	04/09/2016	1	3
10									

Figura 4.13 Ejemplo de hoja Excel luego de exportación de datos de un reporte.

4.4.2 Diseño de pantallas para la interfaz móvil.

Según lo previamente mencionado, la aplicación móvil contempla las funcionalidades del operario que, en campo, registra los valores medidos para los parámetros en los distintos ciclos productivos. La figura 4.14 presenta una propuesta para el inicio de pantalla de la aplicación móvil, aquí se observa que existe un inicio de sesión y, de ser exitoso el mismo, se muestra una pantalla con datos de usuario y las funcionales a elegir.

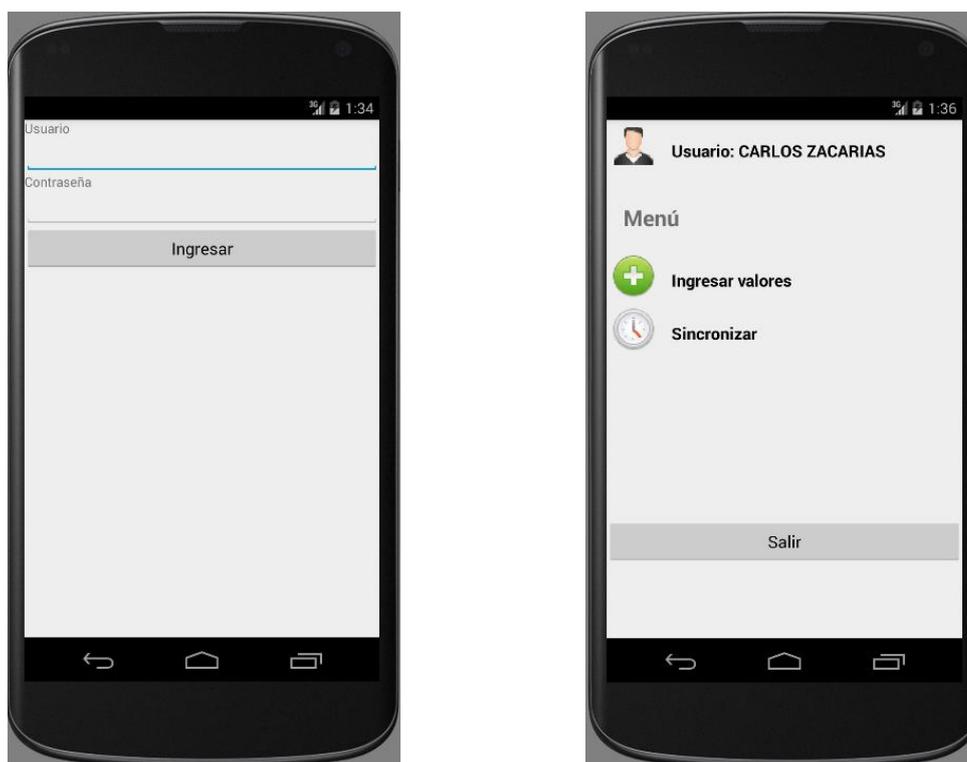


Figura 4.14: Diseño de pantalla de inicio de la aplicación móvil.

Registro de valores: aquí se registran los valores de los parámetros medidos en una piscina. El operario hará lo siguiente: elegirá un sector, luego una piscina, acto seguido: ingresará el valor y, finalmente, confirmará el valor ingresado con los datos solicitado anteriormente (ver figura 4.15).



Figura 4.15: Diseño de pantalla para el registro de valores.

Sincronización de datos: en esta funcionalidad, los datos ingresados a través de la aplicación móvil en una zona donde no hay acceso a una red de la empresa, se podrán transferir hacia la base de datos del sistema de información al momento que sea segura la existencia de una red privada de la organización.



Figura 4.16: Diseño de la pantalla de sincronización de datos.

4.5 Diseño de base de datos.

4.5.1 Modelo Entidad Relación.

A continuación se presenta el diagrama entidad relación propuesto para el sistema de información, para tal fin, se dividid la presentación del mismo en dos partes para otorgar al lector una mejor visibilidad.

En la figura 4.17 se observa que los datos de las funcionalidades del sistema de información se han separado en módulos y sub módulos, a los cuales, los usuarios pueden tener acceso desde los diferentes dispositivos desde los que vayan a iniciar sesión.

