

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS.

**DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LOS
PROCESOS DE COSECHA Y POST-COSECHA DE LA
CAMARONERA “PAMPAS DE CAYANCA”**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**EDGAR GEOVANNY RÍOS PINZÓN
georipi@hotmail.com**

**WILSON HOMERO SUNTAXI LLUMIQUINGA
whomers@gmail.com**

**DIRECTOR: ING. PAÚL VILCA
pavich_2500@hotmail.com**

Quito, Septiembre 2008

DECLARACIÓN

Nosotros, Edgar Geovanny Ríos Pinzón y Wilson Homero Suntaxi Llumiquinga declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Edgar Geovanny Ríos Pinzón

Wilson Homero Suntaxi
Llumiquinga

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Edgar Geovanny Ríos Pinzón y Wilson Homero Suntaxi Llumiquinga, bajo mi supervisión.

Ing. Paúl Vilca
DIRECTOR DE PROYECTO

AGRADECIMIENTO

Agradezco primeramente a Dios por ser quien iluminó mi sendero y me permitió llegar hasta este momento.

A mis padres por ser mis mejores amigos, mi fortaleza, por darme todo lo que tengo y no dejarme caer nunca. Por enseñarme que la perseverancia y el esfuerzo son el mejor camino para alcanzar lo que anhelas. Por ser mi inspiración para alcanzar mis metas, por enseñarme que todo se aprende y que todo esfuerzo es al final recompensa, por todas aquellas enseñanzas que al final se convirtieron en su triunfo y el mío.

A mis dos hermanos por el apoyo incondicional que me brindan a cada instante y cuando lo necesito, haciendo invisible la distancia, por cada consejo y palabras que me acompañan en cada paso que tengo que dar.

A la casualidad de la vida que me otorgó el encontrarme con verdaderos amigos quienes aparecieron para darme la mano y ayudarme a levantar después de cada caída, por su confianza, su lealtad y cada palabra de aliento.

A mi Director de Tesis, Ing. Paúl Vilca, por su generosidad al brindarme la oportunidad de recurrir a su capacidad y experiencia científica en un marco de confianza, afecto y amistad, fundamentales para la realización de este trabajo.

A cada uno de los maestros que participaron en mi desarrollo profesional durante mi carrera, sin su ayuda y conocimientos.

Y en general a todas aquellas personas que de una u otra manera son parte de este triunfo, de lograr alcanzar mi culminación académica, la cual es el anhelo de todos los que así lo deseamos.

Geovanny

AGRADECIMIENTO

Agradezco a Dios, por los padres que me ha dado.

Gracias a mis padres, que me guían con sus consejos y me ayudan a levantarme cada vez que he tropezado, por todo lo que hacen y continúan haciendo por mis infinitamente gracias.

.

A mis hermanos por ser los mejores, estar siempre presentes cuando los necesito y tener creer en mí.

A mis abuelitos, tíos, primos que siempre me han cuidado, ayudado y acompañado en cada una de las etapas de mi vida.

Al Ing. Paul Vilca, por su colaboración y guía durante todo el proceso de desarrollo de este proyecto, y a mis amigos que han sabido ofrecerme su amistad gracias.

Wilson

DEDICATORIA

Le dedico este proyecto a mis Padres, quienes dedicaron todo su tiempo y su esfuerzo para que sea posible.

A mi madre, más que una dedicación, esta obra es tuya por ser quien más lucho junto a mí y quien puso el don de inspiración para lograrla.

A mi padre, por sus consejos que me permitieron ser fuerte y no decaer ante la adversidad.

A mis hermanos, por tender sus manos cuando pensaba que no había solución para algún problema. A ambos este esfuerzo personal les dedico.

A todos aquellos amigos que estuvieron, los que están y los que seguirán aquí.

Geovanny

DEDICATORIA

Dedico la realización de este proyecto a mi familia, que con su paciencia, y dedicación me han ayudado durante su desarrollo, en especial a mis amados padres Francisco Suntaxi, María Llumiquinga; quienes me han brindado su amistad y me han apoyado incondicionalmente en toda mi vida; a mis profesores, quienes compartieron sus conocimientos; a mis hermanos: Klever, Renato, Marcelo y también a mis amigos, quienes me apoyaron dándome ánimo para poder seguir adelante en los estudios, todos estos consejos me han servido y servirán para poder salir adelante en mi vida.

Wilson

INDICE

INDICE.....	VII
INTRODUCCION.....	1
CAPÍTULO 1 PLANTEAMIENTO DEL PROBLEMA	3
1.1 DESCRIPCIÓN DEL PROBLEMA.....	3
1.1.1 LA EMPRESA.....	3
1.1.2 SITUACIÓN ACTUAL DE LA EMPRESA.....	3
1.1.3 CONCEPTUALIZACIÓN DE LOS PROCESOS DE COSECHA Y POST-COSECHA	5
1.1.3.1 Cosecha	5
1.1.3.2 Post-Cosecha	6
1.1.4 SITUACIÓN ACTUAL DE LOS PROCESOS DE COSECHA Y POST-COSECHA.....	6
1.2 DETERMINACIÓN DE LA METODOLOGÍA DE DESARROLLO A APLICAR....	7
1.2.1 METODOLOGÍA	7
1.2.1.1 Definición.....	7
1.2.1.2 Tendencia.....	7
1.2.1.2.1 Metodologías Ágiles	7
1.2.1.2.2 Metodología Tradicional	8
1.2.2 METODOLOGIA A UTILIZARSE	9
1.2.2.1 Rup (Rational Unified Process)	10
1.2.2.2 Xp (eXtreme Programming)	11
1.2.2.3 Rad (Rapid Application Development)	12
1.2.2.4 Msf (Microsoft Solution Framework)	13
1.2.2.5 Evaluación de Metodologías	14
1.2.3 SELECCIÓN DE LA METODOLOGÍA	15
1.3 DETERMINACIÓN DE LA HERRAMIENTA A UTILIZAR	16
1.3.1 DESCRIPCIÓN DE HERRAMIENTAS TIPO FRONT END Y BACK END....	16
1.3.1.1 Front-End	16
1.3.1.1.1 Visual Studio .Net 2005 C#	17
1.3.1.1.2 Eclipse Java	17
1.3.1.1.3 JBuilder	18
1.3.1.2 Back-End	19
1.3.1.2.1 MySql	19
1.3.1.2.2 Oracle 10g	20
1.3.1.2.3 Microsoft SQL Server Express Edition 2005	21
1.3.1.3 Revisión de Herramientas.....	23
1.3.2 SELECCIÓN DE LA HERRAMIENTA DE MODELADO.. ¡Error! Marcador no definido.	
CAPÍTULO 2 DESARROLLO DEL SISTEMA INFORMATICO PARA LOS PROCESOS DE COSECHA Y POST-COSECHA DE LA CAMARONERA “PAMPAS DE CAYANCA”.....	26
2.1 ANALISIS.....	26
2.1.1 ESPECIFICACIÓN DE REQUERIMIENTOS	26
2.1.1.1 Documento de Visión.....	26
2.1.1.1.1 Propósito.....	26
2.1.1.1.2 Alcance	26
2.1.1.2 Posicionamiento	27
2.1.1.2.1 Oportunidad de Negocio.....	27
2.1.1.2.2 Sentencia que define el problema	27
2.1.1.2.3 Sentencia que define la posición del Producto	28
2.1.1.3 Descripción de Participantes y Usuarios	29

2.1.1.3.1	Resumen de los Interesados en el proyecto.....	29
2.1.1.3.2	Resumen de Usuarios.....	30
2.1.1.3.3	Perfiles de usuario.....	31
2.1.1.4	Resumen Del Producto.....	32
2.1.1.4.1	Perspectiva del Producto.....	32
2.1.1.4.2	Supuestos y dependencias.....	34
2.1.1.4.3	Licenciamiento e Instalación.....	34
2.1.2	ESPECIFICACIONES SUPLEMENTARIAS	35
2.1.2.1	Introducción	35
2.1.2.1.1	Propósito	35
2.1.2.1.2	Alcance	35
2.1.2.2	Funcionalidad	35
2.1.2.2.1	Registro de usuarios del sistema.....	35
2.1.2.2.2	Generar Históricos	35
2.1.2.2.3	Calculo Cotización.....	36
2.1.2.2.4	Registros de liquidaciones.....	36
2.1.2.3.1	Capacitación a los usuarios del sistema.....	36
2.1.2.3.2	Facilidad de uso	36
2.1.2.3.3	Presentación Visual.....	36
2.1.2.4	Confiabilidad	37
2.1.2.4.1	Disponibilidad	37
2.1.2.4.2	Tiempo transcurrido para mantenimiento	37
2.1.2.5.1	Tiempo de respuesta.....	37
2.1.2.5.2	Carga de Trabajo	37
2.1.2.5.3	Degradación.....	38
2.1.2.5.4	Utilización de recursos	38
2.1.2.6.1	Lenguaje de modelado UML.....	38
2.1.2.7.1	Interfaces de Software.....	38
2.1.2.7.2	Interfaces de Comunicación	38
2.1.3	MODELO DEL NEGOCIO	39
2.1.4	CASOS DE USO	39
2.1.4.1	Diccionarios de actores.....	40
2.1.4.2	Descripción de Casos de Uso	41
2.1.4.2.1	Gestionar Especie	41
2.1.4.2.2	Gestionar Piscina	43
2.1.4.2.3	Gestionar Empacadora.....	45
2.1.4.2.4	Gestionar Cotización	47
2.1.4.2.5	Gestionar Cosecha.....	49
2.1.4.2.6	Gestionar Despacho.....	52
2.1.4.2.7	Gestionar Liquidación.....	54
2.1.4.2.8	Generar Histórico	56
2.1.4.2.9	Gestionar Usuario	57
2.1.5	DIAGRAMAS DE COLABORACION.....	59
2.1.5.1	Diagrama de Colaboración del Caso de Uso Gestionar Especie.....	59
2.1.5.2	Diagrama de Colaboración del Caso de Uso Gestionar Piscina.....	61
2.1.5.3	Diagrama de Colaboración del Caso de Uso Gestionar Empacadora. ..	63
2.1.5.4	Diagrama de Colaboración del Caso de Uso Gestionar Cotización.....	65
2.1.5.5	Diagrama de Colaboración del Caso de Uso Gestionar Cosecha.	67
2.1.5.6	Diagrama de Colaboración del Caso de Uso Gestionar Despacho.	69
2.1.5.7	Diagrama de Colaboración del Caso de Uso Gestionar Liquidación.	71
2.1.5.8	Diagrama de Colaboración del Caso de Uso Generar Histórico.....	73
2.1.5.9	Diagrama de Colaboración del Caso de Uso Gestionar Usuario.....	74
2.1.6	PAQUETE DE ANÁLISIS	76
2.1.7	SUBSISTEMAS DEL MÓDULO DEL SISTEMA.	77

2.2	DISEÑO	78
2.2.1	DISEÑO DE CLASES.....	79
2.2.2	DISEÑO DE CASOS DE USO.....	80
2.2.2.1	Subsistema Gestionar Especie.....	80
2.2.2.2	Subsistema Gestionar Piscina.....	81
2.2.2.3	Subsistema Gestionar Empacadora.....	82
2.2.2.4	Subsistema Gestionar Cotización.....	83
2.2.2.5	Subsistema Gestionar Cosecha.....	84
2.2.2.6	Subsistema Gestionar Despacho.....	85
2.2.2.7	Subsistema Gestionar Liquidación.....	86
2.2.2.8	Subsistema Generar Histórico.....	86
2.2.2.9	Subsistema Gestionar Usuario.....	87
2.2.3	DISEÑO DE INTERFACES	87
2.2.3.1	Mapa de Navegación	88
2.2.3.2	Estructura de páginas.....	89
2.2.4	ARQUITECTURA DEL SISTEMA.....	92
2.2.5	MODELO DE DESPLIEGUE.....	94
2.3	IMPLEMENTACION	94
2.3.1	MODELO DE IMPLEMENTACION	95
2.3.2	IMPLEMENTACIÓN DE LA BASE DE DATOS.....	96
2.3.2.2	Modelo Conceptual de la base de datos	98
2.3.2.3	Modelo Físico de la base de datos.....	99
2.3.3	ESTÁNDARES DE PROGRAMACIÓN.....	100
2.3.3.1	Estándares de Controles.....	101
2.3.3.2	Estándares de diseño de la base de datos	102
2.4	PRUEBAS	104
2.4.1	CASOS Y PROCEDIMIENTOS DE PRUEBAS.....	104
2.4.1.1	Caso de Prueba Inicio de Sesión de Usuario (Ingreso Correcto)	105
2.4.1.2	Caso de Prueba Inicio de Sesión de Usuario (Ingreso Fallido).....	105
2.4.1.3	Caso de Prueba Crear Especie.....	106
2.4.1.4	Caso de Prueba Modificar Especie	107
2.4.1.5	Caso de Prueba Eliminar Especie (Datos Correctos).....	108
2.4.1.6	Caso de Prueba Crear Piscina.....	109
2.4.1.7	Caso de Prueba Modificar Piscina	110
2.4.1.8	Caso de Prueba Eliminar Piscina (Datos Correctos).....	111
2.4.1.9	Caso de Prueba Crear Empacadora	111
2.4.1.10	Caso de Prueba Modificar Empacadora.....	113
2.4.1.11	Caso de Prueba Eliminar Empacadora (Datos Correctos)	114
2.4.1.12	Caso de Prueba Crear Cotización.....	115
2.4.1.13	Caso de Prueba Modificar Cotización	116
2.4.1.14	Caso de Prueba Eliminar Cotización (Datos Correctos)}.....	117
2.4.1.15	Caso de Prueba Crear Cosecha	118
2.4.1.16	Caso de Prueba Modificar Cosecha.....	119
2.4.1.17	Caso de Prueba Eliminar Cosecha (Datos Correctos)	121
2.4.2	EVALUACION DE RESULTADOS.....	122
2.4.2.1	Herramientas	122
2.4.2.2	Rendimiento.....	122
2.4.2.3	Utilidad.....	122
2.4.2.4	Manejo.....	122
2.4.2.5	Consistencia	123
CAPÍTULO 3 CONCLUSIONES Y RECOMENDACIONES		125
3.1	CONCLUSIONES.....	125
3.2	RECOMENDACIONES.....	127

BIBLIOGRAFÍA.....	128
ANEXOS.....	132

INDICE DE TABLAS

TABLA 1. 1 CRITERIOS SELECCIÓN DE METODOLOGÍA.....	14
--	----

TABLA 1. 2 PONDERACIÓN DE CRITERIOS SELECCIÓN DE METODOLOGÍA	15
TABLA 1. 3 DIFERENCIAS ENTRE HERRAMIENTAS FRONT-END	23
TABLA 1. 4 PONDERACIÓN DE DIFERENCIAS ENTRE HERRAMIENTAS FRONT-END	24
TABLA 1. 5 DIFERENCIAS ENTRE HERRAMIENTAS BACK-END	24
TABLA 1. 6 PONDERACIÓN DE DIFERENCIAS ENTRE HERRAMIENTAS BACK-END	25
TABLA 2. 1 DEFINICIÓN DEL PROBLEMA	27
TABLA 2. 2 POSICIÓN DEL PRODUCTO	28
TABLA 2. 3 RESUMEN PARTICIPANTES	29
TABLA 2. 4 RESUMEN USUARIOS	30
TABLA 2. 5 PERFIL DE USUARIO ADMINISTRADOR DEL SISTEMA	31
TABLA 2. 6 PERFIL DE USUARIO GERENTE	31
TABLA 2. 7 PERFIL DE USUARIO EMPLEADO	32
TABLA 2. 8 DICCIONARIO DE ACTORES	40
TABLA 2. 9 DESCRIPCIÓN DEL CASO DE USO GESTIONAR ESPECIE	41
TABLA 2. 10 DESCRIPCIÓN DEL CASO DE USO GESTIONAR PISCINA	43
TABLA 2. 11 DESCRIPCIÓN DEL CASO DE USO GESTIONAR EMPACADORA	45
TABLA 2. 12 DESCRIPCIÓN DEL CASO DE USO GESTIONAR COTIZACIÓN	47
TABLA 2. 13 DESCRIPCIÓN DEL CASO DE USO GESTIONAR COSECHA	49
TABLA 2. 14 DESCRIPCIÓN DEL CASO DE USO GESTIONAR DESPACHO	52
TABLA 2. 15 DESCRIPCIÓN DEL CASO DE USO GESTIONAR LIQUIDACIÓN	54
TABLA 2. 16 DESCRIPCIÓN DEL CASO DE USO GENERAR HISTÓRICO	56
TABLA 2. 17 DESCRIPCIÓN DEL CASO DE USO GESTIONAR USUARIO	57
TABLA 2. 18 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR ESPECIE	59
TABLA 2. 19 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR PISCINA	61
TABLA 2. 20 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR EMPACADORA	63
TABLA 2. 21 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR COTIZACIÓN	65
TABLA 2. 22 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR COSECHA	67
TABLA 2. 23 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR DESPACHO	70
TABLA 2. 24 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR LIQUIDACIÓN	71
TABLA 2. 25 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GENERAR HISTÓRICO	73
TABLA 2. 26 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR USUARIO	74
TABLA 2. 27 PAQUETE DE ANÁLISIS DE MODULO DEL SISTEMA	76
TABLA 2. 28 ESTRUCTURA DE PÁGINAS	89
TABLA 2. 29 DESCRIPCIÓN ESTRUCTURA PAGINAS LIST	90
TABLA 2. 30 DESCRIPCIÓN ESTRUCTURA PAGINAS FRM	91
TABLA 2. 31 DESCRIPCIÓN DE LA ARQUITECTURA DEL SISTEMA	93
TABLA 2. 32 MAPEO DE CLASES A ENTIDADES DE LA BASE DE DATOS	97
TABLA 2. 33 DESCRIPCIÓN ESTÁNDARES DE PROGRAMACIÓN	100
TABLA 2. 34 DESCRIPCIÓN ESTÁNDARES DE CONTROLES	101
TABLA 2. 35 DESCRIPCIÓN ESTÁNDARES DE DISEÑO BASE DE DATOS	102
TABLA 2. 36 DESCRIPCIÓN EQUIPO DE PRUEBAS	104
TABLA 2. 37 PRUEBA DE INICIO DE SESIÓN DE USUARIO (INGRESO CORRECTO)	105
TABLA 2. 38 PRUEBA DE INICIO DE SESIÓN DE USUARIO (INGRESO FALLIDO)	105
TABLA 2. 39 PRUEBA CREAR ESPECIE	106
TABLA 2. 40 PRUEBA MODIFICAR ESPECIE	107
TABLA 2. 41 PRUEBA ELIMINAR ESPECIE	108
TABLA 2. 42 PRUEBA CREAR PISCINA	109
TABLA 2. 43 PRUEBA MODIFICAR PISCINA	110
TABLA 2. 44 PRUEBA ELIMINAR PISCINA (DATOS CORRECTOS)	111
TABLA 2. 45 PRUEBA CREAR EMPACADORA	111
TABLA 2. 46 PRUEBA MODIFICAR EMPACADORA	113
TABLA 2. 47 PRUEBA ELIMINAR EMPACADORA	114
TABLA 2. 48 PRUEBA CREAR COTIZACIÓN	115
TABLA 2. 49 PRUEBA MODIFICAR COTIZACIÓN	116
TABLA 2. 50 PRUEBA ELIMINAR COTIZACIÓN (DATOS CORRECTOS)	117
TABLA 2. 51 PRUEBA CREAR COSECHA	118
TABLA 2. 52 PRUEBA MODIFICAR COSECHA	119
TABLA 2. 53 PRUEBA ELIMINAR COSECHA (DATOS CORRECTOS)	121

INDICE DE ILUSTRACIONES

FIGURA 2.1 PERSPECTIVA DEL PRODUCTO	32
FIGURA 2.2 MODELO DEL NEGOCIO	39
FIGURA 2.3 MODELO CASOS DE USOS.....	39
FIGURA 2.4 CASO DE USO GESTIONAR ESPECIE	41
FIGURA 2.5 CASO DE USO GESTIONAR PISCINA	43
FIGURA 2.6 CASO DE USO GESTIONAR EMPACADORA	45

FIGURA 2.7 CASO DE USO GESTIONAR COTIZACIÓN	47
FIGURA 2.8 CASO DE USO GESTIONAR COSECHA.....	49
FIGURA 2.9 CASO DE USO GESTIONAR DESPACHO.....	52
FIGURA 2.10 CASO DE USO GESTIONAR LIQUIDACIÓN.....	54
FIGURA 2.11 CASO DE USO GENERAR HISTÓRICO.....	56
FIGURA 2.12 CASO DE USO GESTIONAR USUARIO	57
FIGURA 2.13 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR ESPECIE.....	59
FIGURA 2.14 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR PISCINA.....	61
FIGURA 2.15 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR EMPACADORA.....	63
FIGURA 2.16 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR COTIZACIÓN.....	65
FIGURA 2.17 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR COSECHA	67
FIGURA 2.18 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR DESPACHO	69
FIGURA 2.19 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR LIQUIDACIÓN	71
FIGURA 2.20 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GENERAR HISTÓRICO	73
FIGURA 2.21 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR USUARIO.....	74
FIGURA 2.22 DEPENDENCIA ENTRE LOS PAQUETES DEL MÓDULO DEL SISTEMA.....	77
FIGURA 2.23 SUBSISTEMAS DEL MÓDULO DEL SISTEMA.....	78
FIGURA 2.24 DIAGRAMA DE CLASES.....	79
FIGURA 2.25 DIAGRAMA DE SECUENCIA DEL CASO DE USO GESTIONAR ESPECIE	80
FIGURA 2.26 DIAGRAMA DE SECUENCIA DEL CASO DE USO GESTIONAR PISCINA.....	81
FIGURA 2.27 DIAGRAMA DE SECUENCIA DEL CASO DE USO GESTIONAR EMPACADORA.....	82
FIGURA 2.28 DIAGRAMA DE SECUENCIA DEL CASO DE USO GESTIONAR COTIZACIÓN.	83
FIGURA 2.29 DIAGRAMA DE SECUENCIA DEL CASO DE USO GESTIONAR COSECHA.	84
FIGURA 2.30 DIAGRAMA DE SECUENCIA DEL CASO DE USO GESTIONAR DESPACHO.	85
FIGURA 2.31 DIAGRAMA DE SECUENCIA DEL CASO DE USO GESTIONAR LIQUIDACIÓN.....	86
FIGURA 2.32 DIAGRAMA DE SECUENCIA DEL CASO DE USO GENERAR HISTÓRICO.....	86
FIGURA 2.33 DIAGRAMA DE SECUENCIA DEL CASO DE USO GESTIONAR USUARIO.....	87
FIGURA 2.34 MAPA DE NAVEGACIÓN.....	88
FIGURA 2.35 ESTRUCTURA DE PÁGINAS.....	89
FIGURA 2.36 ESTRUCTURA DE LA PAGINA LIST	91
FIGURA 2.37 ESTRUCTURA DE LA PÁGINA FRM	92
FIGURA 2.38 DIAGRAMA DE LA ARQUITECTURA DEL SISTEMA	93
FIGURA 2.39 DIAGRAMA DE DESPLIEGUE	94
FIGURA 2.40 DIAGRAMA DE COMPONENTES	95
FIGURA 2.41 MODELO CONCEPTUAL DE LA BASE DE DATOS SACP.....	98
FIGURA 2.42 MODELO FÍSICO DE LA BASE DE DATOS SACP	99

INTRODUCCION

El presente trabajo trata el desarrollo de un sistema de apoyo a los procesos de Cosecha y Post-Cosecha dentro de la camaronera "Pampas de Cayanca" para lo cual se definieron tres capítulos.

En el capítulo 1 se ha descrito lo referente a la situación actual de la camaronera, la selección tanto de la metodología como la de las herramientas para el desarrollo, que han sido elegidas en base a criterios y parámetros establecidos.

El capítulo 2 contiene la documentación técnica correspondiente al flujo de trabajo de las etapas de análisis, diseño, implementación y pruebas.

En la etapa de análisis se presentan la especificación de requerimientos para lo cual se tomo en cuenta la situación actual de los procesos de Cosecha y Post-Cosecha de la camaronera, el esquema y especificación del modelo del negocio y la definición de casos de usos del sistema, además, de los modelos de colaboración del sistema correspondientes a los casos de usos definidos en el flujo correspondiente.

En la etapa de diseño se presenta el diseño de clases, además se presentan los modelos de secuencia de los casos de usos del sistema, además el diseño de interfaces, el diseño arquitectónico y el modelo de despliegue.

En la etapa de implementación se implemento el prototipo con la utilización de las herramientas seleccionadas en el capítulo 1, posteriormente, en la etapa de pruebas se llevaron a cabo las pruebas para verificar que los requerimientos fueron cumplidos y la evaluación de las mismas.

En el capítulo 3, se presentan las conclusiones y recomendaciones que se pudo obtener de todas las etapas del desarrollo del presente proyecto.

Como producto de este desarrollo se obtuvo el Sistema para los procesos de Cosecha y Post-cosecha de la camaronera "Pampas de Cayanca" SACP, que fue desarrollado bajo tecnología.NET, específicamente en el lenguaje C#, el mismo que llegó a cumplir los requerimientos, planteados para este proyecto

CAPÍTULO 1 PLANTEAMIENTO DEL PROBLEMA

1.1 DESCRIPCIÓN DEL PROBLEMA.

1.1.1 LA EMPRESA.

Para el desarrollo de este proyecto se cuenta con la camaronera “Pampas de Cayanca”, que es una empresa afiliada a la Cámara de Comercio de El Oro, fundada en 1995.

La camaronera en estudio está constituida como una sociedad anónima, encargada del manejo administrativo y técnico de la misma.

La camaronera “Pampas de Cayanca” tiene a su disposición treinta hectáreas de área de cultivo, segmentada en tres piscinas camaroneras de aproximadamente nueve hectáreas cada una.