La figura 4.18 muestra la definición de la entidad “Periodos” la cual representa al concepto de “Campañas” y es el inicio de una campaña la que da origen a la existencia de procesos de producción del langostino blanco, por ello, en un periodo se registra un identificador correlativo, su código, una descripción breve, su fecha de inicio y de fin. En un periodo (campaña), se compran muchos lotes de larvas de langostino y, a su vez, un periodo da origen a muchas “Fases” que representan el registro de una “Etapa” del proceso de producción en una campaña determinada, por ello, la entidad “Fases” tiene como atributos a un identificador correlativo de fase, al identificador de una etapa, al identificador de un periodo, la fecha de inicio y una bandera que identifica si la fase ha termino o no.

El proceso de producción, en sus diferentes “Etapas”, hace uso de “Piscinas” de cultivo adscritas a diferentes “Sectores”, así, una “Fase” tiene una o muchas “Piscinas” y una “Piscina” puede pertenecer, en el tiempo, a una o muchas “Fases”, es por ello que en el modelo se ha definido una entidad que permitirá registrar de mejor manera esta relación y se denomina “FasesPiscinas”. La piscina cuenta con los siguientes atributos: un identificador correlativo de la piscina, el identificador del sector al que pertenece, un código alfanumérico único, el nombre que se le da usualmente a la piscina, su superficie, una bandera booleana que define si la piscina está activa o no y una breve descripción textual que se quiera dar a la piscina.

Toda piscina, cuando está en uso en cierta fase de producción, será cargada con uno o muchos lotes de larvas, esta relación se registra en la entidad “CargaPiscinas” cuyos atributos son: un identificador correlativo, el identificador del lote, el identificador de la fase y la piscina (entidad “FasesPiscinas”), la cantidad de larvas, el peso, la densidad y la fecha en la que se registró el ingreso de la biomasa que el lote aporta en la carga, finalmente, está el identificar del usuario que ha registrado la carga y la fecha de inserción del registro.

Para el seguimiento del proceso productivo, se han definido “Parámetros”, los cuales son medidos y generan muchos “Valores” cuyos atributos son: identificador correlativo del valor, identificador del parámetro medido, identificador del proveedor que ha hecho la medición, identificador de la fase y la piscina (entidad “FasesPiscinas”) donde ha sido hecha la medición, el dato del valor medido, comentarios a la medición, fecha de registro del valor (que es la fecha en la que se ha medido), identificador del usuario que inserta el

registro y la fecha y hora de ingreso del registro en el modelo. Vale señalar que esta entidad admite que no se registre un proveedor dado que puede ser la misma empresa propietaria la que lo hace a través de su personal, a su vez, también se permite la no inserción de un identificador de fase y piscina, dado que el parámetro a medir puede no pertenecer a una piscina, como por ejemplo: la temperatura ambiental.

El sistema de información requiere identificar si en cierta fase, el valor de un parámetro está o no dentro de lo esperado, ante esto, el modelo define la entidad “Límites” con los atributos: identificador correlativo del límite, identificador del parámetro, identificador de la fase, fecha de inicio de aplicación del límite, fecha de finalización de la aplicación del límite, valor máximo y valor mínimo permitidos. Con esta definición de “Límites”, se da origen a la creación de “Alertas” que tiene los siguientes atributos: identificador correlativo de alertas, identificador del límite que ha sido quebrantado, identificador del valor que quebranta el límite y una bandera que indica si esta alerta ya ha sido vista o no.

Finalmente, la entidad “Usuarios”, además de relacionarse con las entidades vistas en la figura 4.17, también lo hace con: “Valores”, “Lotes”, “Fases” y “CargaPiscinas” para definir atributos de auditoría que permitan conocer al usuario que ha insertado o editado un registro en el modelo. Como se observa, estas relaciones no se han mostrado en los diagramas para mantener un orden y evitar confusión en la visualización de los mismos.

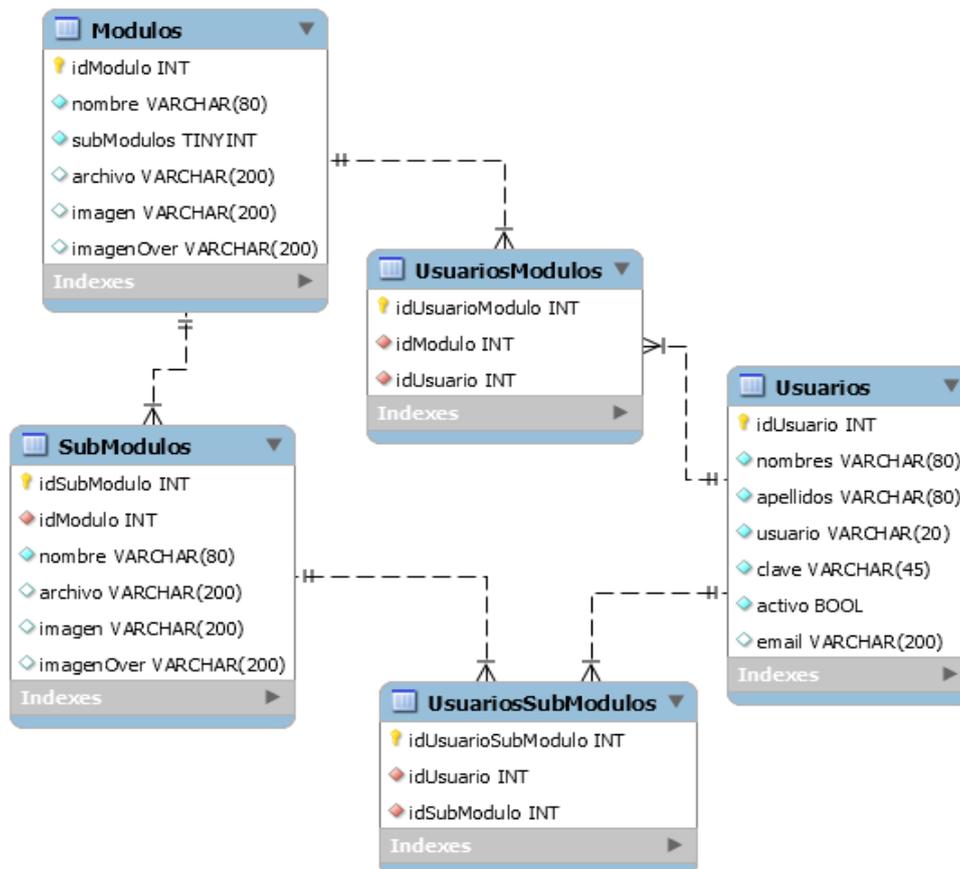


Figura 4.17 Modelo Entidad Relación (parte 1)