La camaronera se encuentra ubicada en el cantón Barbones en la Provincia de El Oro, a veinte minutos de Puerto Bolívar.

1.1.2 SITUACIÓN ACTUAL DE LA EMPRESA.

Se ha determinado que la empresa tiene algunas falencias en cuanto a los registros de información en los procesos de cosecha y post- cosecha, tales como:

- La principal deficiencia en los procesos es la ausencia de datos para generar información, debido al mal manejo en el almacenamiento de los mismos; lo que ocasiona postergación en la toma de decisiones.

- Debido a la cantidad de información que se manipula en estos procesos no han podido ser controlados a tiempo debido a la falta de una calendarización de eventos.
- No existe un seguimiento y control de estos procesos por lo que cualquier cálculo de producción es supuesto y teórico, lo cual ocasiona que para la toma de decisiones se le dé un mayor grado de importancia a los datos históricos que los actuales.
- No existe información sobre el número de cosechas de cada una de las piscinas existentes en la camaronera. Debido a que el ciclo de producción admite realizar de tres a cuatro cosechas anuales por piscina, es indispensable conocer toda la información relacionada a las diferentes cosechas del año y lograr así realizar una comparación entre los diferentes ciclos de producción.
- Al momento de cosechar solo se registran el total de libras producidas por la piscina y no se conserva la cantidad de animales cosechados por categoría en cada una de las liquidaciones, perdiendo así información que sería un considerable apoyo al momento de pretender conocer qué tipo de camarón está exportando la camaronera.
- Es importante señalar que los precios del camarón se modifican continuamente, la alza o la baja de este están dadas por la oferta y demanda del mercado internacional, actualmente no se tiene un registro de los precios ni su fluctuación que ayudaría a proyectar las ventas en los periodos de mayor demanda y a precios altos.
- No se tiene información sobre la evaluación de las ventas (liquidaciones) a las empacadoras. Precisamente por ejemplo el monto total entregado a la empacadora no puede ser conocido debido a la carencia de históricos.

- No existe un registro detallado de la cantidad en libras del camarón que se envía a la empacadora, con lo que se produce contratiempos en la entrega del mismo.

Todas estas falencias por parte de la empresa deben ser resueltas de manera eficaz y efectiva para que logre seguir manteniéndose, competitivamente en el mercado en el que esta se desenvuelve, y sobre todo por el aspecto más importante de todos que es el compromiso que tiene de satisfacer las necesidades del exportador

1.1.3 CONCEPTUALIZACIÓN DE LOS PROCESOS DE COSECHA Y POST-COSECHA

Son aquellos que se dan luego de un ciclo de tres meses en el que el camarón ha alcanzado su tamaño comercial.

1.1.3.1 Cosecha

La cosecha inicia con un muestreo para ver el grado de dureza del camarón, si este es satisfactorio se procede a extraerlo de las piscinas.

Consecuentemente se procede a disminuir lentamente el nivel del agua de las piscinas a través de las compuertas de salidas, para provocar una corriente de agua que consigue la salida tanto del agua como de los animales.

En la compuerta se coloca con anterioridad una malla en la cual se van almacenando los especímenes todavía vivos. Esto permite capturar volúmenes de camarón casi sin pérdida. A medida que se llena la malla se realizan cierres continuos, lo cual permite recolectar la cantidad de camarón almacenado para luego ser colocados en recipientes y luego llevado a los lavaderos, esto hasta vaciar completamente la piscina.

A medida que baja el nivel de agua de la piscina quedan ciertas pozas de agua en la cual tienden a concentrarse cierta cantidad de camarones los cuales son extraídos fácilmente mediante la utilización de atarrayas¹.

1.1.3.2 Post-Cosecha

Posterior a la cosecha se procede a colocar los diferentes especímenes capturados en gavetas de distintas formas con hielo traídos por la empacadora a despachar, para luego ser procesados, con lo que se evita la excesiva manipulación del producto lo que contribuye a la obtención de una mejor calidad y conservación del mismo.

Las empacadoras transportan el producto con sus propios vehículos, receptan estos volúmenes de camarón y consecutivamente se entregan las liquidaciones correspondientes a la cosecha, al gerente de la camaronera, por el producto recibido.

1.1.4 SITUACIÓN ACTUAL DE LOS PROCESOS DE COSECHA Y POST-COSECHA.

Los procesos actualmente no tienen fundamentos técnicos, ni teóricos, debido a que estos son realizados de una forma empírica, que por lo general es aplicada en empresas pequeñas que se dedican a este tipo de negocio.

Simultáneamente no se lleva un seguimiento de lo que se ha realizado en los procesos de cosecha y post- cosecha como se menciona en la sección 1.1.2, por lo que no se consigue realizar análisis de registros históricos que ayuden al mejoramiento de ingresos de la camaronera.

¹ ATARRAYAS: Son redes de forma cónica que son lanzadas para que caigan sobre un área específica.

1.2 DETERMINACIÓN DE LA METODOLOGÍA DE DESARROLLO A APLICAR

Para la determinación de la metodología se tomara en cuenta ciertos criterios relacionados con el tipo de sistema a desarrollarse que nos permitirán determinar la metodología más apropiada para continuar con el desarrollo e implementación del sistema.

1.2.1 METODOLOGÍA ²

1.2.1.1 Definición

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software.

Una metodología puede seguir uno o varios modelos de ciclo de vida, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales.

1.2.1.2 Tendencia

1.2.1.2.1 *Metodologías Ágiles* ³

Las metodologías ágiles son sin duda uno de los temas recientes en ingeniería de software que están acaparando gran interés.

La curiosidad que siente la mayor parte de ingenieros de software, profesores, e incluso alumnos, sobre las metodologías ágiles hace prever una fuerte proyección

² WIKILEARNING; Metodologías de desarrollo de software;
http://www.wikilearning.com/metodologias_de_desarrollo_de_software-wkc-3617.htm

³ FOWLER, Martin; The New Methodology, <http://martinfowler.com/articles/newMethodology.html>

industrial. Por un lado, para muchos equipos de desarrollo el uso de metodologías tradicionales les resulta muy lejano a su forma de trabajo actual considerando las dificultades de su introducción e inversión asociada en formación y herramientas.

Por otro, las características de los proyectos para los cuales las metodologías ágiles han sido especialmente pensadas se ajustan a un amplio rango de proyectos industriales de desarrollo de software; aquellos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos, requisitos volátiles, y/o basados en nuevas tecnologías.

Se han popularizado ya cerca de una decena de métodos ágiles, con Valores y Técnicas similares: Rapid Application Development (RAD) de James Martin (el primer método Ágil), Adaptative Software Development (ASD), Feature Driven Development (FDD), Crystal Clear, Dynamic Software Development Method (DSDM), Scrum, Programación eXtrema (eXtreme Programming o XP), probablemente el más difundido por su carácter “contestatario”.

1.2.1.2.2 Metodología Tradicional⁴

Los organismos y corporaciones han desarrollado una plétora⁵ de estándares comprensivos que han ido jalonando la historia y poblando los textos de metodologías e ingeniería de software: CMM, Spice, BootStrap, TickIt, derivaciones de ISO9000, SDCE, Trillium.

Algunos son verdaderamente métodos; otros, metodologías de evaluación o estimación de conformidad; otros más, estándares para metodologías o meta-modelos. Al lado de ellos se encuentra lo que se ha llamado una ciénaga⁶ de estándares generales o específicos de industria a los que se someten organizaciones que desean (o deben) articular sus métodos conforme a diversos criterios de evaluación, vehículos de selección de contratistas o marcos de referencia.

⁴ FOWLER, Martin; The New Methodology, <http://martinfowler.com/articles/newMethodology.html>

⁵ PLETORA: *Abundancia excesiva de alguna cosa.*

⁶ CIENAGA: *Es un cuerpo de agua con circulación de agua en dos estaciones inversas*

Hay muchos frameworks y disciplinas, pero lo primordial no es la proliferación de variedades, sino la sujeción de todos los ejemplares a unos pocos tipos de configuraciones o flujos posibles.

Los modelos de las metodologías tradicionales difieren bastante en su conformación y en su naturaleza, pero exaltan casi siempre las virtudes del planeamiento y poseen un espíritu normativo. Comienzan el análisis completo de los requerimientos del usuario. Después de un largo período de intensa interacción con usuarios y clientes, los ingenieros establecen un conjunto definitivo y exhaustivo de rasgos, requerimientos funcionales y no funcionales. Esta información se documenta en forma de especificaciones para la segunda etapa, el diseño, en el que los arquitectos, trabajando junto a otros expertos en temas puntuales, generan la arquitectura del sistema. Luego los programadores implementan ese diseño bien documentado y finalmente el sistema completo se prueba y se despacha.

Las metodologías tradicionales imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

1.2.2 METODOLOGIA A UTILIZARSE

Dentro del mundo de desarrollo de software existen varias metodologías que ofrecen diferentes contextos que se pueden seguir en cuanto a la ejecución de un proyecto.

Para efectos del presente trabajo con la ayuda de artículos de libros e Internet, se presenta un cuadro comparativo con las metodologías de desarrollo de software más utilizadas de donde se elegirá la más adecuada para continuar con el desarrollo e implementación del sistema objeto del presente proyecto, cabe

mencionar que no se trata de un estudio, puesto que no es parte del alcance del proyecto.

1.2.2.1 Rup (Rational Unified Process)^{7 8}

RUP (*Proceso Unificado de Rational*) fue desarrollado como el proceso complementario al UML (Unified Modeling Language). El RUP es un armazón de proceso y como tal puede acomodar una gran variedad de procesos. De hecho ésta es la crítica principal al RUP por parte de algunos autores: como puede ser cualquier cosa acaba siendo nada. Prefieren un proceso que diga qué hacer en lugar de dar opciones infinitas.

Como resultado de esta mentalidad de armazón de procesos, el RUP puede usarse en un estilo muy tradicional de cascada o de una manera ágil. Como resultado se puede usar el RUP como un proceso ágil, o como un proceso pesado, todo depende de cómo se lo adapte a cada ambiente.

RUP divide en 4 fases el desarrollo del software:

- Inicio
- Elaboración
- Construcción
- Transmisión

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones

⁷ MENDOZA, María; Metodologías De Desarrollo De Software;
http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf.

⁸ Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. Descripción del Proceso Unificado de Desarrollo.
<https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducción%20a%20RUP.doc>.

precedentes. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas, de Desarrollo y de Soporte

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente, esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

1.2.2.2 Xp (eXtreme Programming)^{9 10}

XP (*Programación Extrema*) es una de las metodologías de desarrollo de software de bajo riesgo y flexible para proyectos de corto plazo, pequeños y medianos equipos, y cuyo plazo de entrega es de inmediato. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

El ciclo de vida ideal de XP consiste de seis fases:

- Exploración
- Planificación de la Entrega (Release)
- Iteraciones
- Producción
- Mantenimiento
- Muerte del Proyecto

⁹ FOWLER, Martin; *The New Methodology* ; <http://martinfowler.com/articles/newMethodology.html>
Extreme Programming; A gentle introduction; <http://www.extremeprogramming.org/>

¹⁰ LETELIER, Patricio; PENADES, Carmen; *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*; <http://www.willydev.net/descargas/masyxp.pdf>

La ausencia de énfasis en la arquitectura durante las primeras iteraciones (no hay arquitectos en XP) es un factor negativo

XP trata de dar al cliente el software que él necesita y cuando lo necesita con la entrega de pequeños sistemas rápidamente desarrollados, al menos uno cada 2 o 3 meses, fortalece al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

Esta adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico, pues pone gran énfasis en las pruebas, para la que propone la programación en pares.

1.2.2.3 Rad (Rapid Application Development)^{11 12}

RAD (*Desarrollo Rápido de Aplicaciones*) comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución

La estrategia para lograr desarrollos rápidos óptimos implica evitar los errores clásicos, aplicando fundamentos de desarrollo, administrar riesgos correctamente y tener prácticas bien coordinadas y agendadas.

RAD se desarrolla, a través de 4 dimensiones principales que son:

- Personas (pues los proyectos más efectivos, son aquellos que sacan partido al potencial humano de sus desarrolladores)
- Proceso (está orientado al Cliente)

¹¹ WIKIMEDIA FOUNDATION, INC .Desarrollo rápido de aplicaciones, http://es.wikipedia.org/wiki/Desarrollo_r%C3%A1pido_de_aplicaciones.

¹² UNIVERCIDAD DE MONTERREY, <http://www.udem.edu.mx/udem/profesores/laceves/rad/m2arad.ppt>

- Producto (cuida que el aspecto, características, calidad y rendimiento de la aplicación sean flexibles)
- Tecnología ya que recomienda utilizar herramientas efectivas probadas en proyectos similares y recomienda manejar lenguajes de más alto nivel.

La calidad definido por RAD, es el grado al cual un uso entregado resuelve las necesidades de usuarios así como el grado al cual un sistema entregado tiene costes de mantenimiento bajos. El RAD aumenta la calidad con la implicación del usuario en las etapas del análisis y del diseño.

1.2.2.4 Msf (Microsoft Solution Framework) ¹³

MSF es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos, se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF es adaptable, escalable ya que consigue organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más, flexible ya que es utilizada en el ambiente de desarrollo de cualquier cliente, además puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología y se pueden liberarse nuevas versiones diariamente

El Modelo de proceso MSF consta de cinco fases distintas:

- Previsión
- Planeamiento
- Desarrollo
- Estabilización
- Implementación

¹³ MENDOZA, María; Metodologías De Desarrollo De Software;
http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

Uno de los beneficios de la metodología de MSF es contar con una evaluación estándar por medio de la cual se puede validar la habilidad de desarrollar software en una organización.

1.2.2.5 Evaluación de Metodologías

La Tabla 1.1 recoge esquemáticamente algunos criterios de las metodologías presentadas. Estos criterios no sólo se refieren al proceso en sí, sino también al contexto del equipo así como a su organización.

Tabla 1. 1 Criterios Selección de Metodología

Criterios	%	RUP	MSF	RAD	XP
Grado de Conocimiento	20	15	10	10	10
Soporte Orientada a Objetos	10	10	10	10	10
Adaptable a Cambios	15	10	15	10	15
Basado en Casos de uso	10	10	5	10	5
Posee Documentación Adecuada	15	15	15	15	10
Facilita la Integración entre las Etapas de Desarrollo	10	10	10	10	10
Relación con UML	10	10	8	8	8
Permite Desarrollar Software Sobre Cualquier Tecnología	10	10	10	10	10
	100	90	83	83	78

Fuente: Los Autores

La notación utilizada en la Tabla 1.1, permite cuantificar el porcentaje que posee cada metodología respecto a los criterios expuestos.

La Tabla 1.2 muestra los distintos valores del porcentaje asignado a cada criterio de la Tabla 1.1.

Tabla 1. 2 Ponderación de Criterios selección de Metodología

Criterios	Regular	Normal	Bueno	Alto
Grado de Conocimiento	5	10	15	20
Soporte Orientada a Objetos	2	5	8	10
Adaptable a Cambios	2	5	10	15
Basado en Casos de uso	2	5	8	10
Posee Documentación Adecuada	2	5	10	15
Facilita la Integración entre las Etapas de Desarrollo	2	5	8	10
Relación con UML	2	5	8	10
Permite Desarrollar Software Sobre Cualquier Tecnología	2	5	8	10

Fuente: Los Autores

1.2.3 SELECCIÓN DE LA METODOLOGÍA

En base a los criterios antes mencionados en la Tabla 1.1 se decidió utilizar RUP como metodología a utilizar, pues presenta la puntuación más alta 90%, a más de que el medio ambiente de desarrollo de software, la naturaleza del software, el conocimiento por parte del equipo de trabajo, así como las herramientas de desarrollo de software son las propicias para la utilización de dicha metodología.

Solo se utilizarán las disciplinas de RUP más útiles para el caso, con el propósito de obtener un producto de calidad que no llegue a ser muy extenso y permita la rápida comprensión del sistema desarrollado.

1.3 DETERMINACIÓN DE LA HERRAMIENTA A UTILIZAR

Para la determinación de las herramientas a utilizar se ha considerado los requerimientos de la empresa, la metodología y el conocimiento que se tiene de las mismas, razón por la cual en esta sección solo se hará una breve descripción; y los criterios por los cuales se las ha seleccionado.

De la misma manera que se realizó en la selección de la metodología para el presente trabajo se recurrió a la ayuda de artículos de libros e Internet, se presenta las herramientas de desarrollo de software más utilizadas de donde se han elegido las más adecuadas para continuar con el análisis de requerimientos, diseño, modelado, construcción, desarrollo e implementación del sistema.

1.3.1 DESCRIPCIÓN DE HERRAMIENTAS TIPO FRONT END Y BACK END.

Estas herramientas permiten optimizar el desarrollo de sistemas y aplicaciones en todas sus fases.

A continuación se presenta una breve descripción de estas:

1.3.1.1 Front-End ¹⁴

Las herramientas de tipo front-end son aquellas que se emplean para el desarrollo de las aplicaciones a nivel de usuario. Para ello se usan los lenguajes de programación, en estas se diseñan los procesos del negocio.

De entre los diversos lenguajes de programación y herramientas que existen hoy en día en el mercado, y por ser los más empleados y populares por sus grandes características y facilidades que prestan a los desarrolladores se las presentan a continuación:

¹⁴ WIKIMEDIA FOUNDATION, INC. Front-end_y_back-end; http://es.wikipedia.org/wiki/Front-end_y_back-end

1.3.1.1.1 *Microsoft Visual C#.Net 2005*¹⁵

Microsoft Visual C#.Net 2005 es parte de la suite Microsoft Visual Studio 2005 la cual permite construir aplicaciones multi-capa, Web (páginas ASP.NET), dispositivos móviles, clientes inteligentes, etc.

Proporciona un entorno de desarrollo muy productivo debido a que se puede arrastrar y colocar cualquier control visual o componente, un editor de códigos enriquecido, un entorno de depuración de primera clase y una gran productividad adicional que le ayuda a elaborar aplicaciones de un modo más rápido.

El conjunto de herramientas de Visual Studio y .NET Framework están diseñadas desde su origen para construir aplicaciones de Sistemas conectados. Además posee nuevas herramientas de diseño y comunicación de bases de datos visuales e integrados que facilitan la elaboración de aplicaciones fiables, escalables, orientadas a los datos para SQL Server y otras bases de datos.

1.3.1.1.2 *Eclipse Java*¹⁶

Eclipse es una plataforma de software de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Eclipse es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios Web, programas en C++ o aplicaciones Java.

Se trata de un entorno de desarrollo integrado (IDE¹⁷) en el que se encuentran todas las herramientas y funciones necesarias para desarrollar cualquier trabajo,

¹⁵ WIKIMEDIA FOUNDATION INC, Microsoft Visual Studio.
http://es.wikipedia.org/wiki/Microsoft_Visual_Studio.

¹⁶ WIKIMEDIA FOUNDATION INC. Eclipse (software).
[http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software)).

recogidas además en una interesante interfaz que lo hace fácil y agradable de usar.

Cuenta con un editor de texto donde se puede ver el contenido del fichero en el que se está trabajando, una lista de tareas, y otros módulos similares. Si bien las funciones de Eclipse son más bien de carácter general, las características del programa se pueden ampliar y mejorar mediante el uso de plug-ins.

1.3.1.1.3 *JBuilder*¹⁸

La versión JBuilder 2007 está basada en Eclipse.

JBuilder 2007, posee gran compatibilidad para las versiones anteriores de JBuilder, integra un renovado entorno rápido de aplicaciones (RAD) para JBuilder, herramientas y un marco de colaboración con la plataforma abierta de Eclipse.

Además acelera la velocidad de desarrollo con GUIs de Visual EJB y Web Services, ofreciendo un equilibrio entre los mundos open source y el software comercial y con un portal de colaboración integrado, además de que incrementa la confianza de los programadores a través de un mejor control sobre complementos de open source.

Mejorando sus aclamadas capacidades RAD (desarrollo rápido de aplicaciones) y de colaboración en equipo, JBuilder 2007 ha sido completamente rediseñado para aprovechar la plataforma Eclipse y ayudar a las organizaciones a ser más exitosas y productivas con la fuente abierta.

JBuilder 2007 posee características de colaboración de clase empresarial que devuelven la seguridad al proceso de desarrollo de software y al mismo tiempo aprovechan un ecosistema de complementos comerciales y de open source, herramientas de terceros y estructuras.

¹⁷ IDE: *Integrated Development Environment*

¹⁸ SAM SISTEMAS. JBuilder® 2007. <http://www.samsistemas.com.ar/partners/jbuilder.html>.

1.3.1.2 Back-End¹⁹

Las herramientas de tipo back-end son aquellas que se utilizan para el almacenamiento de datos es decir son herramientas de base de datos.

De entre las diversas herramientas que existen de este tipo, las más empleadas y populares hoy en día en el mercado por sus grandes características y prestaciones para soportar gran cantidad de datos en entornos empresariales y a nivel de Usuarios son:

1.3.1.2.1 *MySql*²⁰

MySql es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones, surgió como un manejador de pequeñas bases de datos, rápidas y ágiles.

Es un sistema de administración relacional de bases de datos, dado que archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas.

Además combina confiabilidad y funcionamiento de la empresa-grado con las características estándar-obedientes del SQL 2003 avanzados, haciéndole la solución más rentable de la base de datos para los usos negocio-críticos.

Es software de fuente abierta, es decir que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySql y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySql usa el GPL (GNU General Public License) para definir qué puede hacer y que no puede hacer con el software en diferentes situaciones.

¹⁹ WIKIMEDIA FOUNDATION, INC .Front-end_y_back-end; http://es.wikipedia.org/wiki/Front-end_y_back-end

²⁰ MYSQL HISPANO, <http://www.mysql-hispano.org/articles.php>

Una de sus principales ventajas es que es soportada por la mayoría de los proveedores de alojamiento Web (webhosting²¹), por lo cual se encuentra instalada en casi todos los servidores Web de Internet.

1.3.1.2.2 Oracle 10g^{22 23}

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Su mayor defecto es su enorme precio, (según versiones y licencias). Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.

Oracle se encuentra prácticamente en todas las industrias alrededor del mundo.

Oracle es la primera compañía de software que desarrolla e implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de

²¹ Webhosting: *Es un servicio de almacenamiento de información, imágenes, videos, así como de otro tipo de contenidos accesibles a través de "Servidores Web".*

²² ORACLE. Oracle Enterprise Manager 10g Release 2.
http://www.oracle.com/lang/es/enterprise_manager/index.html.

²³ WIKIMEDIA FOUNDATION INC. Oracle. <http://es.wikipedia.org/wiki/Oracle>. 20 Abril 2008.

productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. Oracle es el proveedor mundial líder de software para administración de información, y la segunda empresa de software.

Oracle a partir de la versión 10g Release 2, cuenta con 5 ediciones:

- Oracle Database Enterprise Edition (EE).
- Oracle Database Standard Edition (SE).
- Oracle Database Standard Edition One (SE1).
- Oracle Database Express Edition (XE).
- Oracle Database Personal Edition (PE).

La única edición gratuita es la Express Edition, que es compatible con las demás ediciones de Oracle Database 10gR2 y Oracle Database 11g.

1.3.1.2.3 *Microsoft SQL Server Express Edition 2005*²⁴

SQL Server Express Edition es una edición gratuita de MS SQL Server, con la que se puede utilizar las bases de datos SQL sin ningún coste adicional.

A pesar de ser gratuita, posee una serie de funcionalidades que pueden cubrir de sobra los requisitos iniciales de muchos proyectos serios. En algunos casos los puede cubrir para todo su ciclo de vida útil como en el caso de un blog o pequeñas aplicaciones Web CMS²⁵ para Internet.

Entre sus características podemos destacar:

- SQL Server 100 % compatible:
 - ✓ Con todas sus funcionalidades: transacciones, procedimientos almacenados, etc.

²⁴ ARSYS.ES. Servidores Dedicados - SQL Server 2005 Express Edition.

<http://www.arsys.es/ayuda/directorio/productos/servidores-dedicados/sql-server-express-edition.htm>. 2007.

²⁵ CMS: *Content Management System*

- ✓ Facilita a los desarrolladores el escalado de sus aplicaciones a SQL Server de ediciones superiores.
- ✓ Desarrollos distribuibles a clientes que trabajen con SQL Server.
- Herramienta de administración y consulta SQL Server Management Studio Express.
- Utiliza un procesador y hasta 1GB de memoria.
- Tamaño de bases de datos limitado a 4GB.

Además SQL Server Express Edition es una edición de SQL Server que permite, en cualquier momento, migrar a una edición superior.

Por ello, si en algún momento, por el crecimiento de cualquier actividad, se necesita mayor rendimiento o capacidad en las bases de datos, se podrá beneficiarse de las siguientes ventajas:

- No se necesitará aplicar cambios en los desarrollos al ser 100% compatibles.
- Las bases de datos se mantendrán intactas al pasar a una edición superior de SQL Server (en el mismo servidor).

Permite gestionar las bases de datos SQL Server Express Edition desde la herramienta de administración y consulta SQL Server Management Studio Express.

Servidor Dedicado Genérico Windows

- Desde el propio servidor, ejecutando la aplicación anterior que se encuentra pre instalada por omisión.
- En remoto, descargando la aplicación desde la Web de Microsoft e instalándosela en su equipo de trabajo.

Servidor Dedicado Administrado Windows

- En remoto, descargando la aplicación desde la Web de Microsoft e instalándosela en su equipo de trabajo.

Migrar en un futuro la base de datos de un sistema Express a una edición profesional es algo trivial, por lo que si un proyecto requiere SQL Server 2005, es imprescindible conocer bien lo que da de sí la versión Express de la misma.

1.3.1.3 Revisión de Herramientas

Las Tabla 1.3 y Tabla 1.5 recoge esquemáticamente algunas diferencias de las herramientas front-end y back-end.