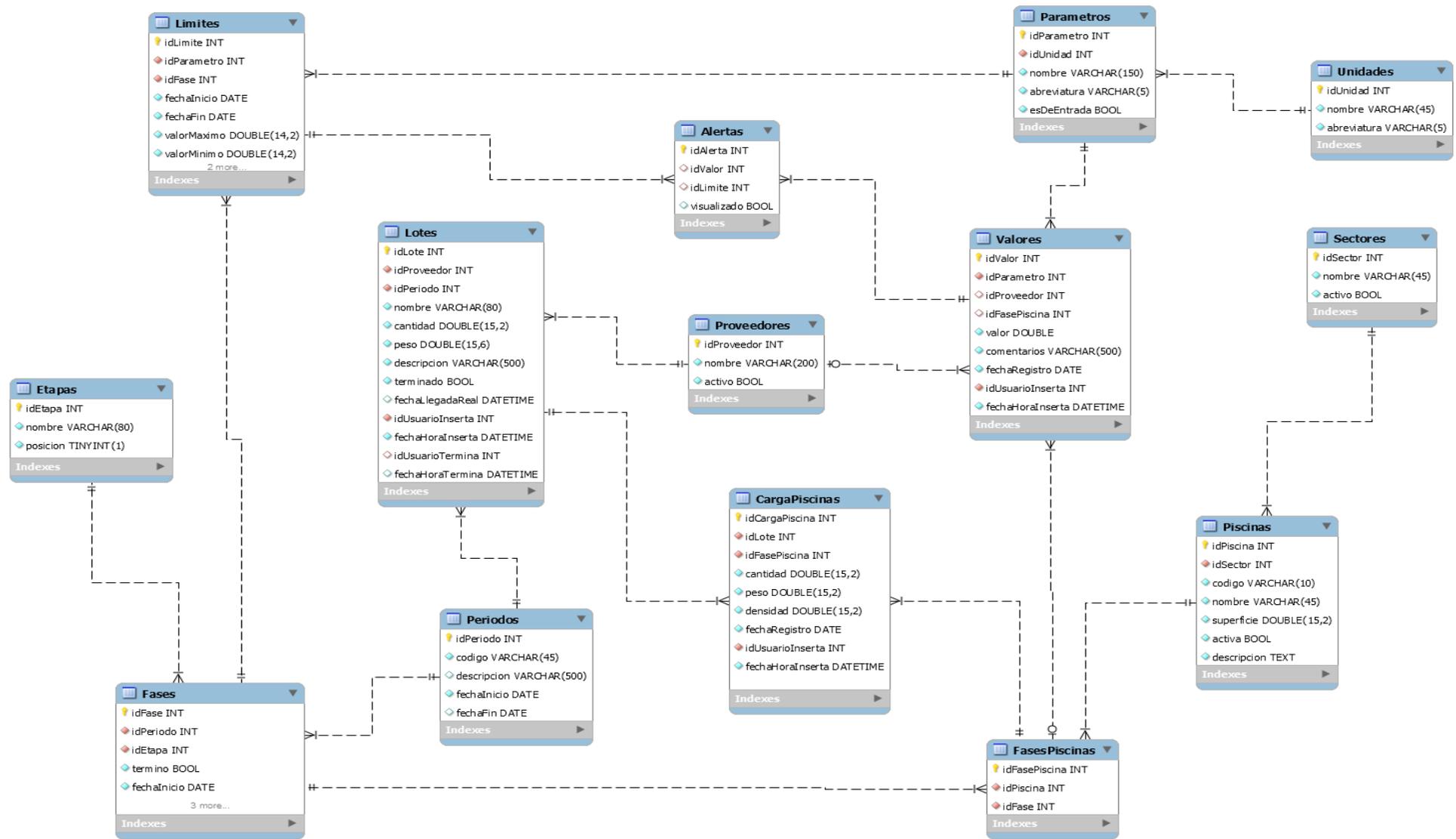


Figura 4.18 Modelo Entidad Relación (parte 2)

4.5.2 Diccionario de datos.

Alertas			
En esta tabla se almacenarán los registros de alerta que el sistema de información detecte cuando el valor de un parámetro ha salido de los límites establecidos para el mismo.			
Nombre de columna	Tipo de dato	PK	Comentario
idAlerta	INT	✓	Identificador de la alerta.
idValor	INT		Identificador del valor fuera del límite establecido para su correspondiente parámetro en la fase respectiva.
idLimite	INT		Identificador del límite trasgredido.
visualizado	TINYINT(1)		Define si la alerta ya ha sido vista o no.

CargaPiscinas			
Esta tabla contendrá los registros de aporte de un lote a una piscina en una determinada fase del proceso productivo.			
Nombre de columna	Tipo de dato	PK	Comentario
idCargaPiscina	INT	✓	Identificador de la carga de un lote a una piscina en una fase.
idLote	INT		Identificador del lote origen.
idFasePiscina	INT		Identificador de la fase que se está desarrollando en determinada piscina.
cantidad	DOUBLE(15,2)		Número que indica la cantidad estimada de larvas que aporta el lote a una determinada fase.
peso	DOUBLE(15,2)		Peso estimado del aporte del lote a la fase.
densidad	DOUBLE(15,2)		Cantidad estimadas de larvas por unidad de volumen que aporta el lote de origen a la fase.
fechaRegistro	DATE		Fecha en que se registra el ingreso del aporte de un lote a una fase.
idUsuarioInserta	INT		Identificador del usuario que ingresa el registro.
fechaHoraInserta	DATETIME		Fecha y hora de ingreso del registro.

Etapas			
Contiene el listado de etapas de un proceso de producción de langostino blanco			
Nombre de columna	Tipo de dato	PK	Comentario
idEtapa	INT	✓	Identificador de la etapa.
nombre	VARCHAR(80)		Nombre de la etapa.
posicion	TINYINT(1)		Valor ordinal de la posición de una etapa dentro de un proceso de producción de langostino blanco.