Tabla 1. 3 Diferencias entre herramientas Front-End

Parámetros	%	Microsoft Visual C# .Net 2005	Eclipse	JBulder
Costo Licencia	30	20 (Bajo-Alto)	30 (Libre-Gratis)	20 (Bajo-Alto)
Facilidad de Aprendizaje	20	20	15	15
Grado Conocimiento	30	25	10	10
Soporte Técnico	20	20	10	15
	100%	85%	65%	60%

Fuente: Los Autores

Por lo tanto la herramienta de Programación recomendable para el sistema es Microsoft Visual C# .Net 2005 específicamente el lenguaje C#, cuyo análisis representa el 85%, ya que es sencillo, moderno y orientado a objetos, permitiendo a los programadores de aplicaciones Web crear una gran variedad de las mismas.

Se tomo en cuenta la funcionalidad de Microsoft Visual C# ya que esta proporciona plantillas de proyectos, asistentes para código y la biblioteca de .NET Framework como soporte directo para los desarrolladores.

La Tabla 1.4 recoge los distintos valores del porcentaje asignado a cada parámetro de la Tabla 1.3.

Tabla 1. 4 Ponderación de Diferencias entre herramientas Front-End

Parámetro	Bajo	Moderado	Optimo
Costo Licencia	20	25	30
Facilidad de Aprendizaje	10	15	20
Grado Conocimiento	10	15	20
Soporte Técnico	10	15	20

Fuente: Los Autores

Tabla 1. 5 Diferencias entre herramientas Back-End

Parámetros	%	Microsoft SQL Server 2005 Express Edition	Oracle 10g	MySql
Costo Licencia	30	30 (Libre-Gratis)	10 (Bajo-Alto)	30 (Libre-Gratis)
Capacidad	20	15	20	15
Grado Conocimiento	30	25	10	10
Soporte Técnico	20	20	20	10
	100%	90%	60%	65%

Fuente: Los Autores

Por lo tanto es recomendable emplear como herramienta de administración de base de datos del sistema SQL Server 2005 Express Edition, cuyo análisis representa el 90%, teniendo en cuenta que nos ofrece una solución integrada de administración y análisis de datos permitiendo que las aplicaciones que trabajan

con esta sean más seguras, estables y confiables ya que reduce la complejidad y el soporte de aplicación de base de datos.

La Tabla 1.6 muestra los distintos valores del porcentaje asignado a cada parámetro de la Tabla 1.5.

Tabla 1. 6 Ponderación de Diferencias entre herramientas Back-End

Parámetros	Bajo	Medio	Optimo
Costo Licencia	10	15	30
Capacidad	10	15	20
Grado Conocimiento	10	25	30
Soporte Técnico	10	15	20

Fuente: Los Autores

Para la selección de la herramienta de desarrollo y base de datos para la implementación del sistema se ha elegido la herramienta Microsoft Visual Studio 2005 C# con la ayuda de la tecnología de desarrollo Web AJAX y Microsoft SQL Server 2005 Express Edition ya que son las herramientas que han obtenido las más altas puntuaciones en la Tabla 1.2 y Tabla 1.3.

CAPÍTULO 2 DESARROLLO DEL SISTEMA INFORMATICO PARA LOS PROCESOS DE COSECHA Y POST-COSECHA DE LA CAMARONERA “PAMPAS DE CAYANCA”.

2.1 ANALISIS

2.1.1 ESPECIFICACIÓN DE REQUERIMIENTOS

2.1.1.1 Documento de Visión

2.1.1.1.1 Propósito

El presente documento, tiene como finalidad identificar el posicionamiento, características, perspectivas, alcance y de forma general la funcionalidad del sistema de administración de la Camaronera “Pampas de Cayanca” que se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

Esta funcionalidad se basa en la gestión de los procesos de Cosecha y Post-Cosecha, de forma que dichos procesos sean capaces de atender los distintos pedidos que les son realizados.

Los detalles de cómo el sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso.

2.1.1.1.2 Alcance

Este documento inicia con la explicación del problema, que se solucionara con la implementación del sistema; describe el posicionamiento que tendrá el producto en el mercado, así como la identificación de los artefactos y los usuarios del mismo.

Se define una perspectiva general del producto, los módulos (gestión de cosecha, gestión de despachos, gestión de empacadoras, etc.) que tendrá y las principales funciones, junto con un detalle de cómo cada característica del sistema se convierte en solución para el problema.

2.1.1.2 Posicionamiento

2.1.1.2.1 Oportunidad de Negocio

Este sistema permitirá a la empresa informatizar la información de sus procesos de cosecha y post-cosecha (gestión de cosecha, gestión de despachos, gestión de empacadoras, etc.) lo cual supondrá un acceso rápido y sencillo a los datos, gracias a interfaces Web sencillas y amigables. Además, los datos accedidos estarán siempre actualizados, lo cual es un factor muy importante para poder llevar un buen control.

2.1.1.2.2 Sentencia que define el problema

Tabla 2. 1 Definición del Problema

El problema de	<ul style="list-style-type: none"> • No se posee una administración ágil en la información de las cosechas (proceso de Cosecha), despachos y liquidaciones (proceso de Post-Cosecha), realizados a las empacadoras. • No existe información actualizada de las empacadoras. • No se cuentan con históricos que ayuden a la toma de decisiones por parte de la Administración.
Afecta a	<ul style="list-style-type: none"> • Representantes de las Empacadoras, • Empleados de la Camaronera,

	<ul style="list-style-type: none"> • Encargados de transporte, • Departamento de Gerencia.
El impacto asociado es	<ul style="list-style-type: none"> • Una administración de cosechas poco eficientes. • Retraso en la toma de decisiones por parte de la Administración. • Perdida de mercado
Una solución adecuada será	<ul style="list-style-type: none"> • Acceso inmediato a la información necesaria de la empacadora. • Facilitara el manejo de información de los procesos de Cosecha (cosecha) y Post-Cosecha (despacho y liquidación).

Fuente: Los Autores

2.1.1.2.3 Sentencia que define la posición del Producto

Tabla 2. 2 Posición del Producto

Para	<ul style="list-style-type: none"> • Personal administrativo de la camaronera “Pampas de Cayanca”, • Empleados de la camaronera
Quienes	<ul style="list-style-type: none"> • Necesitan información prioritaria de las piscinas, cosechas, despachos y las liquidaciones. • Registran la información de los procesos de cosecha y post-cosecha de la camaronera.
SACP	Es un sistema Web para la administración de los procesos de cosecha y post-cosecha de la camaronera “Pampas de Cayanca”.
Que	<ul style="list-style-type: none"> • Registrara la información necesaria de cada uno de los usuarios. • Mostrara información confiable de los históricos de la camaronera. • Almacenara la información necesaria para gestionar

	los procesos de Cosecha (cosecha) y Post-Cosecha (despacho, liquidación) de la camaronera.
A diferencia	<ul style="list-style-type: none"> • Del registro manual de la información de los procesos de Cosecha (cosecha) y Post-Cosecha (despacho, liquidación) de la camaronera. • No poseer un registro digitalizado y centralizado de la información de cada cosecha, despacho y liquidación. • No poseer registros históricos de manera inmediata.
Esta Aplicación	<ul style="list-style-type: none"> • Permitirá gestionar los procesos de Cosecha (cosecha) y Post-Cosecha (despacho, liquidación) de la camaronera mediante una interfaz Web sencilla y amigable. • Proporcionara un acceso rápido y actualizado a la información desde cualquier punto que tenga acceso a la información. • Facilitara la toma de decisiones con la ayuda de históricos. • Permitirá la visualización de la información completa de una liquidación

Fuente: Los Autores

2.1.1.3 Descripción de Participantes y Usuarios

2.1.1.3.1 Resumen de los Interesados en el proyecto

Tabla 2. 3 Resumen Participantes

Nombre	Descripción	Responsabilidades
Directiva de la Camaronera "Pampas de Cayanca"	Grupo de personas que dirigen y administran la camaronera "Pampas de Cayanca"	<ul style="list-style-type: none"> • Elaborar planes estratégicos y controlar que los mismos se cumplan

		<ul style="list-style-type: none"> • Controlar los recursos financieros
Personal Administrativo de la camaronera	Grupo de personas que laboran en la camaronera “Pampas de Cayanca” cumpliendo funciones administrativas.	<ul style="list-style-type: none"> • Llevan el registro de las empacadoras con las que se trabaja. • Planifican las cosechas de cada una de las piscinas

Fuente: Los Autores

2.1.1.3.2 Resumen de Usuarios

Tabla 2. 4 Resumen Usuarios

Nombre	Descripción	Participante al que representa
Administrador	Persona que se encarga de la administración de los usuarios del sistema, así como de generar los históricos de la camaronera.	Personal administrativo de la camaronera
Gerente	Persona encargada de la gestionar los procesos los procesos de Cosecha (cosecha) y Post-Cosecha (despacho, liquidación) del sistema, así como de realizar las cotizaciones.	Personal administrativo de la camaronera
Empleado	Persona encarga de ingresar la información	Personal administrativo de la camaronera

	de los procesos de la camaronera.	
--	-----------------------------------	--

Fuente: Los Autores

2.1.1.3.3 Perfiles de usuario

Administrador del Sistema

Tabla 2. 5 Perfil de usuario Administrador del Sistema

Descripción	Persona que se encarga de la administración de los usuarios del sistema, así como de generar los históricos de la camaronera.
Tipo	Experto en sistemas informáticos y de computación
Responsabilidades	<ul style="list-style-type: none"> • Definir y configurar usuarios. • Mantener el correcto funcionamiento de los diferentes componentes del sistema.

Fuente: Los Autores

Gerente

Tabla 2. 6 Perfil de usuario Gerente

Descripción	Persona encargada de la gestionar los procesos los procesos de Cosecha (cosecha) y Post-Cosecha (despacho, liquidación) del sistema, así como de realizar las cotizaciones.
Tipo	Conocedor del manejo de sistemas informáticos
Responsabilidades	<ul style="list-style-type: none"> • Registrar y modificar la información de las cosechas, despachos y liquidaciones de la camaronera. • Análisis de históricos de las cosechas de la camaronera.

Fuente: Los Autores

Empleado

Tabla 2. 7 Perfil de usuario Empleado

Descripción	Persona encarga de ingresar la información de los procesos de la camaronera.
Tipo	Conocedor del manejo de sistemas informáticos
Responsabilidades	Registrar y modificar la información de las piscinas, especies, empacadoras, cosechas y despachos de la camaronera.

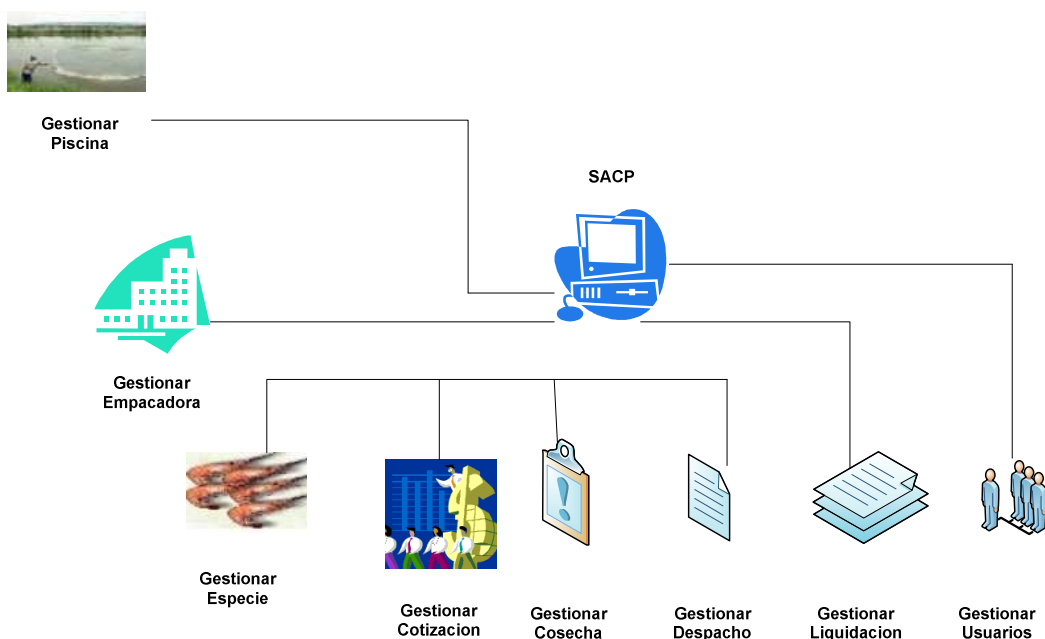
Fuente: Los Autores

2.1.1.4 Resumen Del Producto.

El sistema a desarrollar tiene como objetivo automatizar la información que se realiza en la gestión de los procesos de Cosecha y Post-Cosecha de la camaronera, la cual contempla la Gestión de Usuarios, Piscinas, Especies, Cosechas, Despachos, Liquidaciones y Cotizaciones además de la generación de Históricos.

2.1.1.4.1 Perspectiva del Producto

Figura 2.1 Perspectiva del Producto



Fuente: Los Autores

Gestionar Usuarios del Sistema.

Permite el registro, mantenimiento y validación de los usuarios del sistema.

El mantenimiento de usuarios implica la eliminación, modificación de los datos del personal de la camaronera (secretaria, gerente, supervisor) y asignación de perfiles.

Gestionar Piscina

Permite la gestión de las piscinas de la camaronera, incluye la creación, modificación y eliminación de los mismos

Gestionar Especie

Permite la gestión de las especies de la camaronera, incluye la creación, modificación y eliminación de los mismos

Gestionar Empacadora

Permite la gestión de las Empacadoras a las cuales se despacha el producto (camarón) de la camaronera, incluye la creación, modificación y eliminación de los mismos

Gestionar Cosecha

Permite la gestión de las cosechas de la camaronera así como la asignación de la empacadora a cada cosecha, incluye la creación, modificación y eliminación de los mismos.

Gestionar Despacho

Permite la gestión de los despachos de cada cosecha así como el registro de información adicional necesaria para la creación del mismo.

Gestionar Liquidación

Permite la gestión de la liquidación, así como la visualización de su estado actual, incluye la creación, modificación y eliminación de las mismas

Gestionar Cotización.

Permite la gestión de cada cotización por piscina, con la información del precio y la empacadora a la cual se pretende despachar el producto.

2.1.1.4.2 Supuestos y dependencias

El sistema a desarrollar es independiente de cualquier otro sistema existente dentro de la Camaronera y la información que ingresen los usuarios será validada por el personal encargado de la camaronera.

La camaronera “Pampas de Cayanca” deberá proporcionar los servidores, una red local con acceso a Internet para el mantenimiento y gestión del sistema.

2.1.1.4.3 Licenciamiento e Instalación

El sistema será desarrollado con herramientas que necesitan licencia, lo que implica el pago de licencias del producto, pero únicamente del sistema ya que la camaronera no paga por el uso del software siguiente el cual debe ser previamente instalada antes del producto.

- ✓ Servidor Web Internet Information Server versión 6 o superior.
- ✓ Framework 2.0.
- ✓ Microsoft ASP.NET 2.0 Ajax Extensions 1.0.
- ✓ Microsoft SQL Server 2005 Express.

2.1.2 ESPECIFICACIONES SUPLEMENTARIAS

2.1.2.1 Introducción

2.1.2.1.1 Propósito

El propósito de este documento es especificar aquellos requerimientos que no se capturan en los modelos de casos de uso. Estos requerimientos incluyen características de usabilidad, confiabilidad, desempeño y soporte de la aplicación.

2.1.2.1.2 Alcance

El alcance de este documento se limita a especificar los requerimientos que no se capturan en los modelos de casos de uso para el proceso de desarrollo del sistema.

Es importante recalcar que se especificaran sobre todo las características que tendrá el sistema.

2.1.2.2 Funcionalidad

2.1.2.2.1 Registro de usuarios del sistema

Proceso que permite registrar usuarios del sistema, con su respectivo perfil.

2.1.2.2.2 Generar Históricos

Proceso que permite visualizar información obtenida como resultado de realizar una consulta específica

2.1.2.2.3 *Calculo Cotización*

Proceso que permite calcular el valor de la cotización que se realiza a una piscina, con la respectiva información

2.1.2.2.4 *Registros de liquidaciones*

Proceso que permite registrar varios despachos en una misma liquidación.

2.1.2.3 **Usabilidad**

2.1.2.3.1 *Capacitación a los usuarios del sistema*

Capacitación a los administradores del sistema.

Instruir al administrador del sistema sobre el funcionamiento y el mantenimiento de cada uno de los módulos del sistema, el tiempo requerido para la capacitación será de 3 horas.

Capacitación al Gerente y Empleados.

La capacitación tanto del gerente como de los empleados estará a cargo del administrador del sistema.

2.1.2.3.2 *Facilidad de uso*

El sistema será intuitivo, pues las opciones estarán siempre visibles a los usuarios. La navegabilidad de cada tipo de usuario será definida por medio de sus privilegios, de manera que los interesados tengan una idea clara del tipo de actividades a las que tienen acceso

2.1.2.3.3 *Presentación Visual*

El diseño del sistema será sencillo y con colores no llamativos que vayan de acuerdo al tipo de empresa que va dirigida, su entorno y la ideología de la misma.

2.1.2.4 Confiabilidad

2.1.2.4.1 Disponibilidad

La disponibilidad del sistema será de 24x7, esto es 24 horas los 7 días de la semana.

2.1.2.4.2 Tiempo transcurrido para mantenimiento

El sistema podrá estar fuera de servicio por motivos de mantenimiento por un periodo máximo de dos días.

2.1.2.5 Rendimiento

2.1.2.5.1 Tiempo de respuesta

El tiempo de respuesta para el caso de presentación de información del sistema, dependerá en gran parte a la velocidad de conexión del cliente; sin embargo sin considerar este factor el valor medio del tiempo de respuesta para cualquier presentación de información extraída de la base de datos será de máximo 3 segundos.

2.1.2.5.2 Carga de Trabajo

Una vez puesto en funcionamiento el sistema, se tendrá varios puntos que influenciarán la carga de trabajo del mismo, pero los más influyentes serán:

- El manejo de varias cotizaciones al mismo tiempo.
- La realización de búsquedas y consultas.
- El número máximo de registros que puede manejar la base de datos para:
 - ✓ Cosechas
 - ✓ Despachos

- ✓ Liquidaciones
- ✓ Usuarios

2.1.2.5.3 Degradación

Un factor crítico del sistema será la utilización de datos correctos y completos, por lo que un modo de degradación es la correcta presentación de la información.

2.1.2.5.4 Utilización de recursos

Para que SACP funcione adecuadamente se necesita de un servidor con las siguientes características: Disco duro 100 GB, 1 GB de memoria RAM, procesador Core 2 Duo de 2.0, tarjeta de red Fast Ethernet 10/100 Mbps y una conexión a Internet con una velocidad de transferencia de 128 Kbps.

2.1.2.6 Soporte

2.1.2.6.1 Lenguaje de modelado UML

Para el diseño del SACP utilizaremos el lenguaje de modelado UML 2.0 con la finalidad de estandarizar tanto los diagramas como los componentes, además de facilitar la comprensión de la funcionalidad.

2.1.2.7 Interfaces

2.1.2.7.1 Interfaces de Software

El sistema funcionara bajo un servidor Web Internet Information Server que utilizara el puerto para Web 80.

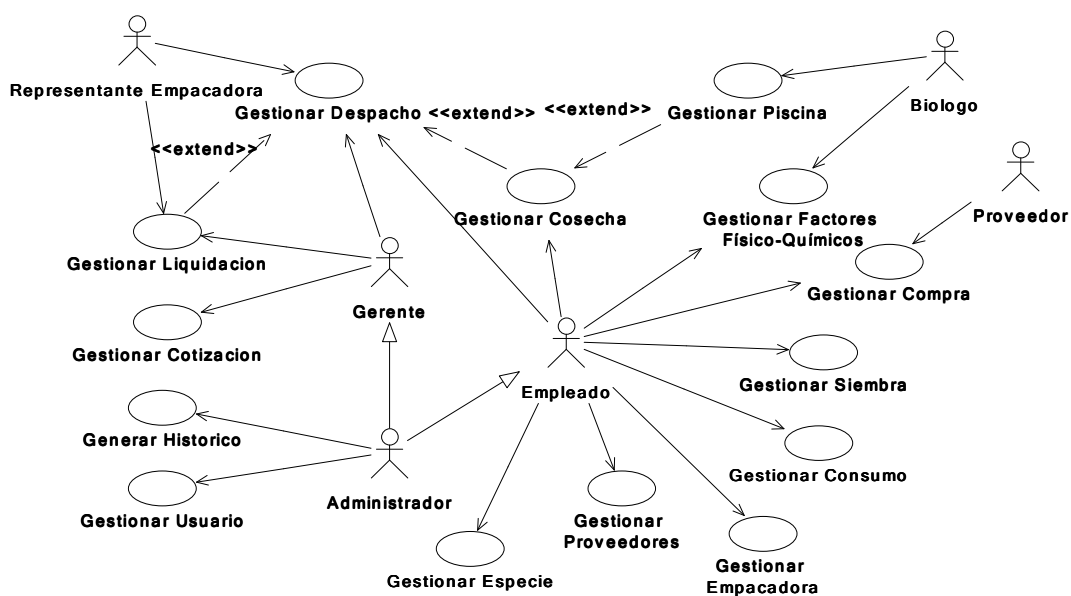
2.1.2.7.2 Interfaces de Comunicación

El sistema se alojara en un servidor propiedad de la Camaronera, no existen conexiones a otros sistemas o dispositivos; el servidor se encuentra conectado a la red interna.

2.1.3 MODELO DEL NEGOCIO

A continuación en la Figura 2.2 se presenta el diagrama correspondiente al modelo del negocio, especificando los usuarios que tendrán los módulos del sistema

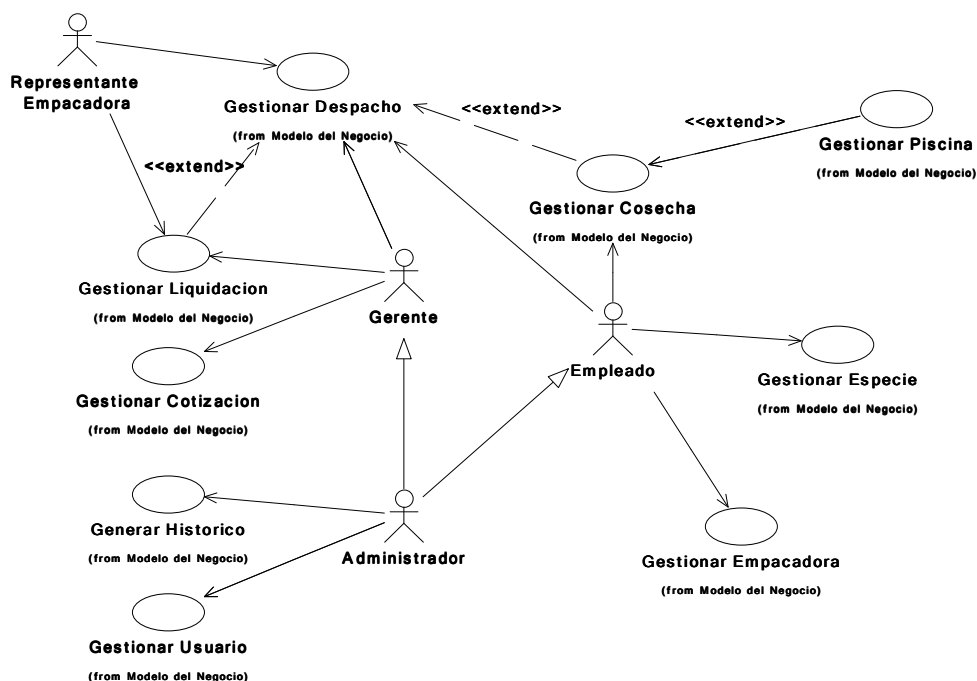
Figura 2.2 Modelo del Negocio



Fuente: Los Autores

2.1.4 CASOS DE USO

Figura 2.3 Modelo Casos de Usos



Fuente: Los Autores

2.1.4.1 Dictionarios de actores

La siguiente sección describe los usuarios que participan en el sistema

Tabla 2. 8 Diccionario de Actores

Empleado	Representa un usuario el cual es el encargado de manipular la información de los diferentes componentes del sistema.
Representante Empacadora	Persona encargada de representar a la empacadora la cual requiere comprar el camarón.
Gerente	Persona encargada de conocer la situación de la camaronera, requiere obtener información de todas las operaciones que se realizan en la camaronera.
Administrador	Persona encargada de la

	administración del sistema, de los usuarios y sus respectivos privilegios.
--	--

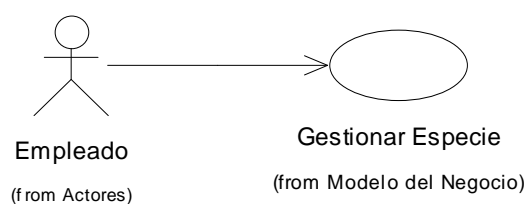
Fuente: Los Autores

2.1.4.2 Descripción de Casos de Uso

En la siguiente sección, se detalla la especificación de cada caso de uso del sistema:

2.1.4.2.1 Gestionar Especie

Figura 2.4 Caso de Uso Gestionar Especie



Fuente: Los Autores

Tabla 2. 9 Descripción del Caso de Uso Gestionar Especie

Casos de uso:	Gestionar Especie
Descripción:	Permite la gestión de especies
Actor:	Empleado
Precondiciones:	Se debe obtener la información de la especie.
Poscondiciones:	El sistema despliega el listado de especies existentes.
Camino básico:	

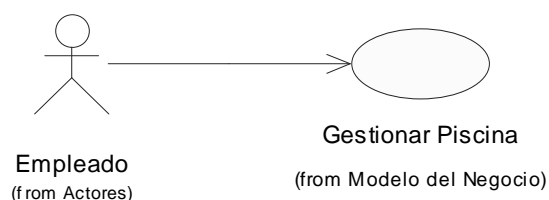
1. Crear	<ol style="list-style-type: none"> 1. El usuario ingresa la información de la nueva especie. 2. El usuario guarda la información. 3. Al usuario se le muestra un mensaje de que la especie ha sido registrada correctamente.
2. Listar	<ol style="list-style-type: none"> 1. El sistema muestra el listado de todas las especies.
3. Modificar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una especie. 2. El usuario modifica los datos de la especie previamente seleccionada. 3. El usuario guarda la información. 4. Los cambios realizados son actualizados en el sistema.
4. Eliminar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una especie. 2. El usuario solicita eliminar especie. 3. El usuario acepta la confirmación de eliminación. 4. La especie es actualizada en el sistema. 5. Al usuario se le muestra un listado de todas las especies.
5. Ver	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una especie. 2. El sistema muestra al usuario el detalle de la especie.
Camino alternativo:	
1. Crear	<ul style="list-style-type: none"> • En 1.2, Si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos. • En 1.3, el usuario selecciona aceptar con lo cual el sistema le muestra el listado de todas

	las especies.
3. Modificar	<ul style="list-style-type: none"> En 3.3, si los datos ingresados no son validos se despliegan mensajes de error en los campos incorrectos.
4. Eliminar	<ul style="list-style-type: none"> En 4.3, el usuario cancela la confirmación de eliminación de la especie, evitando la actualización de este en el sistema En 4.4, si la especie tienen registros asociados el sistema muestra un mensaje de error al usuario, no permitiendo la actualización del mismo.

Fuente: Los Autores

2.1.4.2.2 Gestionar Piscina

Figura 2.5 Caso de Uso Gestionar Piscina



Fuente: Los Autores

Tabla 2. 10 Descripción del Caso de Uso Gestionar Piscina

Casos de uso:	Gestionar Piscina
Descripción:	Permite la gestión de piscinas
Actor:	Empleado
Precondiciones:	Obtener la información necesaria de la piscina.
Poscondiciones:	El sistema despliega el listado de piscinas existentes.
Camino básico:	

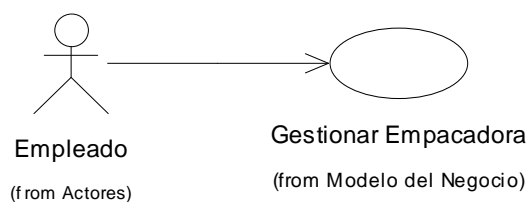
1. Crear	<ol style="list-style-type: none"> 1. El usuario ingresa la información de la nueva piscina 2. El usuario guarda la información 3. Al usuario se le muestra un mensaje de que la piscina ha sido registrada correctamente.
2. Listar	<ol style="list-style-type: none"> 1. El sistema muestra el listado de todas las piscinas.
3. Modificar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una piscina. 2. El usuario modifica los datos de la piscina previamente seleccionada. 3. El usuario guarda la información. 4. Los cambios realizados son actualizados en el sistema.
4. Eliminar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una piscina. 2. El usuario solicita eliminar piscina. 3. El usuario acepta la confirmación de eliminación. 4. La piscina es actualizada en el sistema. 5. Al usuario se le muestra un listado de todas las piscinas.
5. Ver	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una piscina. 2. El sistema muestra al usuario el detalle de la piscina.
Camino alternativo:	

1. Crear	<ul style="list-style-type: none"> • En 1.2, Si los datos ingresados no son validos se despliegan mensajes de error en los campos incorrectos. • En 1.3, el usuario selecciona aceptar con lo cual el sistema le muestra el listado de todas las piscinas.
3. Modificar	<ul style="list-style-type: none"> • En 3.3, si los datos ingresados no son validos se despliegan mensajes de error en los campos incorrectos.
4. Eliminar	<ul style="list-style-type: none"> • En 4.3, el usuario cancela la confirmación de eliminación de la piscina, evitando la actualización de este en el sistema • En 4.4, si la piscina tiene registros asociados el sistema muestra un mensaje de error al usuario, no permitiendo la actualización del mismo.

Fuente: Los Autores

2.1.4.2.3 Gestionar Empacadora

Figura 2.6 Caso de Uso Gestionar Empacadora



Fuente: Los Autores

Tabla 2. 11 Descripción del Caso de Uso Gestionar Empacadora

Casos de uso:	Gestionar Empacadora
Descripción:	Permite la gestión de empacadoras.
Actor:	Empleado

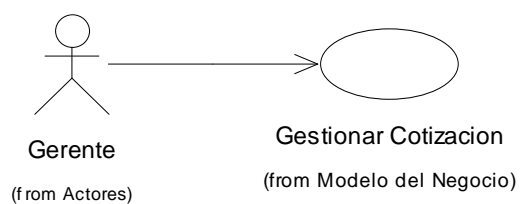
Precondiciones:	Obtener información necesaria de la empaedora.
Poscondiciones:	El sistema despliega el listado de empaedoras existentes.
Camino básico:	
1. Crear	<ol style="list-style-type: none"> 1. El usuario ingresa la información de la nueva empaedora. 2. El usuario guarda la información 3. Al usuario se le muestra un mensaje de que la empaedora ha sido registrada correctamente
2. Listar	<ol style="list-style-type: none"> 1. El sistema muestra el listado de todas las empaedoras.
3. Modificar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una empaedora. 2. El usuario modifica los datos de la empaedora previamente seleccionada 3. El usuario guarda la información 4. Los cambios realizados son actualizados en el sistema.
4. Eliminar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una empaedora 2. El usuario solicita eliminar la empaedora. 3. El usuario acepta la confirmación de eliminación. 4. La empaedora es actualizada en el sistema. 5. Al usuario se le muestra un listado de todas las empaedoras.
5. Ver	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una empaedora 2. El sistema muestra al usuario el detalle de la empaedora.
Camino alternativo:	

1. Crear	<ul style="list-style-type: none"> • En 1.2, si los datos ingresados no son validos se despliegan mensajes de error en los campos incorrectos. • En 1.3, el usuario selecciona aceptar con lo cual el sistema le muestra el listado de todas las Empacadoras.
3. Modificar	<ul style="list-style-type: none"> • En 3.3 si los datos ingresados no son validos se despliegan mensajes de error en los campos incorrectos.
4. Eliminar	<ul style="list-style-type: none"> • En 4.2, el usuario cancela la confirmación de eliminación de la empacadora, evitando la actualización de este en el sistema • En 4.3, si la empacadora tiene registros asociados el sistema muestra un mensaje de error al usuario, no permitiendo la actualización del mismo.

Fuente: Los Autores

2.1.4.2.4 Gestionar Cotización

Figura 2.7 Caso de Uso Gestionar Cotización



Fuente: Los Autores

Tabla 2. 12 Descripción del Caso de Uso Gestionar Cotización

Casos de uso:	Gestionar Cotización
Descripción:	Permite la gestión de cotizaciones
Actor:	Gerente

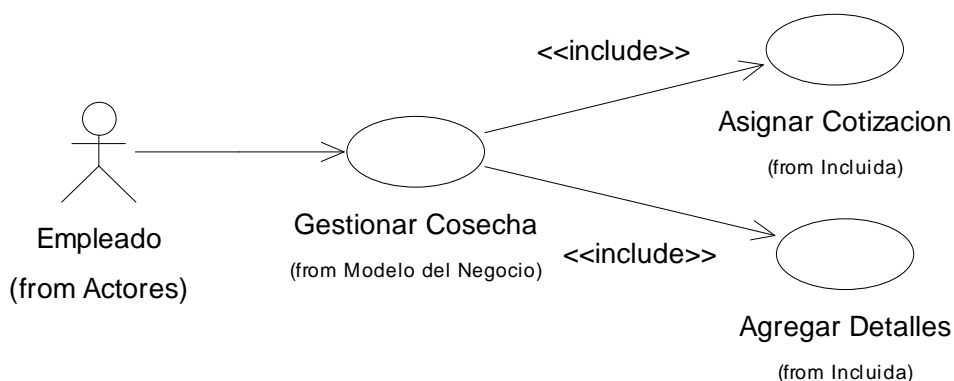
Precondiciones:	Se debe obtener la información de la cotización.
Poscondiciones:	El sistema despliega el listado de cotizaciones existentes.
Camino básico:	
1. Crear	<ol style="list-style-type: none"> 1. El usuario ingresa la información de la nueva cotización. 2. El usuario guarda la información 3. Al usuario se le muestra un mensaje de que la cotización han sido registrada correctamente.
2. Listar	<ol style="list-style-type: none"> 1. El sistema muestra el listado de todas las cotizaciones.
3. Modificar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una cotización. 2. El usuario modifica los datos de la cotización previamente seleccionada 3. El usuario guarda la información 4. Los cambios realizados son actualizados en el sistema
4. Eliminar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una cotización 2. El usuario solicita eliminar cotización. 3. El usuario acepta la confirmación de eliminación. 4. La cotización es actualizada en el sistema. 5. Al usuario se le muestra un listado de todas las cotizaciones.
5. Ver	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una cotización 2. El sistema muestra al usuario el detalle de la cotización.
Camino alternativo:	

<p>1. Crear</p>	<ul style="list-style-type: none"> • En 1.2, Si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos. • En 1.3, el usuario selecciona aceptar con lo cual el sistema le muestra el listado de todas las cotizaciones.
<p>3. Modificar</p>	<ul style="list-style-type: none"> • En 3.3, si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos.
<p>4. Eliminar</p>	<ul style="list-style-type: none"> • En 4.3, el usuario cancela la confirmación de eliminación de la cotización, evitando la actualización de este en el sistema • En 4.4, si la cotización tienen registros asociados el sistema muestra un mensaje de error al usuario, no permitiendo la actualización del mismo.

Fuente: Los Autores

2.1.4.2.5 Gestionar Cosecha

Figura 2.8 Caso de Uso Gestionar Cosecha.



Fuente: Los Autores

Tabla 2. 13 Descripción del Caso de Uso Gestionar Cosecha

<p>Casos de uso:</p>	<p>Gestionar Cosecha</p>
-----------------------------	---------------------------------

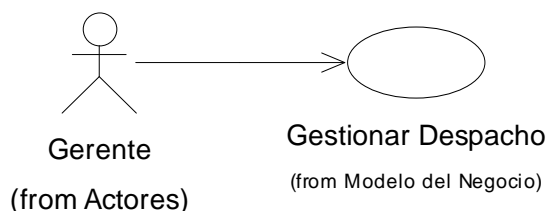
Descripción:	Permite la gestión de cosechas
Actor:	Gerente
Precondiciones:	<ul style="list-style-type: none"> • Se debe obtener la información necesaria de la cosecha. • Se debe seleccionar la piscina a la cual se va a cosechar.
Poscondiciones:	El sistema despliega el listado de cosechas existentes.
Camino básico:	
1. Crear	<ol style="list-style-type: none"> 1. El usuario ingresa la información de la nueva cosecha. 2. El usuario guarda la información. 3. Al usuario se le muestra un mensaje de que la cosecha ha sido registrada correctamente.
2. Listar	<ol style="list-style-type: none"> 1. El sistema muestra el listado de todas las cosechas.
3. Modificar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una cosecha. 2. El usuario modifica los datos de la cosecha previamente seleccionada. 3. El usuario guarda la información. 4. Los cambios realizados son actualizados en el sistema.
4. Eliminar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una cosecha. 2. El usuario solicita eliminar cosecha. 3. El usuario acepta la confirmación de eliminación. 4. La cosecha es actualizada en el sistema. 5. Al usuario se le muestra un listado de todas las cosechas.
5. Ver	<ol style="list-style-type: none"> 1. El usuario previamente debe haber

	<p>seleccionado una cosecha.</p> <p>2. El sistema muestra al usuario el detalle de la Cosecha.</p>
6. Agregar Detalle	<p>1. El usuario para poder crear una cosecha debe agregar sus respectivos detalles.</p> <p>2. Al usuario se le muestra una lista de los detalles.</p>
7. Asignar Cotización	<p>1. El usuario asigna una cotización.</p>
Camino alternativo:	
1. Crear	<ul style="list-style-type: none"> • En 1.2, Si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos. • En 1.3, el usuario selecciona aceptar con lo cual el sistema le muestra el despacho a crear.
3. Modificar	<ul style="list-style-type: none"> • En 3.3, si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos.
4. Eliminar	<ul style="list-style-type: none"> • En 4.3, el usuario cancela la confirmación de eliminación de la cosecha, evitando la actualización de este en el sistema • En 4.4, si la cosecha tiene registros asociados el sistema muestra un mensaje de error al usuario, no permitiendo la actualización del mismo
6. Agregar Detalle	<ul style="list-style-type: none"> • En 6.1, si no se agrega un detalle se muestra un mensaje de error, no permitiendo la creación de una cosecha.
7. Asignar Cotización	<ul style="list-style-type: none"> • En 7.1, si el usuario no asigna una cotización no se impide la creación.

Fuente: Los Autores

2.1.4.2.6 Gestionar Despacho

Figura 2.9 Caso de Uso Gestionar Despacho



Fuente: Los Autores

Tabla 2. 14 Descripción del Caso de Uso Gestionar Despacho

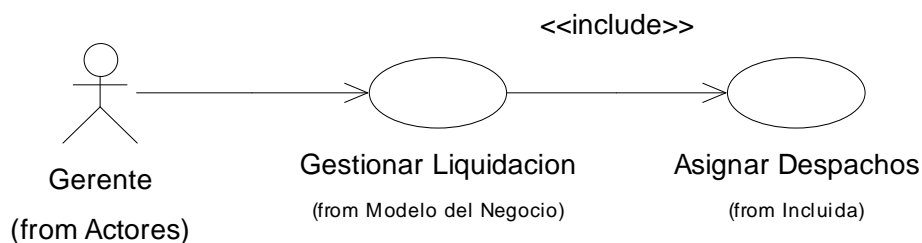
Casos de uso:	Gestionar Despacho
Descripción:	Permite la gestión de despachos
Actor:	Gerente
Precondiciones:	Cuando una Cosecha es creada automáticamente se crea su despacho correspondiente.
Poscondiciones:	<ul style="list-style-type: none"> • El despacho es registrado en el sistema. • El usuario podrá seleccionar un despacho para modificar, eliminar o ver su detalle. • El sistema despliega el listado de despachos existentes.
Camino básico:	
1. Crear	<ol style="list-style-type: none"> 1. El usuario ingresa la información del nuevo despacho. 2. El usuario guarda la información. 3. Al usuario se le muestra un mensaje de que el despacho ha sido registrado correctamente.
2. Listar	<ol style="list-style-type: none"> 1. El sistema muestra el listado de todos los despachos realizados.
3. Modificar	<ol style="list-style-type: none"> 1. El usuario modifica los datos del despacho

	<p>previamente seleccionado.</p> <ol style="list-style-type: none"> 2. El usuario guarda la información. 3. Los cambios realizados son actualizados en el sistema.
4. Eliminar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado un despacho. 2. El usuario solicita eliminar un despacho. 3. El usuario acepta la confirmación de eliminación. 4. El despacho es actualizado en el sistema
5. Ver	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado un despacho. 2. El sistema muestra al usuario el detalle del despacho.
Camino alternativo:	
1. Crear	<ul style="list-style-type: none"> • En 1.2, si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos. • En 1.3, el usuario selecciona aceptar con lo cual el sistema le muestra el listado de todos los despachos.
3. Modificar	<ul style="list-style-type: none"> • En 3.2, si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos.
4. Eliminar	<ul style="list-style-type: none"> • En 4.3, el usuario cancela la confirmación de eliminación de un despacho, evitando la actualización de este en el sistema. • En 4.4, si un despacho tiene registros asociados el sistema muestra un mensaje de error al usuario, no permitiendo la actualización del mismo.

Fuente: Los Autores

2.1.4.2.7 Gestionar Liquidación

Figura 2.10 Caso de Uso Gestionar Liquidación



Fuente: Los Autores

Tabla 2. 15 Descripción del Caso de Uso Gestionar Liquidación

Casos de uso:	Gestionar Liquidación
Descripción:	Permite la gestión de liquidaciones
Actor:	Gerente
Precondiciones:	<ul style="list-style-type: none"> • Obtener la información concerniente a las liquidaciones. • Obtener información procedente del correspondiente despacho.
Poscondiciones:	El sistema muestra el listado de todas las liquidaciones existentes.
Camino básico:	
1. Crear	<ol style="list-style-type: none"> 1. El usuario ingresa la información de la nueva liquidación. 2. El usuario guarda la información. 3. Al usuario se le muestra un mensaje de que la liquidación ha sido registrada correctamente.
2. Listar	<ol style="list-style-type: none"> 1. El sistema muestra el listado de todas las liquidaciones.

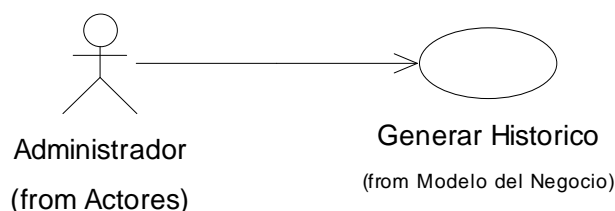
3. Modificar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una liquidación. 2. El usuario modifica los datos de la liquidación previamente seleccionada. 3. El usuario guarda la información. 4. Los cambios realizados son actualizados en el sistema.
4. Eliminar	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una liquidación. 2. El usuario solicita eliminar liquidación. 3. El usuario acepta la confirmación de eliminación. 4. La liquidación es actualizada en el sistema.
5. Ver	<ol style="list-style-type: none"> 1. El usuario previamente debe haber seleccionado una liquidación. 2. El sistema muestra al usuario el detalle de la liquidación.
6. Asignar despacho	<ol style="list-style-type: none"> 1. El usuario debe seleccionar un despacho para poder realizar la liquidación.
Camino alternativo:	
1. Crear	<ul style="list-style-type: none"> • En 1.2, Si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos. • En 1.3, el usuario selecciona aceptar con lo cual el sistema le muestra el listado de todas las liquidaciones.
3. Modificar	<ul style="list-style-type: none"> • En 3.3, si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos.
4. Eliminar	<ul style="list-style-type: none"> • En 4.3, el usuario cancela la confirmación de eliminación de la liquidación, evitando la actualización de este en el sistema

	<ul style="list-style-type: none"> En 4.4, si la liquidación tiene registros asociados el sistema muestra un mensaje de error al usuario, no permitiendo la actualización del mismo.
6. Asignar Despacho	<ul style="list-style-type: none"> En 6.1, si no se asigna despacho se mostrara un mensaje de error, no permitiendo crear la liquidación.

Fuente: Los Autores

2.1.4.2.8 Generar Histórico

Figura 2.11 Caso de Uso Generar Histórico



Fuente: Los Autores

Tabla 2. 16 Descripción del Caso de Uso Generar Histórico

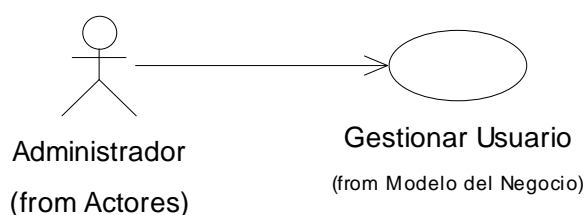
Casos de uso:	Generar Histórico
Descripción:	Permite la generación de históricos.
Actor:	Administrador
Precondiciones:	Se deben poseer registros de cosecha, empacadora y piscina.
Poscondiciones:	El sistema despliega el listado de históricos dependiendo del tipo elegido.
Camino básico:	
1. Histórico.	1. El usuario previamente debe haber

	<p>seleccionado un histórico.</p> <p>2. El sistema muestra al usuario el detalle del histórico.</p>
--	---

Fuente: Los Autores

2.1.4.2.9 Gestionar Usuario

Figura 2.12 Caso de Uso Gestionar Usuario



Fuente: Los Autores

Tabla 2. 17 Descripción del Caso de Uso Gestionar Usuario

Casos de uso:	Gestionar Usuario
Descripción:	Permite la gestión de usuarios.
Actor:	Administrador
Precondiciones:	Se debe obtener la información necesaria del usuario.
Poscondiciones:	El sistema despliega el listado de Usuarios existentes.
Camino básico:	
1. Crear	<ol style="list-style-type: none"> 1. El Administrador ingresa la información del nuevo usuario. 2. El Administrador guarda la información. 3. Al Administrador se le muestra un mensaje de que el usuario ha sido registrado correctamente
2. Listar	<ol style="list-style-type: none"> 1. El sistema muestra el listado de todos los Usuarios.
3. Modificar	<ol style="list-style-type: none"> 1. El Administrador modifica los datos del usuario previamente seleccionado.

	<ol style="list-style-type: none"> 2. El Administrador guarda la información 3. Los cambios realizados son actualizados en el sistema.
4. Eliminar	<ol style="list-style-type: none"> 1. El Administrador previamente debe haber seleccionado un usuario. 2. El Administrador solicita eliminar usuario. 3. El Administrador acepta la confirmación de eliminación. 4. Al Administrador se le muestra un listado de todos los usuarios.
5. Ver	<ol style="list-style-type: none"> 1. El Administrador previamente debe haber seleccionado un usuario. 2. El sistema muestra al Administrador el detalle del usuario.
Camino alternativo:	
1. Crear	<ul style="list-style-type: none"> • En 1.2, si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos. • En 1.3, el Administrador selecciona aceptar con lo cual el sistema le muestra el listado de todos los Usuarios.
3. Modificar	<ul style="list-style-type: none"> • En 3.3, si los datos ingresados no son validos se despliega mensajes de error en los campos incorrectos.
4. Eliminar	<ul style="list-style-type: none"> • En 4.3, el Administrador cancela la confirmación de eliminación del Usuario, evitando la actualización de este en el sistema • En 4.4, si el Usuario tiene registros asociados el sistema muestra un mensaje de error al Administrador, no permitiendo la

	actualización del mismo.
--	--------------------------

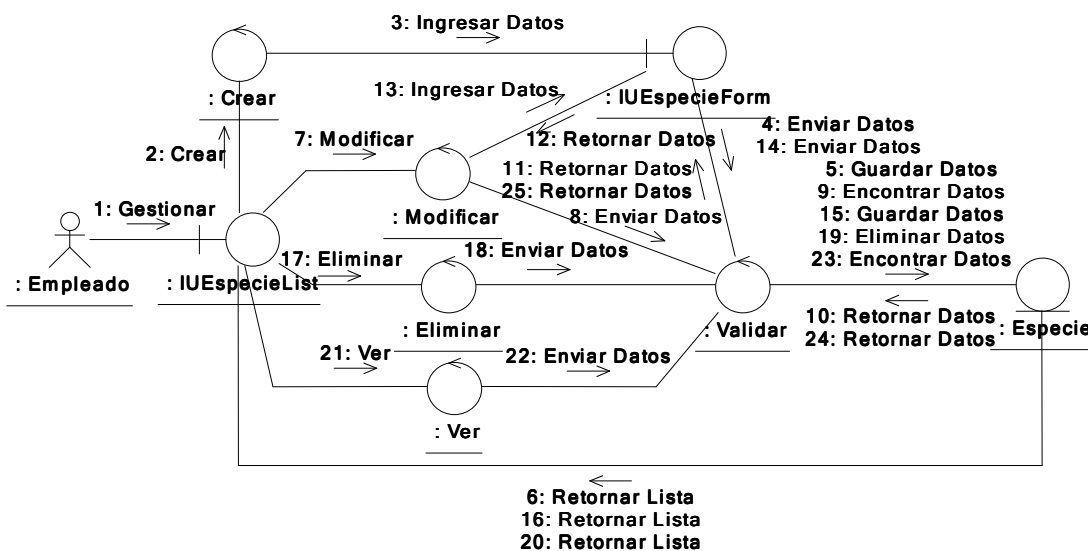
Fuente: Los Autores

2.1.5 DIAGRAMAS DE COLABORACION.

A continuación se presentan los diagramas de colaboración, y los flujos de sucesos correspondientes a la realización de los casos de usos del sistema

2.1.5.1 Diagrama de Colaboración del Caso de Uso Gestionar Especie.

Figura 2.13 Diagrama de Colaboración del Caso de Uso Gestionar Especie



Fuente: Los Autores

Tabla 2. 18 Diagrama de Colaboración del Caso de Uso Gestionar Especie

Flujo de Sucesos: Diagrama de Colaboración del Caso de Uso Gestionar Especie

1. Solicitud del usuario para gestionar especie.
2. Se selecciona la operación crear especie.
3. Se procede a ingresar los datos respectivos de la especie.
4. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 5, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
5. Se procede a guardar los datos en la base de datos.
6. Se retorna al usuario la lista de especies.
7. Se selecciona la operación modificar especie.
8. Se procede a enviar el código de la especie a validar, en caso de que la información sea correcta se procede al paso 9, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
9. Se procede a encontrar la especie solicitada.
10. Se valida la especie encontrada.
11. Se retorna la especie encontrada a la interfaz de Especie.
12. Se muestra el resultado de la operación modificar al usuario.
13. Se procede a ingresar los datos respectivos de la especie.
14. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 15, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
15. Se procede a guardar los datos en la base de datos.
16. Se retorna al usuario la lista de especies.
17. Se selecciona la operación eliminar especie.
18. Se procede a enviar el código de la especie a validar, en caso de que la información sea correcta se procede al paso 19, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
19. Se procede a eliminar la especie de la base de datos.
20. Se retorna al usuario la lista de especies.
21. Se selecciona la operación ver especie.
22. Se procede a enviar el código de la especie a validar, en caso de que

la información sea correcta se procede al paso 23, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.