Fases			
Contiene los registros de las etapas que se van desarrollando en cierta campaña, en momentos determinados.			
Nombre de columna	Tipo de dato	PK	Comentario
idFase	INT	✓	Identificador de la fase.
idPeriodo	INT		Identificador de la campaña en que se ubica la fase.
idEtapa	INT		Identificador de la etapa en la que se ubica la fase.
termino	TINYINT(1)		Define si la fase ha concluído o no.
fechaInicio	DATE		Fecha de inicio de la fase.
fechaFin	DATE		Fecha de culminación de la fase.
idUsuarioInserta	INT		Identificador del usuario que ingresa el registro.
fechaHoraInserta	DATETIME		Fecha y hora de ingreso del registro.

FasesPiscinas			
En esta tabla se almacena los registros de la asignación de piscinas a las distintas fases registradas en el sistema de información			
Nombre de columna	Tipo de dato	PK	Comentario
idFasePiscina	INT	✓	Identificador de la piscina asignada a la fase.
idPiscina	INT		Identificador de la piscina.
idFase	INT		Identificador de la fase.

Limites			
En esta tabla se almacenan los límites asignados a un parámetro en cierta fase de producción del langostino blanco.			
Nombre de columna	Tipo de dato	PK	Comentario
idLimite	INT	✓	Identificador del límite que se desea definir.
idParametro	INT		Identificador del parámetro al que se le establecerá valores límite.
idFase	INT		Identificador de fase en la que se definirá un límite.
fechaInicio	DATE		Fecha en la que se inicia la validez del límite.
fechaFin	DATE		Fecha en que termina la validez del límite.
valorMaximo	DOUBLE(14,2)		Valor numérico máximo que puede tener el límite.
valorMinimo	DOUBLE(14,2)		Valor numérico mínimo que puede tener el límite.
valorMedio	DOUBLE(14,2)		Valor numérico medio que se desea para el límite.
fechaRegistro	DATE		Fecha en la que se ingresó el registro.

Lotes			
Esta tabla contiene los registros de los lotes adquiridos a diversos proveedores			
Nombre de columna	Tipo de dato	P K	Comentario
idLote	INT	✓	Identificador del lote.
idProveedor	INT		Identificador del proveedor del lote.
idPeriodo	INT		Identificador de campaña de ingreso del lote.
nombre	VARCHAR(80)		Nombre asignado al lote.
cantidad	DOUBLE(15,2)		Número estimado de larvas que trae el lote.
peso	DOUBLE(15,6)		Peso registrado a la llegada del lote.
descripcion	VARCHAR(500)		Descripción que se desea otorgar al lote.
terminado	TINYINT(1)		Define si el lote se uso en su totalidad o no.
fechaLlegadaReal	DATETIME		Fecha de ingreso del lote a la empresa.
idUsuarioInserta	INT		Identificador del usuario que ingresa el registro.
fechaHoraInserta	DATETIME		Fecha y hora de inserción del registro.
idUsuarioTermina	INT		Identificador del usuario que define que el lote ya ha sido utilizado en su totalidad.
fechaHoraTermina	DATETIME		Fecha y hora en la que se registró que el lote ha sido utilizado en su totalidad.

Modulos			
Esta tabla guarda nombres de las funcionalidades maestras del sistema de información.			
Nombre de columna	Tipo de dato	PK	Comentario
idModulo	INT	✓	Identificador de un módulo de un software.
nombre	VARCHAR(80)		Nombre del módulo del software.
subModulos	TINYINT		Define si el módulo tiene o no submódulos.
archivo	VARCHAR(200)		Guarda ruta de archivo identificativo del módulo.
imagen	VARCHAR(200)		Guarda ruta de imagen identificativa del módulo.
imagenOver	VARCHAR(200)		Guarda ruta de imagen identificativa del módulo cuando un usuario pase el cursor sobre el mismo.

Parametros			
En esta tabla se encontrarán registrados los parámetros que son utilizados para hacer el seguimiento y control de la operación			
Nombre de columna	Tipo de dato	PK	Comentario
idParametro	INT	✓	Identificador del parámetro.
idUnidad	INT		Identificador de unidad de medida del parámetro.
nombre	VARCHAR(150)		Nombre del parámetro.
abreviatura	VARCHAR(5)		Nombre abreviado del parámetro.
esDeEntrada	TINYINT(1)		Dice si el parámetro es de entrada o de salida del proceso.