23. Se procede a encontrar la especie solicitada.

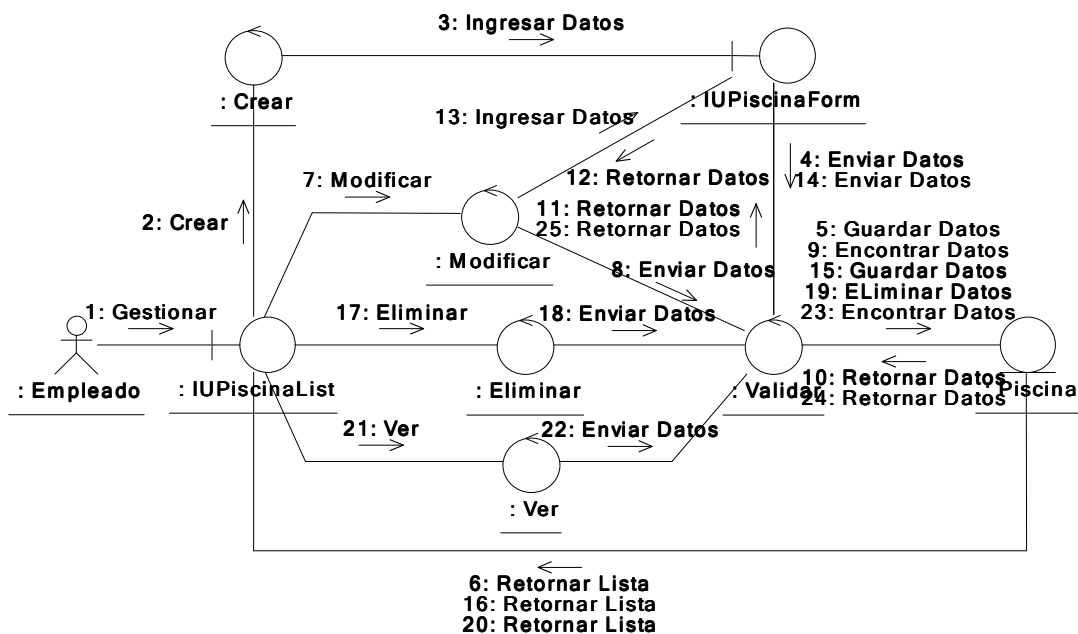
24. Se valida la especie encontrada.

25. Se retorna la especie encontrada a la interfaz de Especie.

Fuente: Los Autores

2.1.5.2 Diagrama de Colaboración del Caso de Uso Gestionar Piscina.

Figura 2.14 Diagrama de Colaboración del Caso de Uso Gestionar Piscina



Fuente: Los Autores

Tabla 2. 19 Diagrama de Colaboración del Caso de Uso Gestionar Piscina

Flujo de Sucesos: Diagrama de Colaboración del Caso de Uso Gestionar Piscina

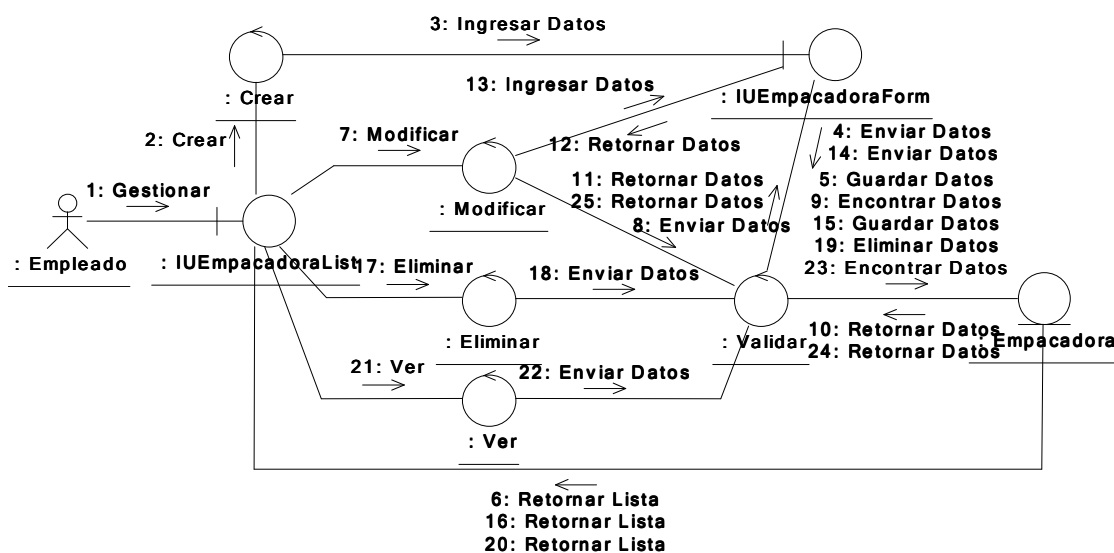
1. Solicitud del usuario para gestionar piscina.
2. Se selecciona la operación crear piscina.
3. Se procede a ingresar los datos respectivos de la piscina.
4. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 5, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
5. Se procede a guardar los datos en la base de datos.
6. Se retorna al usuario la lista de piscinas.
7. Se selecciona la operación modificar piscina.
8. Se procede a enviar el código de la piscina a validar, en caso de que la información sea correcta se procede al paso 9, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
9. Se procede a encontrar la piscina solicitada.
10. Se valida la piscina encontrada.
11. Se retorna la piscina encontrada a la interfaz de Piscina.
12. Se muestra el resultado de la operación modificar al usuario.
13. Se procede a ingresar los datos respectivos de la piscina.
14. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 15, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
15. Se procede a guardar los datos en la base de datos.
16. Se retorna al usuario la lista de piscinas.
17. Se selecciona la operación eliminar piscina.
18. Se procede a enviar el código de la piscina a validar, en caso de que la información sea correcta se procede al paso 18, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
19. Se procede a eliminar la piscina de la base de datos.
20. Se retorna al usuario la lista de piscinas.
21. Se selecciona la operación ver piscina.

22. Se procede a enviar el código de la piscina a validar, en caso de que la información sea correcta se procede al paso 23, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
23. Se procede a encontrar la piscina solicitada.
24. Se valida la piscina encontrada.
25. Se retorna la piscina encontrada a la interfaz de Piscina.

Fuente: Los Autores

2.1.5.3 Diagrama de Colaboración del Caso de Uso Gestionar Empacadora.

Figura 2.15 Diagrama de Colaboración del Caso de Uso Gestionar Empacadora



Fuente: Los Autores

Tabla 2. 20 Diagrama de Colaboración del Caso de Uso Gestionar Empacadora

Flujo de Sucesos: Diagrama de Colaboración del Caso de Uso Gestionar Empacadora

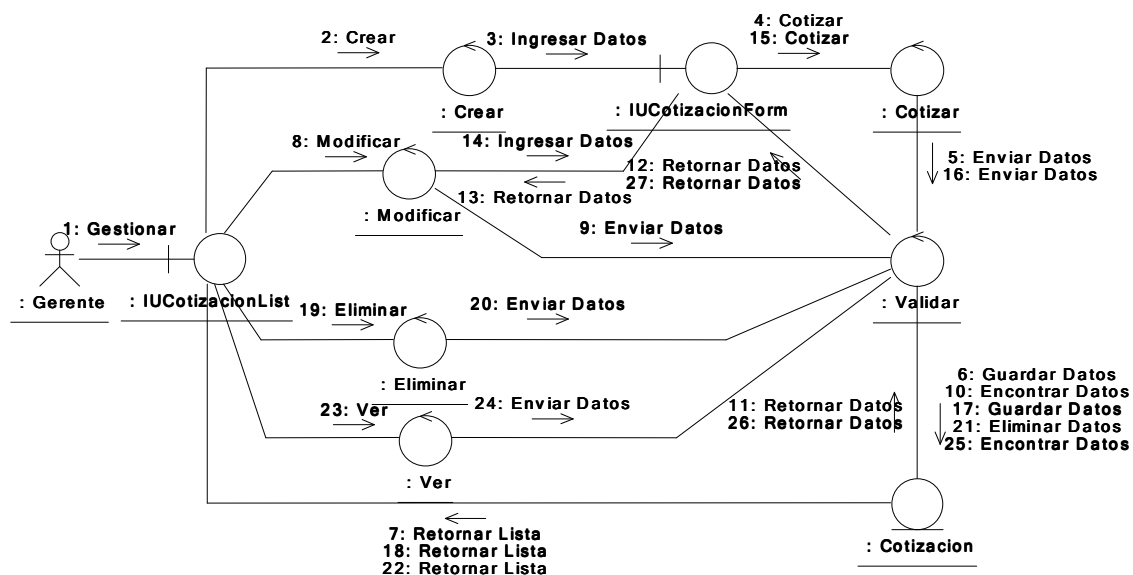
1. Solicitud del usuario para gestionar empacadora.
2. Se selecciona la operación crear empacadora.
3. Se procede a ingresar los datos respectivos de la empacadora.
4. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 5, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
5. Se procede a guardar los datos en la base de datos.
6. Se retorna al usuario la lista de empacadoras.
7. Se selecciona la operación modificar empacadora.
8. Se procede a enviar el código de la empacadora a validar, en caso de que la información sea correcta se procede al paso 9, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
9. Se procede a encontrar la empacadora solicitada.
10. Se valida la empacadora encontrada.
11. Se retorna la empacadora encontrada a la interfaz de Empacadora.
12. Se muestra el resultado de la operación modificar al usuario.
13. Se procede a ingresar los datos respectivos de la empacadora.
14. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 15, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
15. Se procede a guardar los datos en la base de datos.
16. Se retorna al usuario la lista de empacadoras.
17. Se selecciona la operación eliminar empacadora.
18. Se procede a enviar el código de la empacadora a validar, en caso de que la información sea correcta se procede al paso 19, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
19. Se procede a eliminar la empacadora de la base de datos.
20. Se retorna al usuario la lista de empacadoras.
21. Se selecciona la operación ver empacadora.
22. Se procede a enviar el código de la empacadora a validar, en caso de que la información sea correcta se procede al paso 23, caso contrario

- se presenta un mensaje al usuario de la verificación del código enviado.
23. Se procede a encontrar la empaedora solicitada.
 24. Se valida la empaedora encontrada.
 25. Se retorna la empaedora encontrada a la interfaz de Empaedora.

Fuente: Los Autores

2.1.5.4 Diagrama de Colaboración del Caso de Uso Gestionar Cotización.

Figura 2.16 Diagrama de Colaboración del Caso de Uso Gestionar Cotización



Fuente: Los Autores

Tabla 2. 21 Diagrama de Colaboración del Caso de Uso Gestionar Cotización

Flujo de Sucesos: Diagrama de Colaboración del Caso de Uso Gestionar Cotización

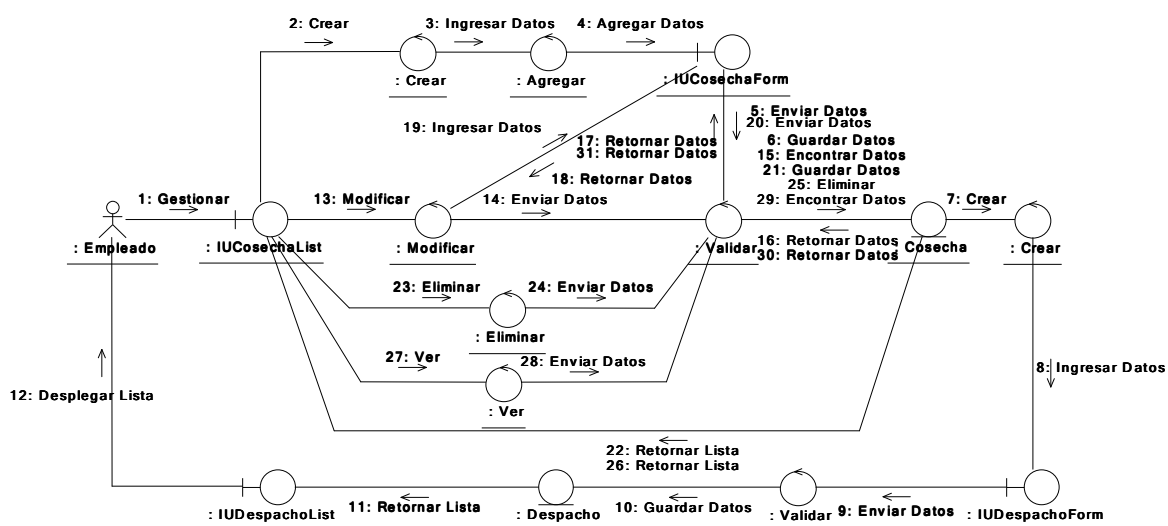
1. Solicitud del usuario para gestionar cotización.
2. Se selecciona la operación crear cotización.
3. Se procede a ingresar los datos respectivos de la cotización.
4. Se realiza la operación cotizar.
5. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 6, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
6. Se procede a guardar los datos en la base de datos.
7. Se retorna al usuario la lista de cotizaciones.
8. Se selecciona la operación modificar cotización.
9. Se procede a enviar el código de la cotización a validar, en caso de que la información sea correcta se procede al paso 10, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
10. Se procede a encontrar la cotización solicitada.
11. Se valida la cotización encontrada.
12. Se retorna la cotiza encontrada a la interfaz de Cotización.
13. Se muestra el resultado de la operación modificar al usuario.
14. Se procede a ingresar los datos respectivos de la cotización.
15. Se realiza la operación cotizar en caso de que sea necesario.
16. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 17, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
17. Se procede a guardar los datos en la base de datos.
18. Se retorna al usuario la lista de cotizaciones.
19. Se selecciona la operación eliminar cotización.
20. Se procede a enviar el código de la cotización a validar, en caso de que la información sea correcta se procede al paso 21, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
21. Se procede a eliminar la cotización de la base de datos.

22. Se retorna al usuario la lista de cotizaciones.
23. Se selecciona la operación ver cotización.
24. Se procede a enviar el código de la cotización a validar, en caso de que la información sea correcta se procede al paso 25, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
25. Se procede a encontrar la cotización solicitada.
26. Se valida la cotización encontrada.
27. Se retorna la cotización encontrada a la interfaz de Cotización.

Fuente: Los Autores

2.1.5.5 Diagrama de Colaboración del Caso de Uso Gestionar Cosecha.

Figura 2.17 Diagrama de Colaboración del Caso de Uso Gestionar Cosecha



Fuente: Los Autores

Tabla 2. 22 Diagrama de Colaboración del Caso de Uso Gestionar Cosecha

Flujo de Sucesos: Diagrama de Colaboración del Caso de Uso Gestionar

Cosecha

1. Solicitud del usuario para gestionar cosecha.
2. Se selecciona la operación crear cosecha.
3. Se procede a ingresar los datos respectivos de la cosecha.
4. Se procede a agregar los detalles de la cosecha.
5. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 6, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
6. Se procede a guardar los datos en la base de datos.
7. Posteriormente se muestra la interfaz de Despacho y se procede a crear el despacho.
8. Se procede a ingresar los datos respectivos del despacho.
9. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 10, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
10. Se procede a guardar los datos en la base de datos.
11. Se retorna al usuario la lista de despachos.
12. Se despliega al usuario la lista de despacho a través de la interfaz de Despacho.
13. Solicitud del usuario para gestionar cosecha, se selecciona la operación modificar cosecha.
14. Se procede a enviar el código de la cosecha a validar, en caso de que la información sea correcta se procede al paso 15, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
15. Se procede a encontrar la cosecha solicitada.
16. Se valida la cosecha encontrada.
17. Se retorna la cosecha encontrada a la interfaz de Cosecha.
18. Se muestra el resultado de la operación modificar al usuario.
19. Se procede a ingresar los datos respectivos de la cosecha.
20. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 21, caso contrario se

presenta un mensaje al usuario de la verificación de los datos ingresados.

21. Se procede a guardar los datos en la base de datos.
22. Se retorna al usuario la lista de cosechas.
23. Se selecciona la operación eliminar cosecha.
24. Se procede a enviar el código de la cosecha a validar, en caso de que la información sea correcta se procede al paso 25, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
25. Se procede a eliminar la cosecha de la base de datos.
26. Se retorna al usuario la lista de cosechas.
27. Se selecciona la operación ver cosecha.
28. Se procede a enviar el código de la cosecha a validar, en caso de que la información sea correcta se procede al paso 29, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
29. Se procede a encontrar la cosecha solicitada.
30. Se valida la cosecha encontrada.
31. Se retorna la cosecha encontrada a la interfaz de Cosecha.

Fuente: Los Autores

2.1.5.6 Diagrama de Colaboración del Caso de Uso Gestionar Despacho.

Figura 2.18 Diagrama de Colaboración del Caso de Uso Gestionar Despacho

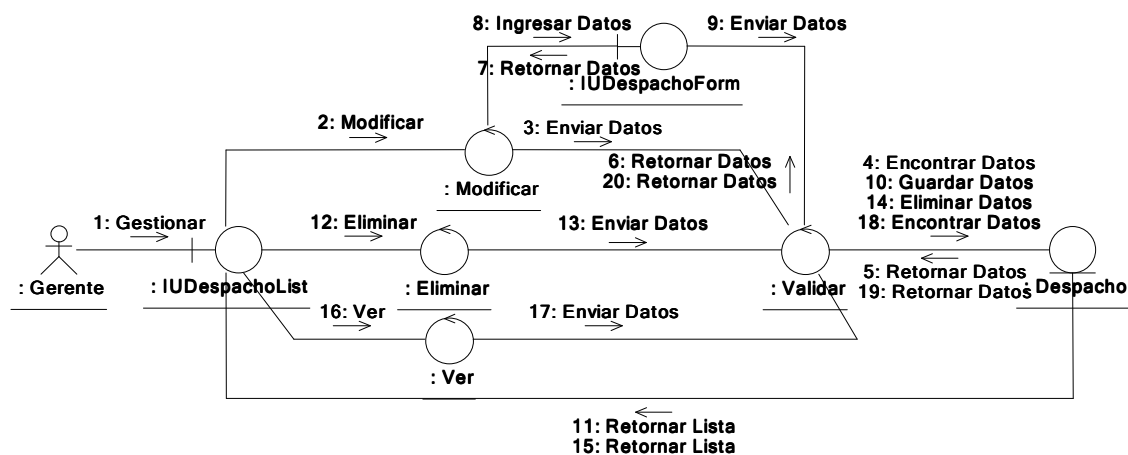


Tabla 2. 23 Diagrama de Colaboración del Caso de Uso Gestionar Despacho

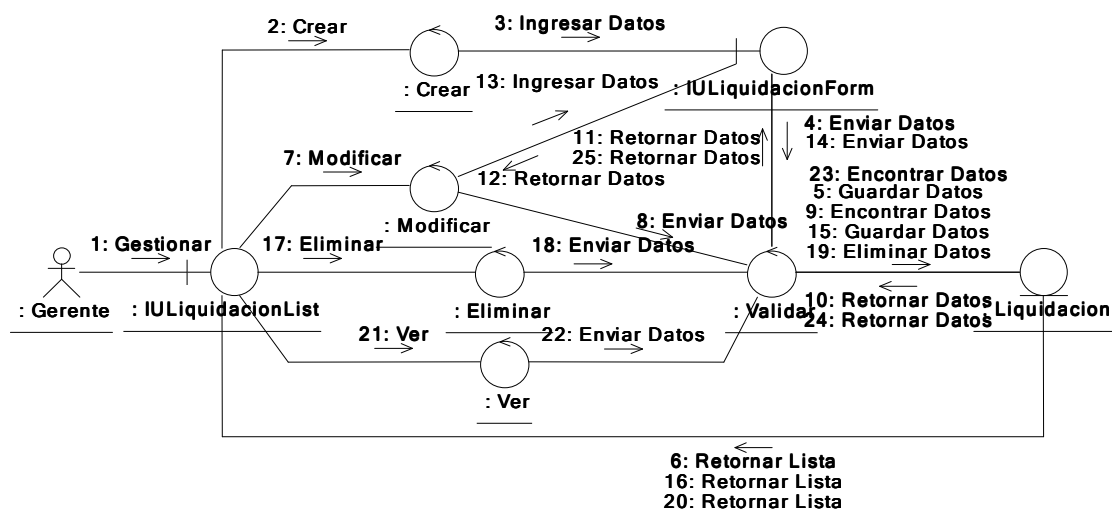
Flujo de Sucesos: Diagrama de Colaboración del Caso de Uso Gestionar Despacho
<ol style="list-style-type: none">1. Solicitud del usuario para gestionar despacho.2. Se selecciona la operación modificar despacho.3. Se procede a enviar el código del despacho a validar, en caso de que la información sea correcta se procede al paso 4, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.4. Se procede a encontrar el despacho solicitado.5. Se valida el despacho encontrado.6. Se retorna el despacho encontrado a la interfaz de Despacho.7. Se muestra el resultado de la operación al usuario.8. Se procede a ingresar los datos respectivos del despacho.9. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 10, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.10. Se procede a guardar los datos en la base de datos.11. Se retorna al usuario la lista de despachos.12. Se selecciona la operación eliminar despacho.13. Se procede a enviar el código de la liquidación a validar, en caso de que la información sea correcta se procede al paso 14, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.14. Se procede a eliminar el despacho de la base de datos.15. Se retorna al usuario la lista de despachos16. Se selecciona la operación ver despacho.17. Se procede a enviar el código del despacho a validar, en caso de que la información sea correcta se procede al paso 18, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.18. Se procede a encontrar el despacho solicitado.19. Se valida el despacho encontrado.

20. Se retorna el despacho encontrado a la interfaz de Despacho.

Fuente: Los Autores

2.1.5.7 Diagrama de Colaboración del Caso de Uso Gestionar Liquidación.

Figura 2.19 Diagrama de Colaboración del Caso de Uso Gestionar Liquidación



Fuente: Los Autores

Tabla 2. 24 Diagrama de Colaboración del Caso de Uso Gestionar Liquidación

Flujo de Sucesos: Diagrama de Colaboración del Caso de Uso Gestionar

Liquidación

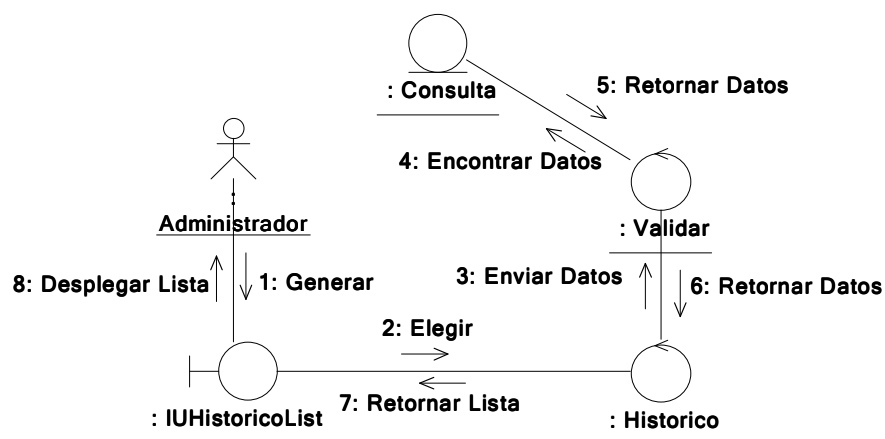
1. Solicitud del usuario para gestionar liquidación.
2. Se selecciona la operación crear liquidación.
3. Se procede a ingresar los datos respectivos de la liquidación.
4. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 5, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
5. Se procede a guardar los datos en la base de datos.
6. Se retorna al usuario la lista de liquidaciones.
7. Se selecciona la operación modificar liquidación.
8. Se procede a enviar el código de la liquidación a validar, en caso de que la información sea correcta se procede al paso 9, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
9. Se procede a encontrar la liquidación solicitada.
10. Se valida la liquidación encontrada.
11. Se retorna la liquidación encontrada a la interfaz de Liquidación.
12. Se muestra el resultado de la operación modificar al usuario.
13. Se procede a ingresar los datos respectivos de la liquidación.
14. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 15, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
15. Se procede a guardar los datos en la base de datos.
16. Se retorna al usuario la lista de liquidaciones.
17. Se selecciona la operación eliminar liquidación.
18. Se procede a enviar el código de la liquidación a validar, en caso de que la información sea correcta se procede al paso 19, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
19. Se procede a eliminar la liquidación de la base de datos.
20. Se retorna al usuario la lista de liquidaciones.
21. Se selecciona la operación ver liquidación.
22. Se procede a enviar el código de la liquidación a validar, en caso de

- que la información sea correcta se procede al paso 23, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
23. Se procede a encontrar la liquidación solicitada.
 24. Se valida la liquidación encontrada.
 25. Se retorna la liquidación encontrada a la interfaz de Liquidación.

Fuente: Los Autores

2.1.5.8 Diagrama de Colaboración del Caso de Uso Generar Histórico.

Figura 2.20 Diagrama de Colaboración del Caso de Uso Generar Histórico



Fuente: Los Autores

Tabla 2. 25 Diagrama de Colaboración del Caso de Uso Generar Histórico

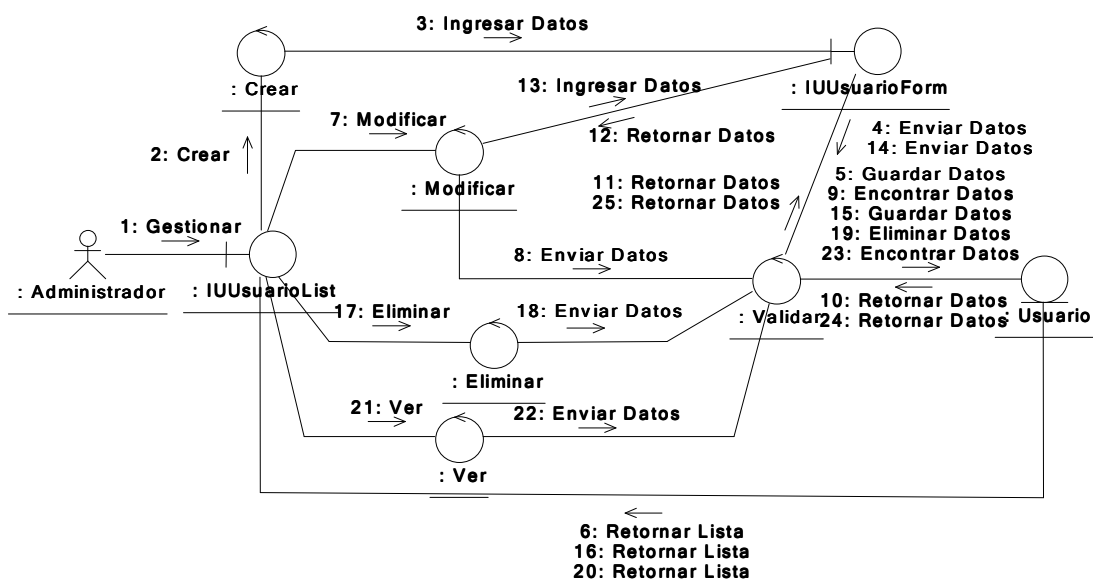
Flujo de Sucesos: Diagrama de Colaboración del Caso de Uso Generar Histórico.

1. Solicitud del usuario para generar histórico.
2. Se selecciona el histórico requerido.
3. Se procede a validar los datos enviados, en caso de que la información sea correcta se procede al paso 4, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
4. Se procede a encontrar los datos en la base de datos.
5. Se valida el histórico encontrado.
6. Se retorna el histórico encontrado.
7. Se retorna el histórico encontrado a la interfaz de Histórico.
8. Se muestra al usuario el histórico retornado.

Fuente: Los Autores

2.1.5.9 Diagrama de Colaboración del Caso de Uso Gestionar Usuario.

Figura 2.21 Diagrama de Colaboración del Caso de Uso Gestionar Usuario



Fuente: Los Autores

Tabla 2. 26 Diagrama de Colaboración del Caso de Uso Gestionar Usuario

Flujo de Sucesos: Diagrama de Colaboración del Caso de Uso Gestionar Usuario

1. Solicitud del usuario para gestionar usuario.
2. Se selecciona la operación crear usuario.
3. Se procede a ingresar los datos respectivos del usuario.
4. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 5, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
5. Se procede a guardar los datos en la base de datos.
6. Se retorna al usuario la lista de usuarios.
7. Se selecciona la operación modificar usuario.
8. Se procede a enviar el código del usuario a validar, en caso de que la información sea correcta se procede al paso 9, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
9. Se procede a encontrar al usuario solicitado.
10. Se valida el usuario encontrado.
11. Se retorna el usuario encontrado a la interfaz de Usuario.
12. Se muestra el resultado de la operación modificar al usuario.
13. Se procede a ingresar los datos respectivos del usuario.
14. Se procede a validar los datos ingresados, en caso de que la información sea correcta se procede al paso 15, caso contrario se presenta un mensaje al usuario de la verificación de los datos ingresados.
15. Se procede a guardar los datos en la base de datos.
16. Se retorna al usuario la lista de usuarios.
17. Se selecciona la operación eliminar usuario.
18. Se procede a enviar el código del usuario a validar, en caso de que la información sea correcta se procede al paso 19, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
19. Se procede a eliminar al usuario de la base de datos.
20. Se retorna al usuario la lista de usuarios.
21. Se selecciona la operación ver usuario.

- 22.** Se procede a enviar el código del usuario a validar, en caso de que la información sea correcta se procede al paso 23, caso contrario se presenta un mensaje al usuario de la verificación del código enviado.
- 23.** Se procede a encontrar al usuario solicitado.
- 24.** Se valida al usuario encontrado.
- 25.** Se retorna el usuario encontrado a la interfaz de Usuario.

Fuente: Los Autores

2.1.6 PAQUETE DE ANÁLISIS

Los paquetes de análisis permiten organizar los documentos del modelo de análisis en grupos manejables.

Para identificarlos se ha tomado en cuenta los siguientes conceptos.

- Están fundamentados en los requerimientos del sistema.
- Se utilizan los casos de usos que representan una separación de intereses dentro del análisis.

De acuerdo a los conceptos anteriores se ha escogido los siguientes paquetes de análisis.

Tabla 2. 27 Paquete de Análisis de Modulo del Sistema

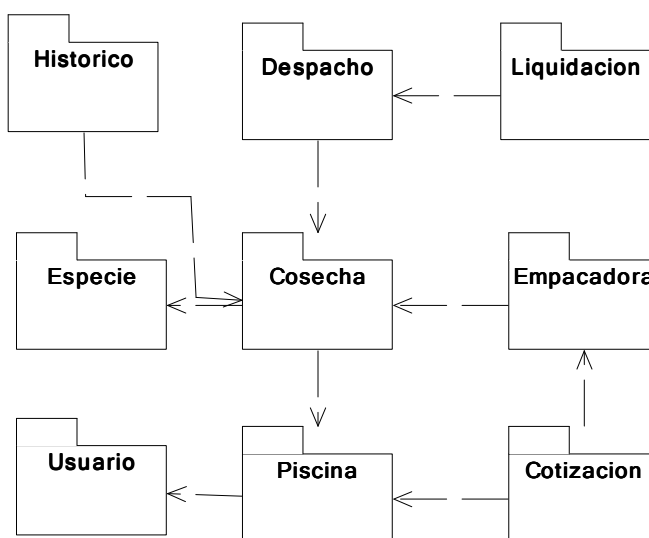
Paquete de Análisis	Caso de Uso
Usuario	Gestionar Usuario
Especie	Gestionar Especie
Cosecha	Gestionar Cosecha
Piscina	Gestionar Piscina
Empacadora	Gestionar Empacadora
Cotización	Gestionar Cotización

Liquidación	Gestionar Liquidación
Histórico	Generar Histórico
Despacho	Gestionar Despacho

Fuente: Los Autores

En la siguiente figura se muestran las relaciones que existen entre los paquetes de análisis del sistema.

Figura 2.22 Dependencia entre los Paquetes del Módulo del Sistema.

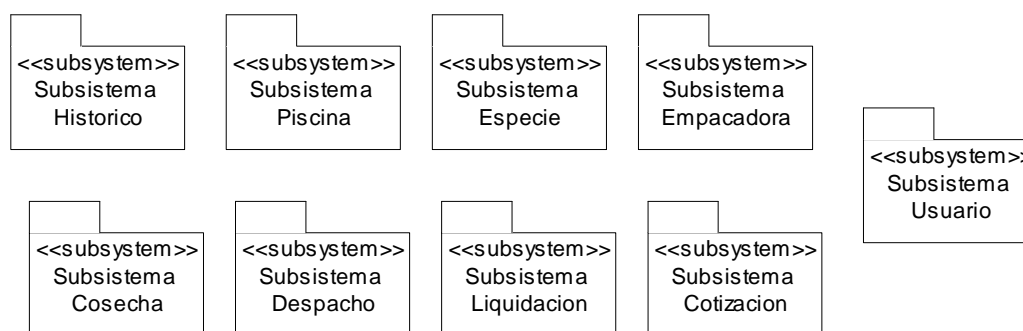


Fuente: Los Autores

2.1.7 SUBSISTEMAS DEL MÓDULO DEL SISTEMA.

Para el desarrollo de este sistema se tomó en cuenta el diseño de nueve subsistemas los cuales tienen un total grado de independencia entre ellos.

Figura 2.23 Subsistemas del Modulo del Sistema



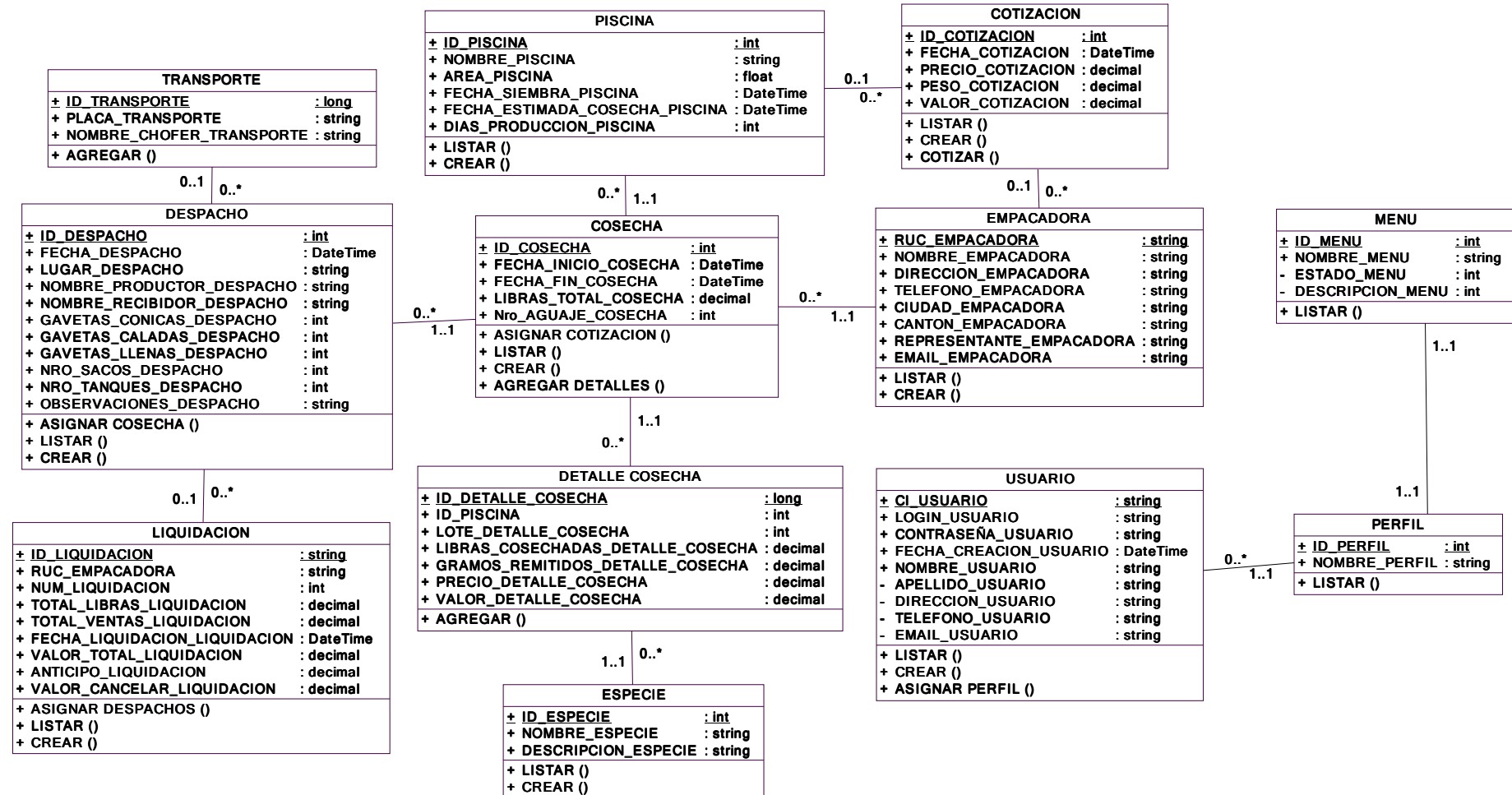
Fuente: Los Autores

2.2 DISEÑO

En el diseño se modela el sistema y se encuentra su forma o arquitectura. Esta parte corresponde al final de la parte de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a la realización de una arquitectura firme y estable.

2.2.1 DISEÑO DE CLASES

Figura 2.24 Diagrama de Clases



Fuente: Los Autores

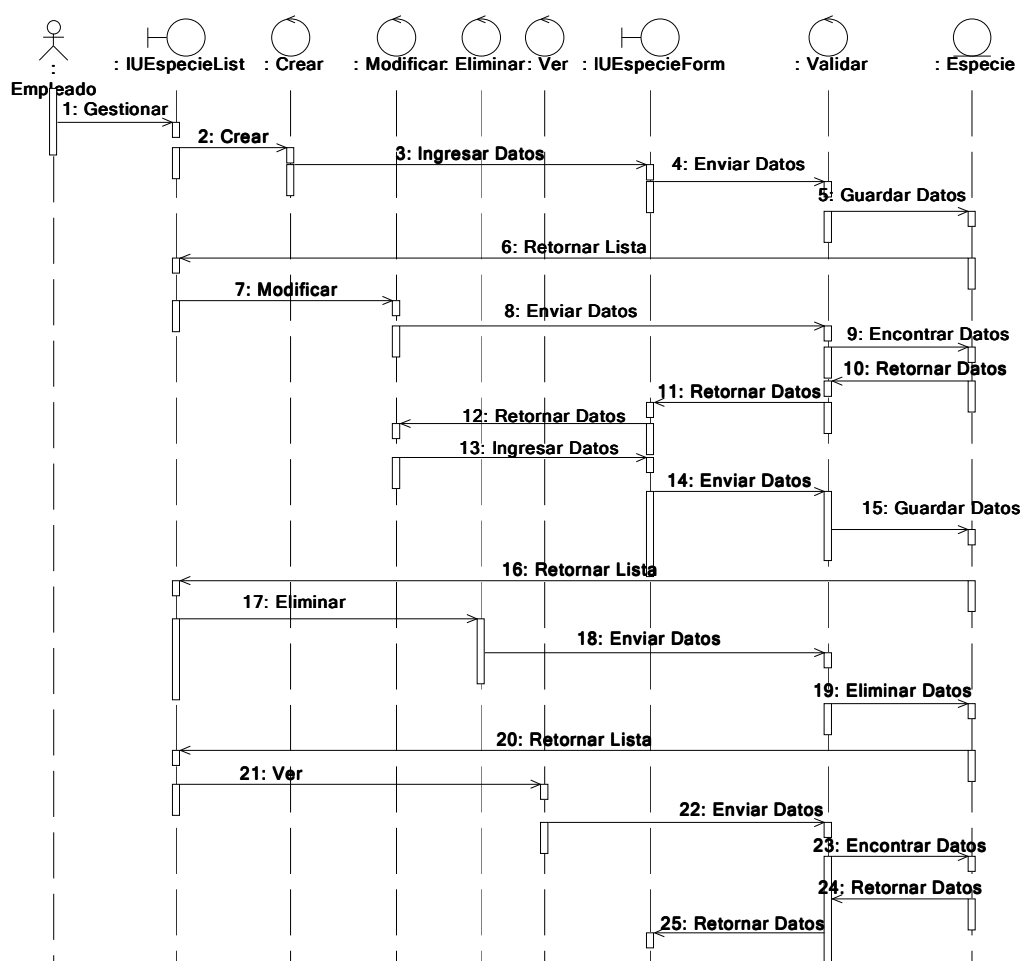
2.2.2 DISEÑO DE CASOS DE USO.

Se realiza en base a los subsistemas identificados durante la etapa de análisis del sistema, considerando que cada subsistema puede tener uno o más casos de uso que interactúan en el sistema.

A continuación se presentan los diagramas de secuencia de los subsistemas de diseño.

2.2.2.1 Subsistema Gestionar Especie.

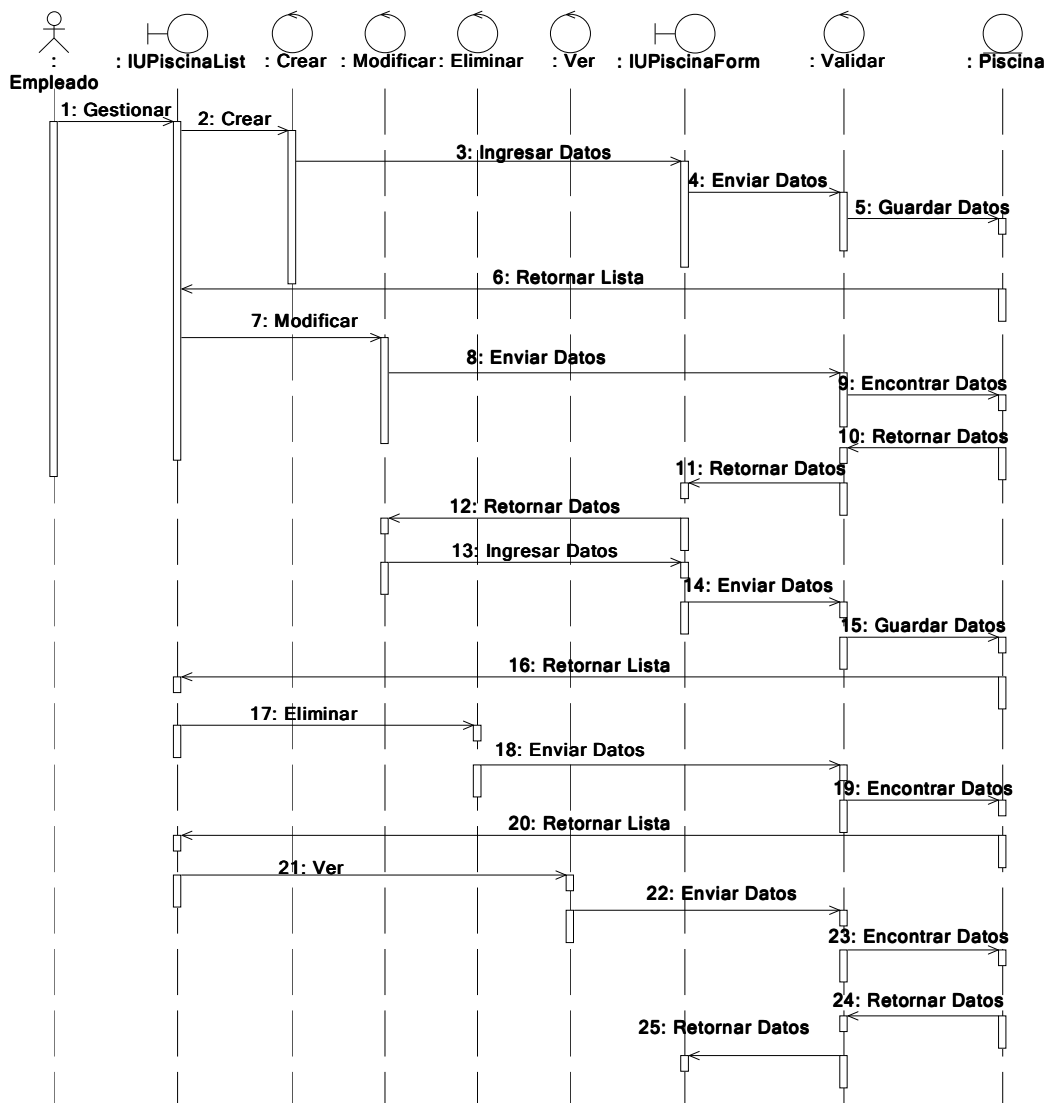
Figura 2.25 Diagrama de Secuencia del caso de uso Gestionar Especie



Fuente: Los Autores

2.2.2.2 Subsistema Gestionar Piscina.

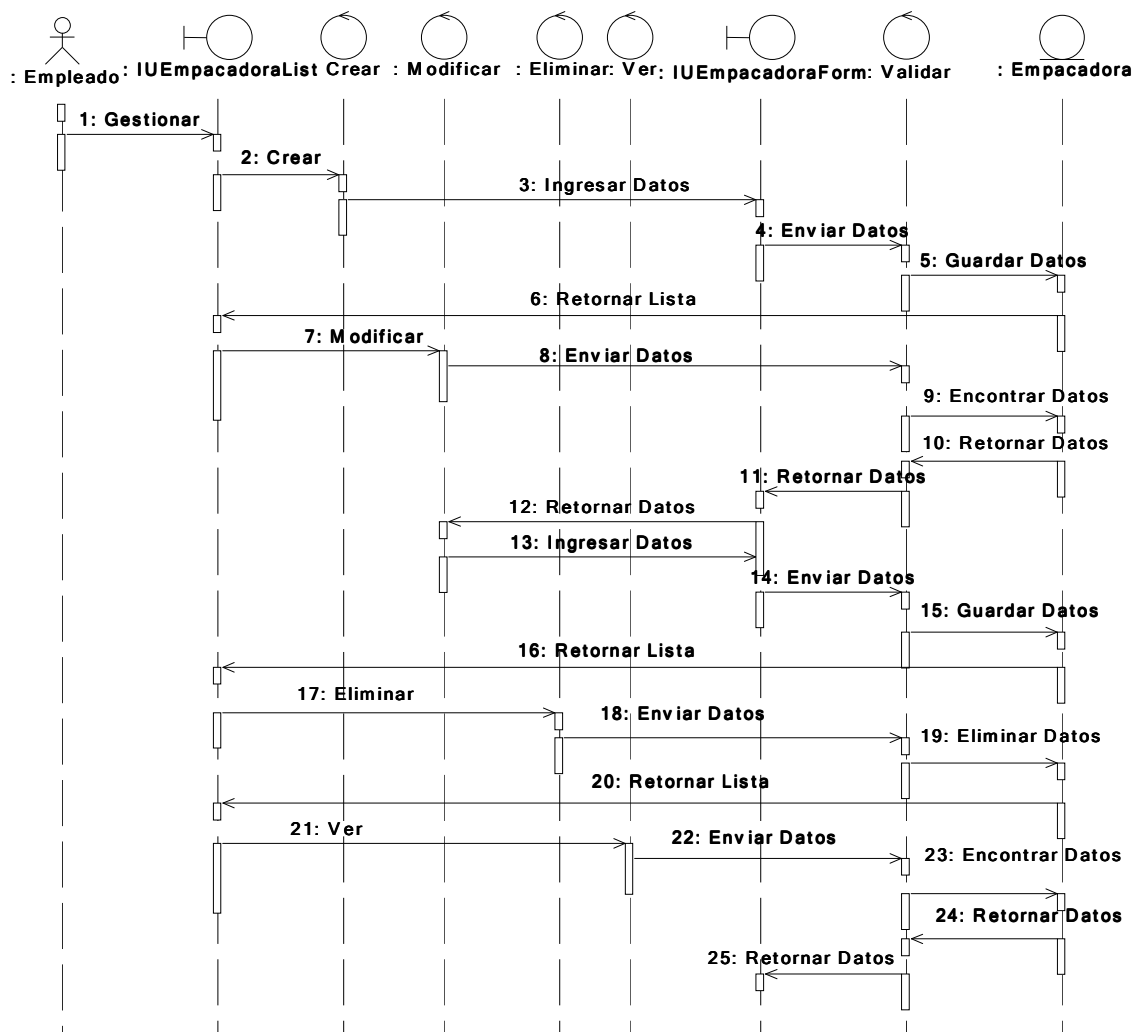
Figura 2.26 Diagrama de Secuencia del caso de uso Gestionar Piscina



Fuente: Los Autores

2.2.2.3 Subsistema Gestionar Empacadora.

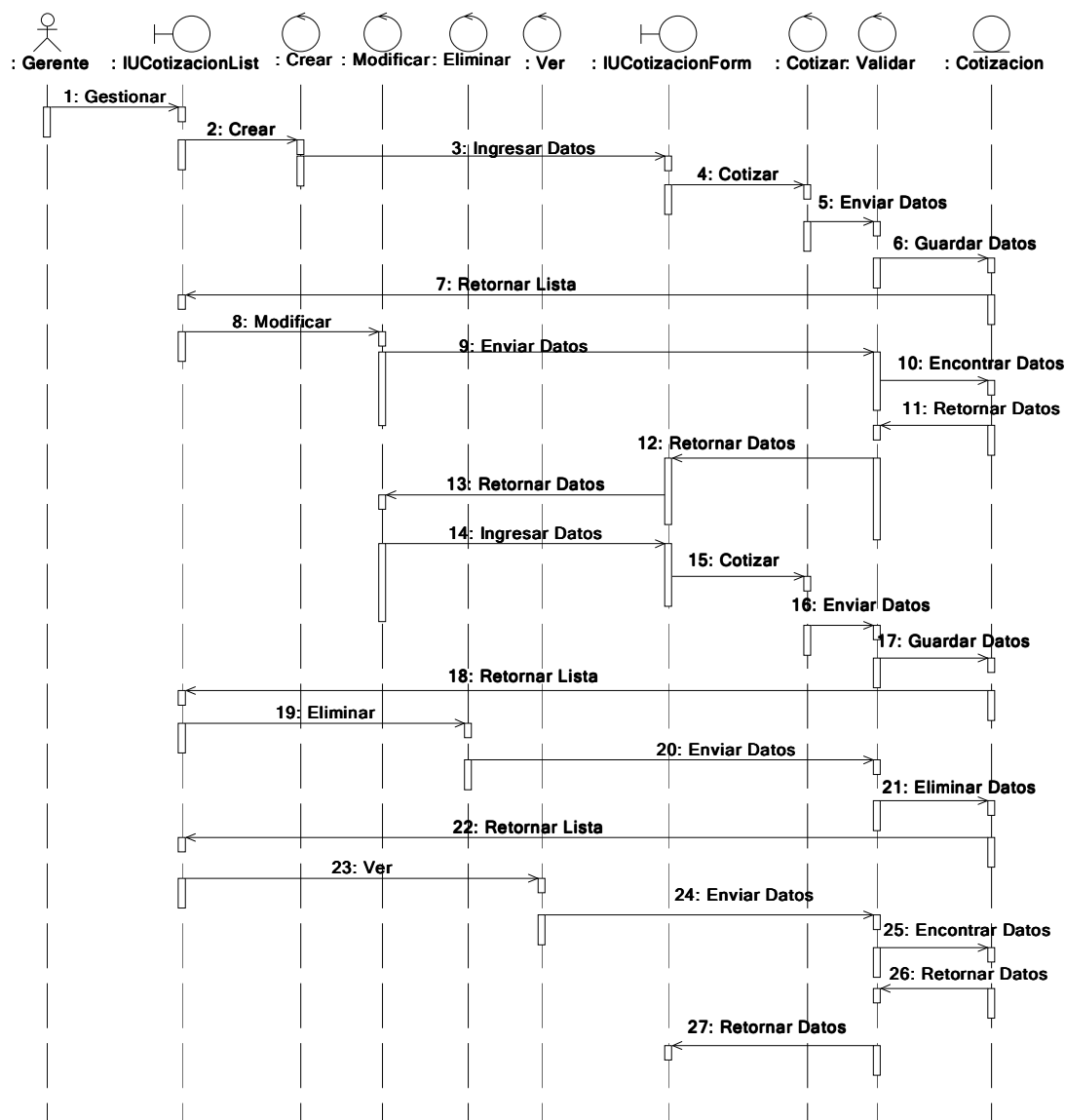
Figura 2.27 Diagrama de Secuencia del caso de uso Gestionar Empacadora