Periodos			
Esta tabla guarda datos de las campañas de producción en un momento determinado.			
Nombre de columna	Tipo de dato	PK	Comentario
idPeriodo	INT	✓	Identificador de la campaña de producción.
codigo	VARCHAR(45)		Código de la campaña de producción.
descripcion	VARCHAR(500)		Texto relevante sobre la campaña de producción.
fechaInicio	DATE		Fecha de inicio de la campaña de producción.
fechaFin	DATE		Fecha en que finaliza la campaña de producción.

Piscinas			
Esta tabla contiene los datos de las piscinas dedicadas al proceso de producción.			
Nombre de columna	Tipo de dato	PK	Comentario
idPiscina	INT	✓	Identificador de la piscina receptora de larvas.
idSector	INT		Identificador del sector al que pertenecen las piscinas.
codigo	VARCHAR(10)		Código identificador de la piscina.
nombre	VARCHAR(45)		Nombre de la piscina.
superficie	DOUBLE(15,2)		Valor en metros cuadrados de la superficie de la piscina.
activa	TINYINT(1)		Define si la piscina está o no operativa para la recepción de larvas.
descripcion	TEXT		Información textual relevante sobre la piscina.

Proveedores			
Esta tabla contiene el listado de los proveedores de la empresa.			
Nombre de columna	Tipo de dato	PK	Comentario
idProveedor	INT	✓	Identificador de un proveedor de productos o servicios propios del proceso de producción.
nombre	VARCHAR(200)		Nombre del proveedor de productos o servicios.
activo	TINYINT(1)		Define si el proveedor sigue o no ofreciendo productos o servicios.

Sectores			
Esta tabla registra los sectores de piscinas que administra la empresa.			
Nombre de columna	Tipo de dato	PK	Comentario
idSector	INT	✓	Identificador del sector de piscinas usadas para el proceso de producción.
nombre	VARCHAR(45)		Nombre asignado al sector de piscinas.
activo	TINYINT(1)		Define si el sector de piscinas está o no apto para la operación dentro del proceso de producción.

SubModulos

Esta tabla registra las funcionalidades que están contenidas dentro de los módulos del sistema de información.

Nombre de columna	Tipo de dato	PK	Comentario
idSubModulo	INT	✓	Identificador de un submódulo del sistema.
idModulo	INT		Identificador del módulo al que pertenece el submódulo definido.
nombre	VARCHAR(80)		Nombre asignado al submódulo.
archivo	VARCHAR(200)		
imagen	VARCHAR(200)		
imagenOver	VARCHAR(200)		

Unidades

En esta tabla se encuentran las unidades de medida utilizadas para los valores de los parámetros.

Nombre de columna	Tipo de dato	PK	Comentario
idUnidad	INT	✓	Identificador de la unidad de medida.
nombre	VARCHAR(45)		Nombre de la unidad de medida.
abreviatura	VARCHAR(5)		Abreviatura de la unidad de medida.

Usuarios

En esta tabla se registran los datos de los usuarios del sistema de información.

Nombre de columna	Tipo de dato	PK	Comentario
idUsuario	INT	✓	Identificador del usuario del software.
nombres	VARCHAR(80)		Nombres del usuario del software.
apellidos	VARCHAR(80)		Apellidos del usuario del software.
usuario	VARCHAR(20)		Nombre de usuario para iniciar sesión en software.
clave	VARCHAR(45)		Clave de acceso encriptada al software.
activo	TINYINT(1)		Define si el usuario puede o no acceder al software.
email	VARCHAR(200)		Correo electrónico del usuario del software.

UsuariosModulos

Esta tabla almacena los registros que definen a que módulos pueden acceder los usuarios.

Nombre de columna	Tipo de dato	PK	Comentario
idUsuarioModulo	INT	✓	Identificador del módulo al que tiene acceso un usuario dentro del software.
idModulo	INT		Identificador del módulo del software.
idUsuario	INT		Identificador del usuario del software.

UsuariosSubModulos

Esta tabla almacena los registros que definen a que sub módulos puede acceder un usuario.

Nombre de columna	Tipo de dato	PK	Comentario
idUsuarioSubModulo	INT	✓	Identificador del submódulo del software al cual tiene acceso un usuario.
idUsuario	INT		Identificador del usuario del software.
idSubModulo	INT		Identificador del submódulo del software.

Valores

Esta tabla almacena los valores resultantes de la medición de los parámetros de la operación en un tiempo determinado.

Nombre de columna	Tipo de dato	PK	Comentario
idValor	INT	✓	Identificador de un valor de un parámetro del proceso de producción de langostino blanco.
idParametro	INT		Identificador del parámetro medido o calculado en el proceso de producción.
idProveedor	INT		Identificador del proveedor que ha medido o calculado el valor de un parámetro. Si lo ha hecho personal de la misma empresa, este valor puede ser nulo.
idFasePiscina	INT		Identificador de la piscina y la fase en la que ha sido medido o calculado el valor del parámetro. Si este dato es de un parámetro que no es medido en una piscina, puede asignársele el valor nulo.
valor	DOUBLE		Valor numérico medido o calculado para un parámetro del proceso productivo.
comentarios	VARCHAR(500)		Descripción textual de información importante de la medida ingresada.
fechaRegistro	DATE		Fecha en la que se tomó o calculó el valor del parámetro de producción.
idUsuarioInserta	INT		Identificador del usuario que inserta el registro en el sistema.
fechaHoraInserta	DATETIME		Fecha de inserción del registro en el sistema.

Capítulo 5

Conclusiones y recomendaciones

5.1 Conclusiones.

- Dadas sus características funcionales del sistema de información, se concluye que el mismo, será útil como primer modelo para la automatización de los procesos de toma de datos de aquellas empresas productoras de langostino blanco que, en la actualidad, llevan a cabo los mencionados procesos de forma manual o mínimamente informatizada (con hojas de cálculo electrónicas).
- Por sus características funcionales, el entorno de su operación y la utilidad del sistema de información, se concluye que la implementación básica del portal web y sus correspondientes servicios, sea bajo una arquitectura física de cliente-servidor, por lo cual, en un sólo servidor físico se ejecutaran simultáneamente los siguientes programas: el servidor de páginas web, el servidor de aplicaciones, el servidor de base de datos, el servidor de almacenamiento de archivos, así como también los servicios necesarios para la operación y mantenimiento de los programas antes mencionados.
- Como resultado del análisis y diseño realizados, se determinó que la arquitectura de software a implementar para el sistema de información, será la modelo-vista-controlador debido a sus ventajas en cuanto a ordenamiento y reutilización del código, así como también, a la mantenibilidad del mismo.
- Debido a las ventajas que ofrece la arquitectura física y lógica a implementar, se concluye que con ello se ofrece la estructura básica y confiable de relación y comunicación con otras herramientas especializadas en análisis de datos o con otros sistemas informáticos que necesiten ser provistos de los datos del sistema informático materia de este trabajo.
- Las gráficas de control a implementar en el sistema informático son un primer acercamiento al estudio de los parámetros de la producción del langostino blanco, por lo que, su comprensión y adopción son de fiable utilidad, tanto para los analistas de datos, como para el personal de supervisión de la producción.
- Por lo analizado en el estudio de factibilidad económica, se concluye que la implementación del proyecto contará con 2 analistas programadores, durará 7.31 meses y costará S/. 72,655.57. En este último ítem, está incluido el salario de los analistas programadores, la compra de equipos de cómputo y la adquisición de 2

licencias de uso de Microsoft Office Home Edition; más no incluye la compra ni de software de servidor, ni de software de estación de trabajo, puesto que estos son de disponibilidad gratuita. Si la empresa que decide implementar el proyecto ya cuenta con los recursos técnicos y de personal, el presupuesto sólo incluiría el salario de los analistas programadores, por tanto, se reduciría a S/. 67,415.57.

- El estudio de factibilidad que se ha realizado en este trabajo, concluye que el proceso, a nivel técnico, operativo y económico es factible de realizar.

5.2 Recomendaciones.

- Basado en la información recibida por los especialistas de la Universidad de Piura, se hace necesario recomendar el inicio de un proyecto de implementación de un sistema de calidad basado en procesos, el cual, -entre otros beneficios- uniformice los procedimientos de toma de datos en la operación.
- Debido al ordenamiento, consistencia y confiabilidad que generará la implementación del sistema informático propuesto, se recomienda la planificación de un proyecto de análisis de datos que permita el estudio profundo de las variables de operación con el fin de alcanzar, no solamente un control de las mismas, sino también la optimización en el uso de los recursos con los que pueda disponer una empresa.