Fuente: Los Autores

2.2.2.4 Subsistema Gestionar Cotización.

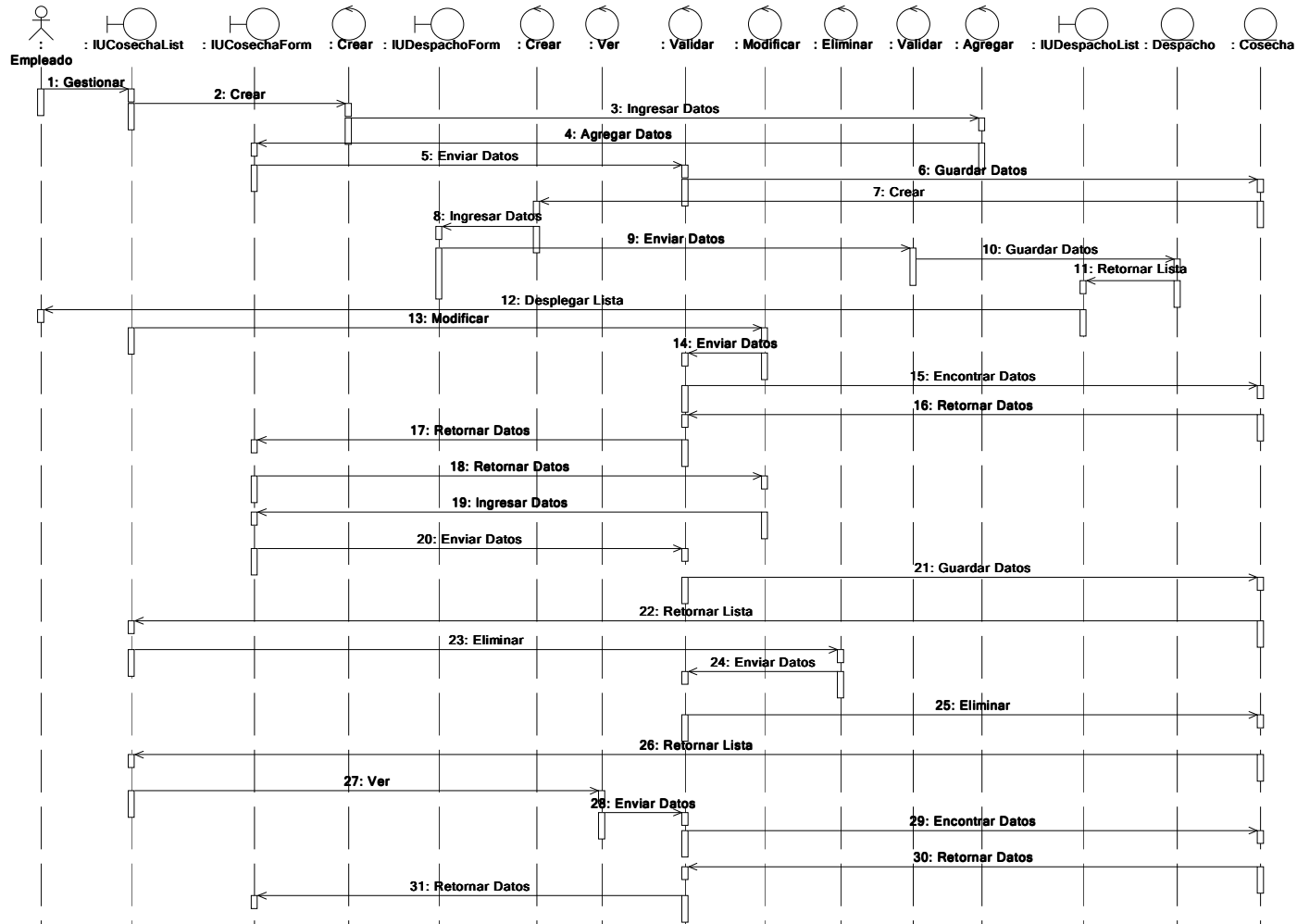
Figura 2.28 Diagrama de Secuencia del caso de uso Gestionar Cotización.



Fuente: Los Autores

2.2.2.5 Subsistema Gestionar Cosecha.

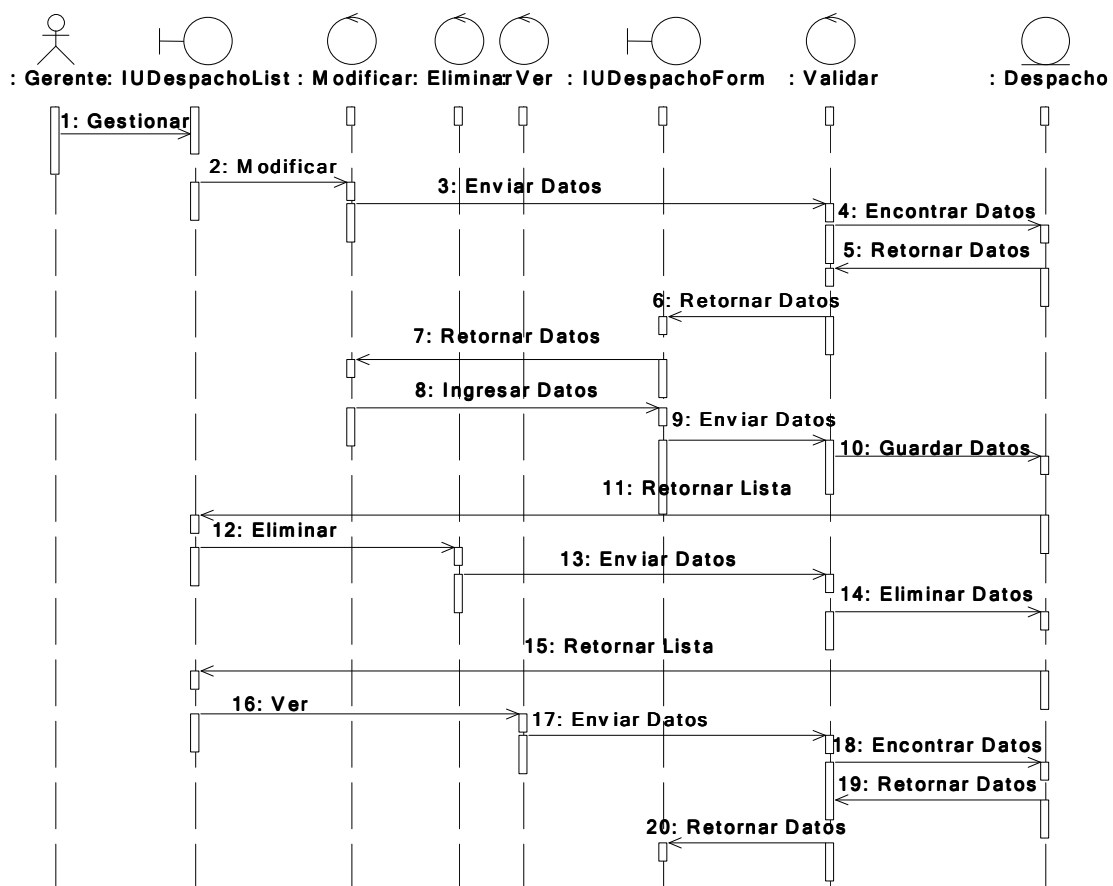
Figura 2.29 Diagrama de Secuencia del caso de uso Gestionar Cosecha.



Fuente: Los Autores

2.2.2.6 Subsistema Gestionar Despacho.

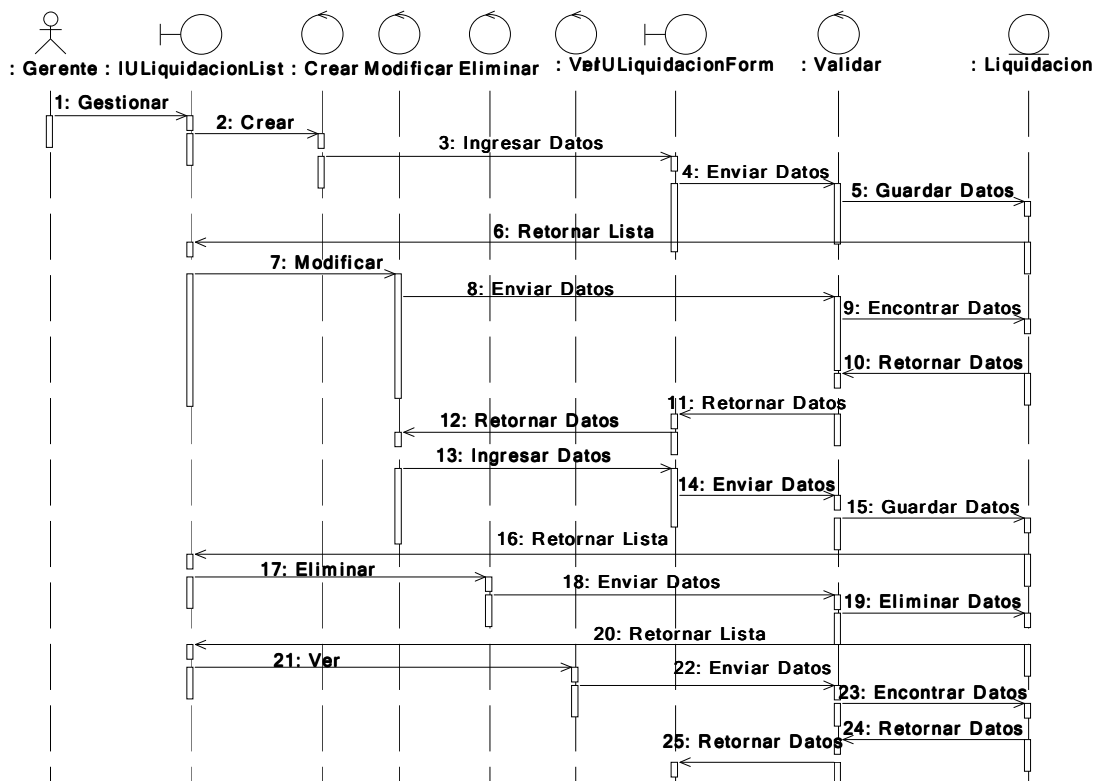
Figura 2.30 Diagrama de Secuencia del caso de uso Gestionar Despacho.



Fuente: Los Autores

2.2.2.7 Subsistema Gestionar Liquidación.

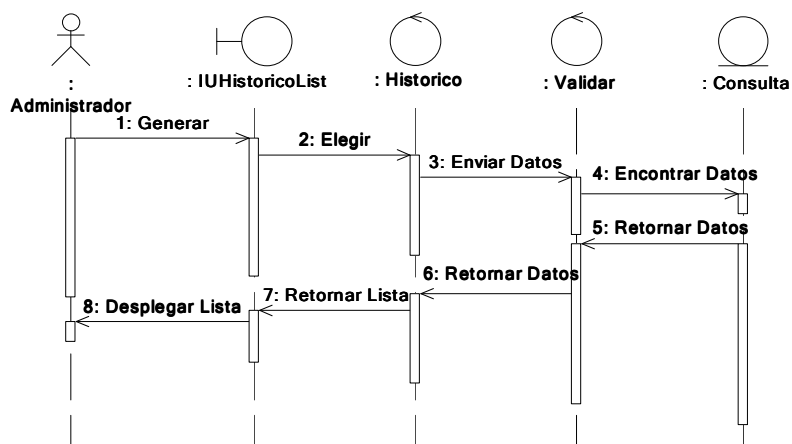
Figura 2.31 Diagrama de Secuencia del caso de uso Gestionar Liquidación.



Fuente: Los Autores

2.2.2.8 Subsistema Generar Histórico.

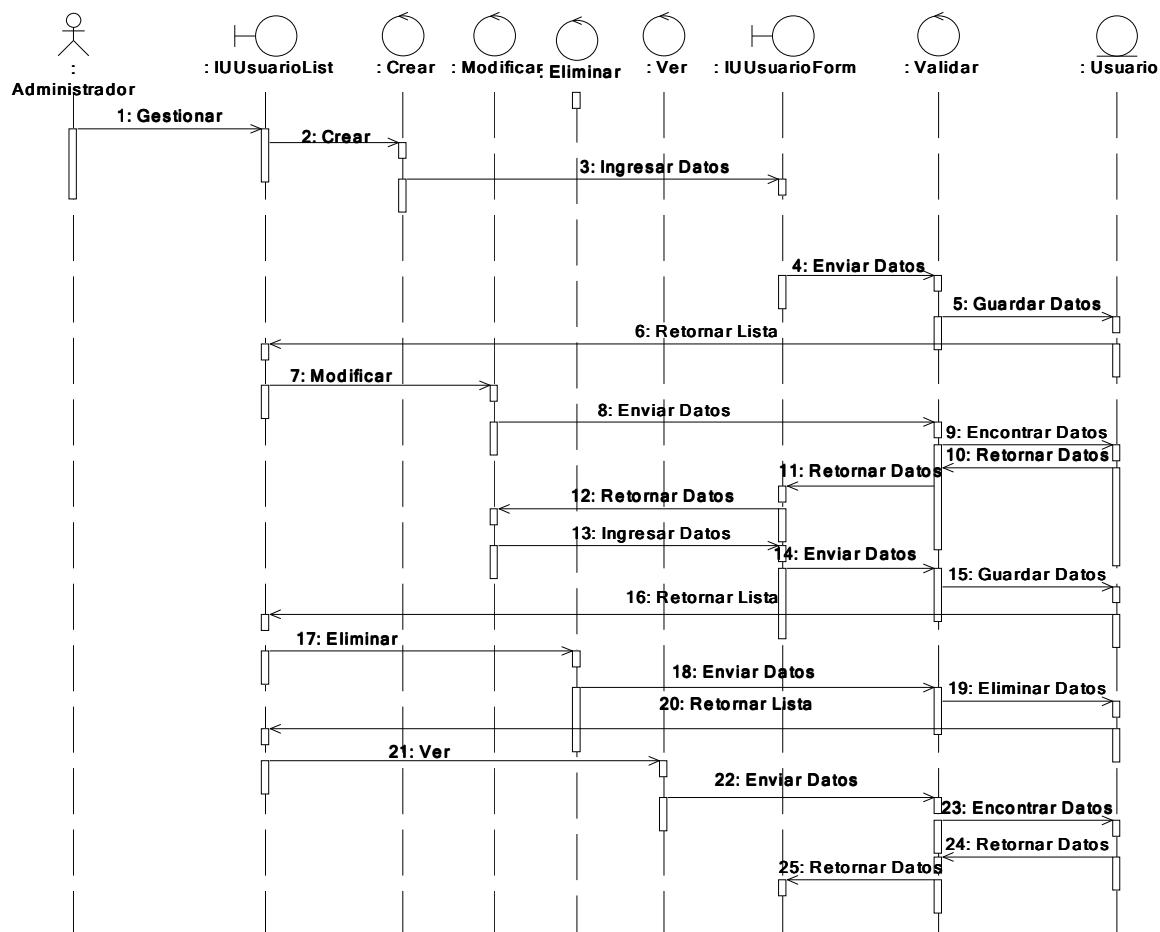
Figura 2.32 Diagrama de Secuencia del caso de uso Generar Histórico



Fuente: Los Autores

2.2.2.9 Subsistema Gestionar Usuario.

Figura 2.33 Diagrama de Secuencia del caso de uso Gestionar Usuario



Fuente: Los Autores

2.2.3 DISEÑO DE INTERFACES

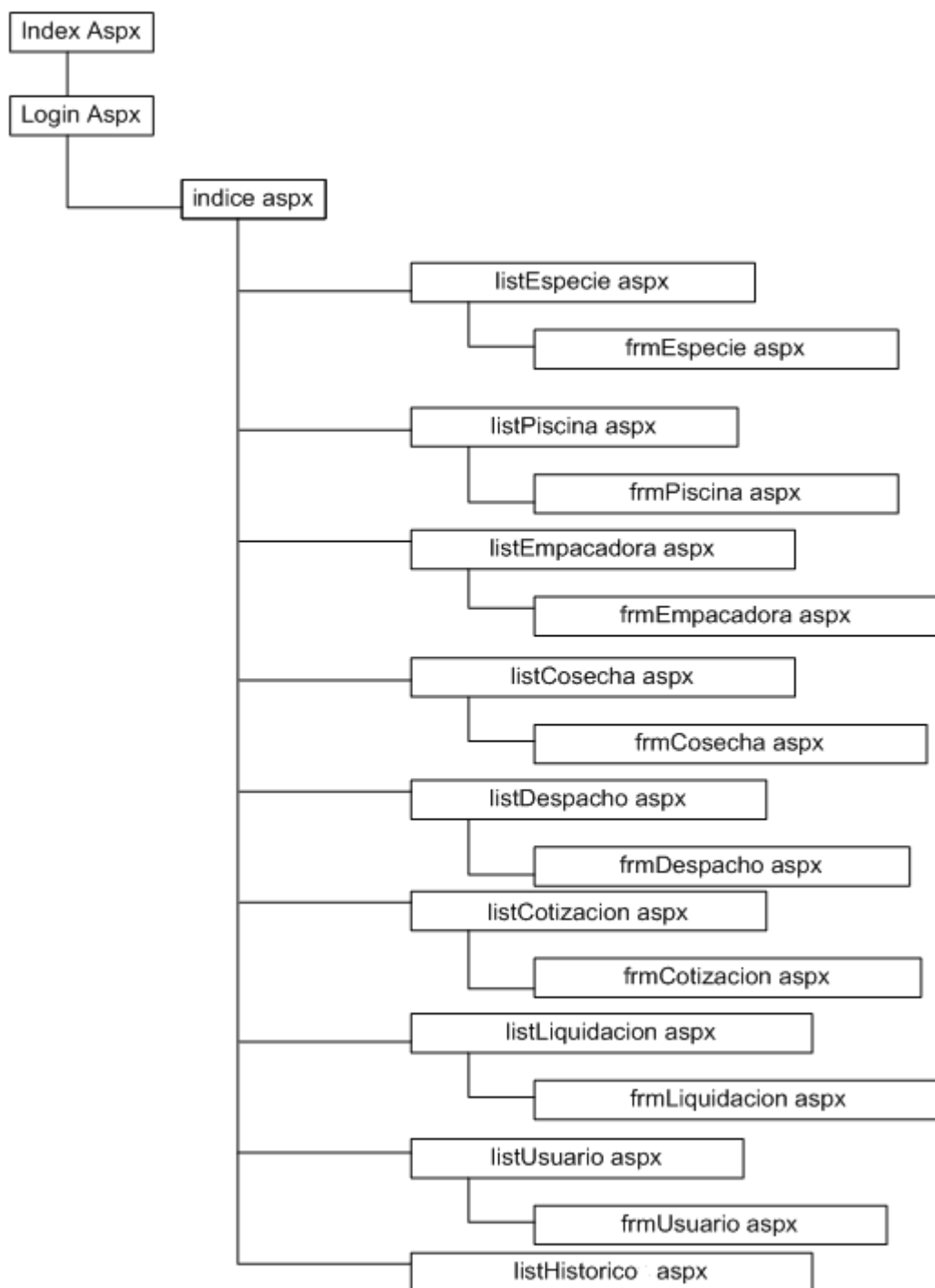
La siguiente sección describe el diseño de las interfaces para cada uno de los módulos del sistema.

Dentro de esta etapa de diseño se debe considerar dos aspectos importantes que son el mapa de navegación y el diseño de la estructura de las páginas Web del sistema.

2.2.3.1 Mapa de Navegación

La Figura 2.34 muestra el mapa de navegación de las actividades a realizar en el sistema.

Figura 2.34 Mapa de Navegación



Fuente: Los Autores

2.2.3.2 Estructura de páginas

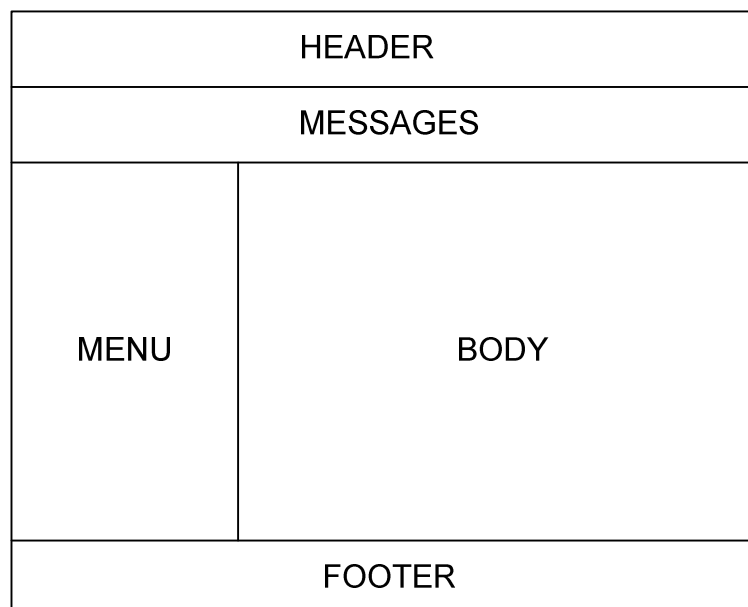
La página del sistema de administración posee la siguiente estructura.

Tabla 2. 28 Estructura de Páginas

Nombre de la Pagina	Descripción
Header	Despliega el Logo del Sistema.
Messages	Despliega toda clase de mensajes al usuario.
Menú	Despliega las distintas acciones que pueden realizarse en el sistema.
Body	El contenido de esta página cambia de acuerdo a la acción que haya sido previamente seleccionada en el menú.
Footer	Contiene imágenes del sistema

Fuente: Los Autores

Figura 2.35 Estructura de Páginas



Fuente: Los Autores

La página Body puede estar formada por dos tipos de páginas: página List y la página Frm las mismas que serán detalladas a continuación.

Las paginas List además son utilizadas por la operación Eliminar.

Tabla 2. 29 Descripción Estructura Paginas List

Tipo de Pagina	List
Descripción	<p>Área de Botones.- Esta área contiene las operaciones que pueden realizarse sobre los objetos que pueden ser: Nuevo, Modificar, Eliminar y Ver.</p> <p>Descripción.- Despliega el nombre de un campo mostrado en la lista.</p> <p>Radio Botones.- Permite seleccionar un objeto de lista desplegada.</p>
Operaciones	<p>Eliminar.- El sistema despliega un cuadro de dialogo en el que el usuario deberá seleccionar entre las opciones Aceptar o Cancelar la operación.</p> <p>En el caso de elegir la opción Aceptar el ítem será eliminado del sistema si y solo si no posee registros asociados a él (Para realizar esta operación es necesario que el usuario haya seleccionado un ítem previamente).</p>
Observaciones	<p>Este tipo de pantallas, son las pantallas principales del menú de operaciones.</p> <p>Se ejecutan cuando se presiona cualquier opción dentro la pagina Menú.</p> <p>Esta opción se ejecuta la primera vez que la pagina es llamada o después de realizarse las operaciones de inserción, modificación ver o eliminación.</p> <p>Esta opción lista todos los ítems almacenados en el sistema, correspondiente a la opción seleccionada en el menú.</p> <p>El usuario debe examinar y seleccionar entre la lista de ítems presentados, al ítem que desea modificar, ver o eliminar.</p>

Fuente: Los Autores

Figura 2.36 Estructura de la Pagina List

AREA DE BOTONES			
○	↕Descripción	↕Descripción	↕Descripción
○			
○			
○			
○			
○			
○			

Fuente: Los Autores

Las paginas Frm son llamadas para realizar operaciones como Crear, Modificar y Ver.

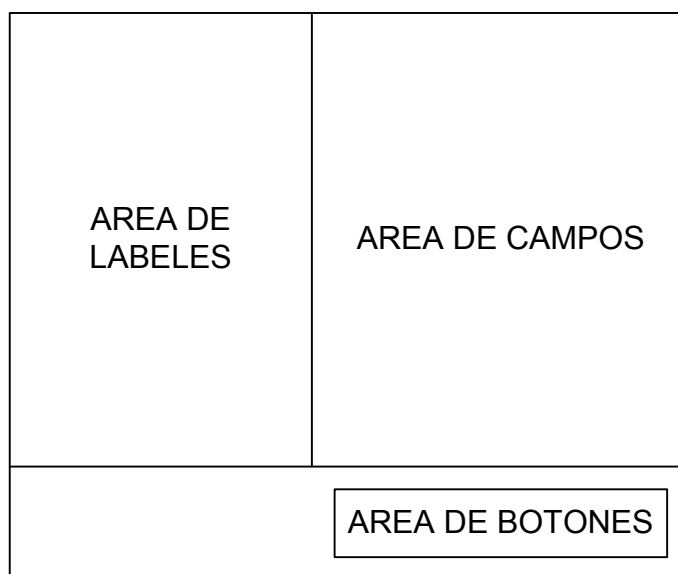
Tabla 2. 30 Descripción Estructura Paginas Frm

Tipo de Pagina	Frm
Descripción	<p>Área de Botones.- Esta área contiene botones con las operaciones que pueden realizarse sobre los objetos (Crear, Modificar y Cancelar).</p> <p>Área de Campos.- Esta área contiene los atributos de los objetos a ser llenados o modificados por el usuario.</p> <p>Área de Labels.- Esta área contiene los campos a ser llenados o notificados.</p>
Operaciones	<p>Crear.- Cuando se ejecute esta acción todos los campos de texto aparecerán vacíos, para que el usuario pueda llenar dichos campos con la información adecuada al menú seleccionado previamente.</p> <p>Modificar.- Cuando se ejecute esta acción todos los campos permitidos se presentaran en modo de edición</p>

	<p>para que el usuario pueda modificarlos (Para realizar esta operación es necesario que el usuario haya seleccionado un ítem previamente).</p> <p>Ver.- Cuando se ejecute esta acción todos los campos permitidos se representan en modo de vista para que el usuario pueda ver toda la información de ese ítem (Para realizar esta operación es necesario que el usuario haya seleccionado un ítem previamente).</p>
Observaciones	Posteriormente después de realizar cualquiera de estas operaciones se ejecuta la página List.