Bibliografía

1. **ANGULO BUSTÍOS, C.** (2005). *Estadística. Segunda Edición*. Piura: Universidad de Piura.
2. **CASS, S.** (Julio de 2016). *The 2016 Top Programming Languages*. Visto en Mayo de 2017, de Spectrum IEEE.
<http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>
3. **CONALLEN, J.** (2003). *Building Web Applications with UML, 2nd Edition*. Addison-Wesley Professional.
4. **CUBILLOS, C.** (s.f.). *Arquitectura Cliente/Servidor*. Visto en Agosto de 2016, de Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso (Chile):
<http://ocw.pucv.cl/cursos-1/arquitectura-de-sistemas-de-software/materiales-de-clases/web-cliente-servidor>
5. **HISTCHFELD K., N., & SALINAS CARO, P.** (s.f.). *Tutorial de UML*. Visto en Abril de 2016, de Universidad de Chile, Departamento de Ciencias de la omputaciòn:
<http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>
6. **HURTADO JARA, O.** (s.f.). Apuntes del curso Análisis y Diseño de Sistemas. Programa académico Ingeniería Industrial y de Sistemas, Universidad de Piura.
7. **JOSHI , K.** (s.f.). *Introduction to Information Systems and Their Capabilities*. Visto en Marzo de 2016, de University of Missouri - St. Louis:
<http://www.umsl.edu/~joshik/msis480/chapt01.htm>
8. **LABORATORIO DEL DEPARTAMENTO DE INFORMÁTICA - UNIVERSIDAD CARLOS III DE MADRID.** (s.f.). *Modelo-Vista-Controlador*. Visto en Setiembre de 2016, de Laboratorio del Departamnto de Informática - Universidad Carlos III de Madrid:
<http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>
9. **LÓPEZ GAONA, A.** (s.f.). *El modelo Entidad-Relación*. Visto en Agosto de 2016, de Posgrado en Ciencia e Ingeniería de la Computación, Facultad de Ciencias, Universidad Nacional Autónoma de México:
<http://hp.fciencias.unam.mx/~alg/bd/er.pdf>

10. **MICROSOFT.** (s.f.). *Model-View-Controller*. Visto en Mayo de 2016, de Microsoft Developer Network:
<https://msdn.microsoft.com/en-us/library/ff649643.aspx>
11. **MICROSOFT.** (s.f.). *N-Tier Data Applications Overview*. Visto en Abril de 2016, de Microsoft:
<https://msdn.microsoft.com/en-us/library/bb384398.aspx>
12. **MICROSOFT PATTERNS & PRACTICES.** (Octubre de 2009). *Microsoft Application Architecture Guide, 2nd Edition*. Visto en Mayo de 2016, de Microsoft:
<https://msdn.microsoft.com/en-us/library/ff650706.aspx>
13. **SANCHEZ PLA, A.** (s.f.). *Introducción a la biocomputación*. Visto en Marzo de 2016, de Universidad de Barcelona:
http://www.ub.edu/stat/docencia/bioinformatica/introbiocomputacio/ServidoresWeb/ServidoresWeb-Concepto_Configuracion_Uso.pdf
14. **WELICKI, L.** (s.f.). *Patrones y Antipatrones: una Introducción - Parte II*. Visto en Mayo de 2016, de Microsoft Developer Network:
<https://msdn.microsoft.com/es-es/library/bb972251.aspx>
15. **WIESSE LÓPEZ, F.** (s.f.). Apuntes del curso Base de datos I. Escuela Tecnológica Superior, Universidad de Piura.
16. **WIKIPEDIA.** (s.f.). *Mobile app*. Visto en Abril de 2016, de Wikipedia, the free encyclopedia:
https://en.wikipedia.org/wiki/Mobile_app#Development
17. **ZACARÍAS VÉLEZ, C.** (Febrero de 2012). Diseño de portal Web para la búsqueda de información sobre productos ofrecidos por negocios aledaños a la Universidad de Piura . *Tesis para optar el grado de Ingeniero Industrial y de Sistemas*. Piura: Facultad de Ingeniería, Universidad de Piura.