Fuente: Los Autores

Figura 2.37 Estructura de la Página Frm



Fuente: Los Autores

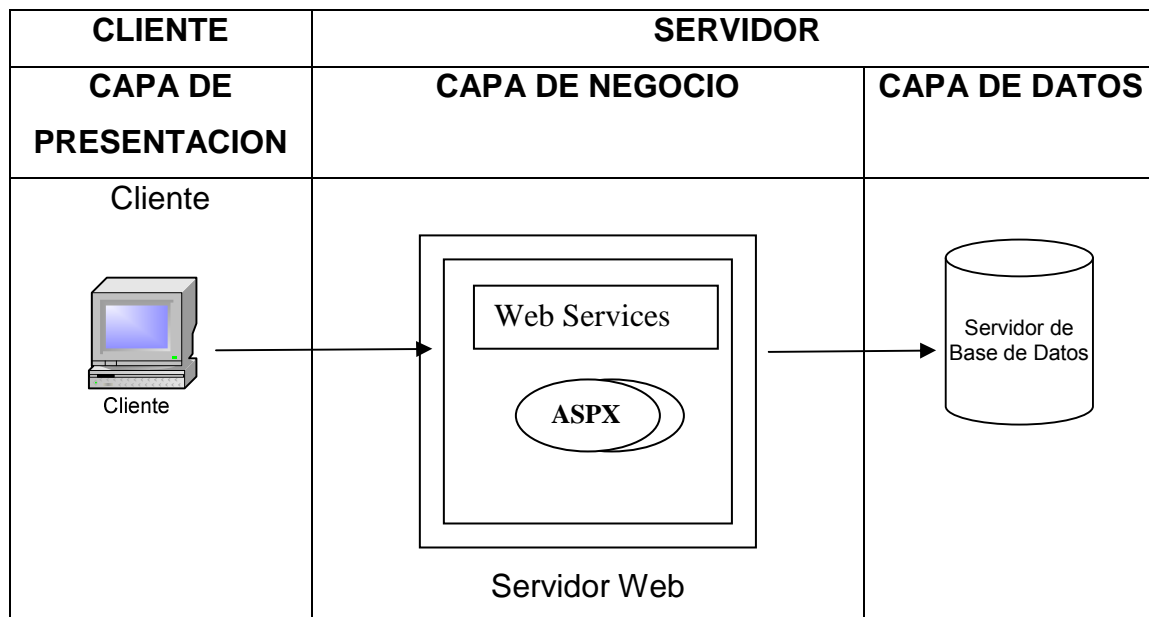
2.2.4 ARQUITECTURA DEL SISTEMA

Para la implementación del sistema se utilizara el modelo de capas, con lo que mediante la organización de los elementos de la aplicación en componentes independientes, se obtendrá una mayor efectividad al momento de administrarlos.

Este sistema está conformado por tres capas:

- Capa de Presentación.
- Capa de Negocio.
- Capa de Datos.

Figura 2.38 Diagrama de la Arquitectura del Sistema



Fuente: Los Autores

Tabla 2. 31 Descripción de la Arquitectura del Sistema

DISTRIBUCION	DESCRIPCION
Capa de Presentación	Contiene elementos que se encuentran tanto en el cliente como en el servidor. El cliente accede al servidor Web mediante la interfaz Web.
Capa de Negocio	Contiene elementos propios de la lógica del negocio de la aplicación.
Capa de Datos	Es responsable de proveer acceso a los datos de la aplicación.

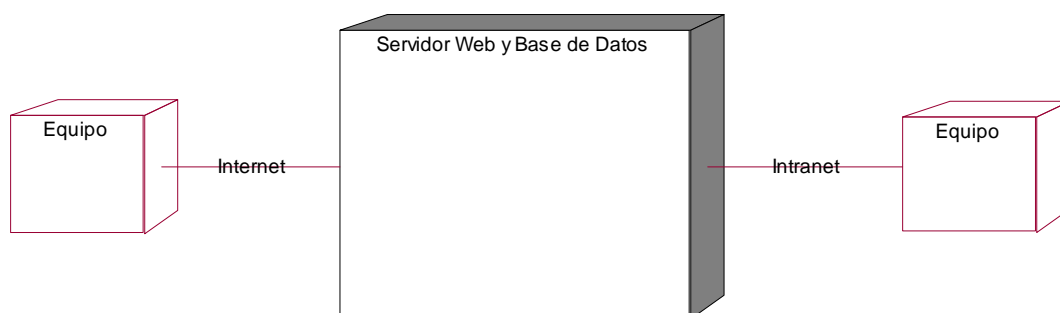
Fuente: Los Autores

2.2.5 MODELO DE DESPLIEGUE.

El modelo de despliegue muestra la disposición física de los diferentes nodos que conforman el sistema.

Los diagramas de despliegue sirven para modelar la configuración de hardware del sistema, mostrando que nodos lo componen.

Figura 2.39 Diagrama de Despliegue



Fuente: Los Autores

- 1 El primer nodo representa a un computador, el cual a través del Internet tendrá acceso al sistema mediante un browser.
- 2 El segundo nodo corresponde al servidor Web del sistema, en el cual se encuentran los componentes necesarios para el manejo de las reglas de negocio, el acceso a la base de datos, así como también la base de datos.
- 3 El tercer nodo representa a un computador, el cual a través de la intranet tendrá acceso al sistema mediante un browser

2.3 IMPLEMENTACION

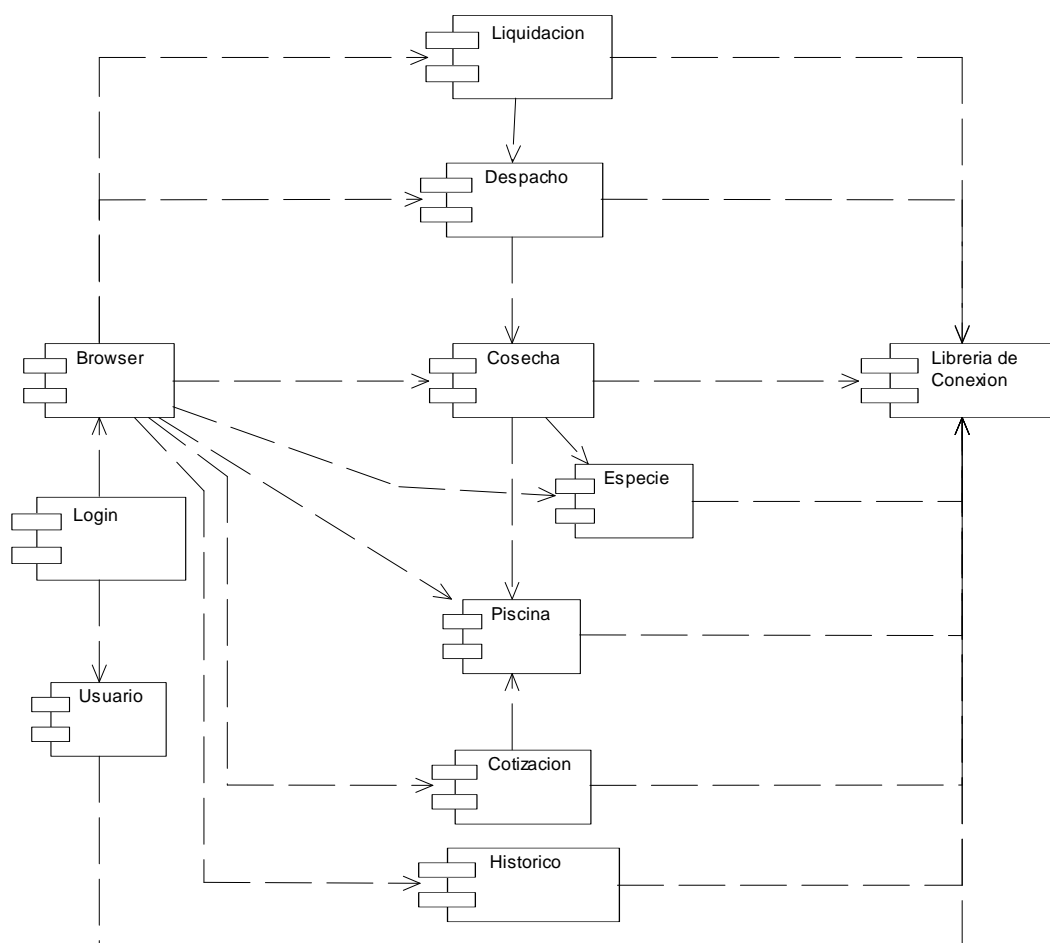
La implementación traduce la información que se obtiene en los modelos UML, a la estructura de código y de la base de datos.

2.3.1 MODELO DE IMPLEMENTACION

Los diagramas de componentes permiten describir los elementos físicos del sistema y las relaciones que existen entre ellos. Estos componentes representan todos los tipos de elementos de software que entran en la fabricación de la aplicación.

El Modelo de Implementación se identificó con los componentes principales que representan los componentes físicos del Sistema.

Figura 2.40 Diagrama de Componentes



Fuente: Los Autores

A continuación se detalla la funcionalidad de cada uno de los componentes que conforman el sistema.

- BROWSER.- Este componente sirve de vínculo entre los componentes y el usuario.
- LIBRERÍA DE CONEXIÓN.- Este componente sirve de vínculo entre los componentes y la base de datos.
- EMPACADORA.- Este componente se encarga de gestionar la información referente a la empacadora.
- COTIZACION.- Este componente se encarga de gestionar la información referente a la cotización.
- LIQUIDACION.- Este componente se encarga de gestionar la información referente a la liquidación.
- ESPECIE.- Este componente se encarga de gestionar la información referente a la especie.
- COSECHA.- Este componente se encarga de gestionar la información referente a la cosecha.
- USUARIO.- Este componente se encarga de gestionar la información referente a los usuarios con sus respectivos perfiles.
- PISCINA.- Este componente se encarga de gestionar la información referente a la piscina.
- HISTORICOS- Este componente se encarga de generar la información referente a históricos.
- LOGIN.- Este componente se encarga de gestionar el ingreso al sistema.

2.3.2 IMPLEMENTACIÓN DE LA BASE DE DATOS

Para realizar la implementación de la base de datos se debe desarrollar los diagramas lógico y físico del Modelo Entidad – Relación para lo que se tomo en cuenta el diagrama de clases final.

De esta manera es posible transformar un modelo de clases en un modelo de datos, siempre y cuando se tome en cuenta que, cada clase se puede convertir en una o más entidades de la base de datos, y que cada atributo de la clase se convierte en un atributo de la entidad.

2.3.2.1 Equivalencias de Clases con entidades

Para la transformación del modelo de clases al modelo relacional de la base de datos se utilizó un ORM²⁶, la equivalencia existente entre tablas y clases se presenta en la Tabla 2.32

Tabla 2. 32 Mapeo de Clases a Entidades de la Base de Datos

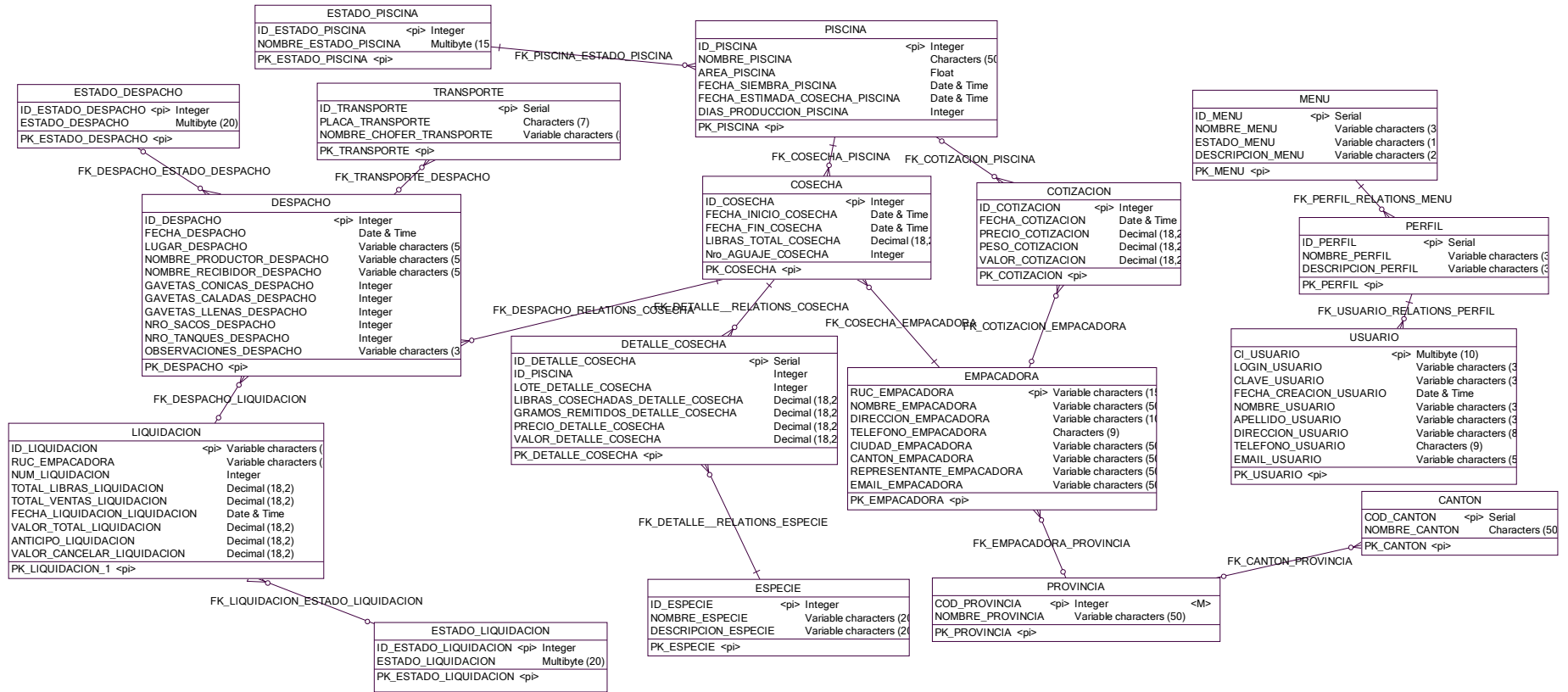
Clase	Entidad
Especie	ESPECIE
Piscina	PISCINA
Empacadora	EMPACADORA
Cosecha	COSECHA
Detalle Cosecha	DETALLE _ COSECHA
Despacho	DESPACHO
Liquidación	LIQUIDACION
Usuario	USUARIO
Perfil	PERFIL
Menú	MENU

Fuente: Los Autores

²⁶ ORM: Object Relational Mapping

2.3.2.2 Modelo Conceptual de la base de datos

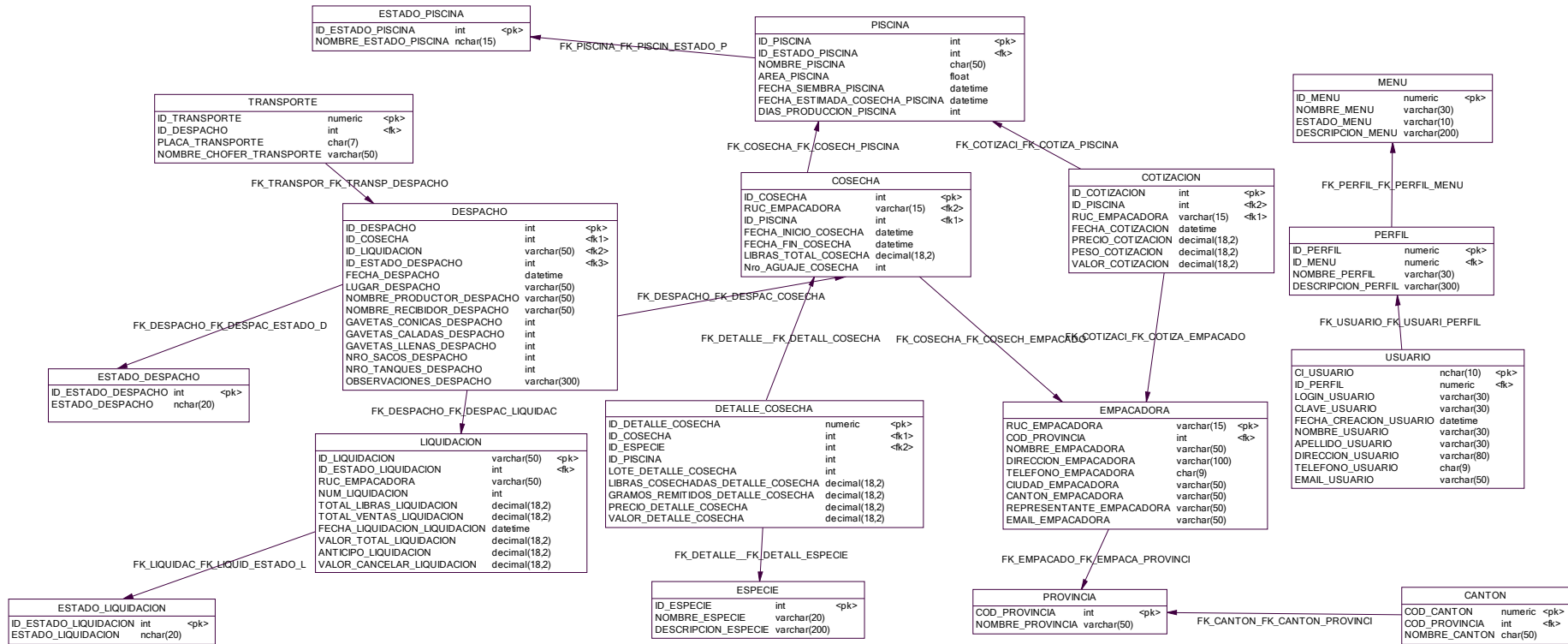
Figura 2.41 Modelo Conceptual de la base de datos SACP



Fuente: Los Autores

2.3.2.3 Modelo Físico de la base de datos

Figura 2.42 Modelo Físico de la base de datos SACP



Fuente: Los Autores

2.3.3 ESTÁNDARES DE PROGRAMACIÓN

La construcción de una aplicación con estándares de programación hace posible que el mantenimiento e implementación de la misma se facilite.

A continuación se definen estándares de programación para el modulo a implementarse, estándares de controles, además del estándar de diseño de la base de datos.

El estándar para este modulo incluye normas para el nombramiento de las paginas, nombramiento de las clases, variables y métodos empleados.

Tabla 2. 33 Descripción Estándares de Programación

ELEMENTO	DESCRIPCION
Paginas ASPX	Deberán tener el mismo nombre de la tabla, antepuestas de la palabra frm y list, de acuerdo a su finalidad. Por ejemplo. frmEmpacadora.aspx listEmpacadora.aspx
Nombres de Clases	Deberán tener el mismo nombre de la tabla, antepuesta de la palabra Clss. Por Ejemplo: Clssempacadora.cs
Métodos y Variables	Los nombres de las variables se especifican con el estándar de todas minúsculas a diferencia de los métodos, los cuales son palabras compuestas eliminando los espacios y poniendo la primera letra en mayúscula y en mayúscula la primera letra de cada palabra siguiente. Por ejemplo: Perfil

	<p>idpiscina</p> <p>Por ejemplo:(Métodos)</p> <p>InsertarUsuario()</p> <p>CargarDatos()</p>
--	--

Fuente: Los Autores

2.3.3.1 Estándares de Controles

Los nombres de los controles se especifican con el estándar de forma que las tres primeras letras en minúscula identifiquen el tipo de control y luego se escribirá un nombre representativo que podrá identificar su utilización.

Por ejemplo:

txtNombreEmpacadora: Control tipo TextBox, Contiene el nombre de la empacadora

Tabla 2. 34 Descripción Estándares de Controles

ELEMENTO	DESCRIPCION
DropDownList	<p>Cuadro Combinado</p> <p>Por ejemplo.</p> <p>ddlProvincia</p>
Button	<p>Botón de Comando</p> <p>Por ejemplo.</p> <p>btnCrearPiscina</p>
Label	<p>Etiqueta</p> <p>Por ejemplo.</p> <p>lblLiquidacion</p>
GridView	Rejilla enlazada a datos

	Por ejemplo. gvLiquidacion
RadioButton	Casilla de Verificación Por ejemplo. rbtLiquidacion
TextBox	Cuadro de Texto Por ejemplo. txtFechaLiquidacion
Image	Cuadro de Imagen Por ejemplo. Img1
HyperLink	Link de comando Por ejemplo. hplLiquidacion

Fuente: Los Autores

2.3.3.2 Estándares de diseño de la base de datos

Tabla 2. 35 Descripción Estándares de Diseño Base de Datos

ELEMENTO	DESCRIPCION
Tablas y Campos	<ul style="list-style-type: none"> Los nombres de las tablas se deben escribir en mayúscula, el nombre de los campos se escriben con mayúscula conjuntamente con el nombre de la tabla separados por el carácter “_”. <p>Por ejemplo:</p> <p style="text-align: center;">LIQUIDACION</p> <p>Por ejemplo(Campos)</p> <p style="text-align: center;">RUC_LIQUIDACION</p>

	<ul style="list-style-type: none">• Los nombres de las tablas deben especificarse en singular, y de acuerdo al punto anterior. <p>Por ejemplo:</p> <p>PISCINA</p> <ul style="list-style-type: none">• En el caso de tablas que se relacionan específicamente con otra tabla, esta relación debe quedar expresada en el nombre. <p>Por ejemplo:</p> <p>FK_DESPACHO_LIQUIDACION</p> <ul style="list-style-type: none">• Toda tabla debe poseer uno o más campos clave.• Toda relación entre tablas debe implementarse mediante constrains (claves foráneas) con integridad referencial, de acuerdo al motor de base de datos utilizado.• Los campos clave deben ubicarse al inicio de la definición de la tabla.• El nombre del campo clave debe estar compuesto por "ID" + el nombre de la tabla en singular (para claves no compuestas). Dependiendo de la naturaleza de la entidad, el nombre de la tabla a usar es el de la misma tabla, o el del relacionado, cabe señalar que si la entidad posee un campo representativo este será utilizado como clave y conservara su nombre propio. <p>Por ejemplo:</p> <p>ID_USUARIO RUC_EMPACADORA</p>
--	--

2.4 PRUEBAS

Para realizar las pruebas mostradas a continuación, se tomo en cuenta los aspectos de validación necesarios para que el sistema brinde un buen desempeño libre de errores.

El equipo para realizar las actividades de Servidor y Cliente fue:

Tabla 2. 36 Descripción Equipo de Pruebas

Parámetro	Descripción
Procesador	Core 2 Duo
Velocidad	2.0 GHz
RAM	2.048 Mb
Sistema Operativo	Windows XP SP2

Fuente: Los Autores

2.4.1 CASOS Y PROCEDIMIENTOS DE PRUEBAS

Las pruebas son un elemento esencial en el ciclo de desarrollo de software y consiste en una revisión final de los requerimientos, análisis, diseño e implementación.

Las pruebas del sistema se realizaron utilizando casos y procedimientos de prueba, para los casos de uso identificados durante la etapa de análisis de requerimientos.

A continuación se detallan los casos y procedimientos de prueba para los casos de uso del sistema.

2.4.1.1 Caso de Prueba Inicio de Sesión de Usuario (Ingreso Correcto)

Tabla 2. 37 Prueba de Inicio de Sesión de Usuario (Ingreso Correcto)

Caso de Prueba	Inicio de sesión usuario
Entrada:	Ingreso de Datos: Nombre de Usuario="Admin". Contraseña:"admin". Presionar Inicio de sesión.
Resultado Esperado:	El sistema despliega el menú correspondiente al usuario que ha ingresado.
Condiciones:	El usuario debe existir dentro del sistema, y deben coincidir el nombre de usuario y la contraseña.
Observaciones:	En caso de error, deberá mostrar la interfaz inicial, con un mensaje informativo.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario de login. • Ingresar el nombre de usuario y la contraseña (Admin) y (admin) • Oprimir el botón (Inicio de sesión)
Resultado Real:	El sistema despliega la interfaz con el menú administrativo correspondiente al usuario.

Fuente: Los Autores

2.4.1.2 Caso de Prueba Inicio de Sesión de Usuario (Ingreso Fallido)

Tabla 2. 38 Prueba de Inicio de Sesión de Usuario (Ingreso Fallido)

Caso de Prueba	Inicio de sesión usuario
Entrada:	Ingreso de Datos: Nombre de Usuario="Admin". Contraseña:"". Presionar Inicio de sesión.

Resultado Esperado:	El sistema despliega el menú correspondiente al usuario que ha ingresado.
Condiciones:	El usuario debe existir dentro del sistema, y deben coincidir el nombre de usuario y la contraseña.
Observaciones:	En caso de error, deberá mostrar la interfaz inicial, con un mensaje informativo.
Procedimiento:	
<ul style="list-style-type: none"> • Ingresar al formulario de login. • Ingresar el nombre de usuario y la contraseña (Admin) y (admin) • Oprimir el botón (Inicio de sesión) 	
Resultado Real:	
El sistema despliega la interfaz de ingreso de datos, con un mensaje con letras de en rojo que dice “El intento de conexión no fue correcto. Inténtelo de nuevo”.	

Fuente: Los Autores

2.4.1.3 Caso de Prueba Crear Especie

Tabla 2. 39 Prueba Crear Especie

Caso de Prueba	Crear Especie
Entrada:	Ingreso de Datos: Nombre Especie:” NORMAL”. Descripción:” Producto de buena calidad”. Presionar Crear.
Resultado Esperado:	El sistema despliega la lista de todas las especies entre los cuales se encuentra la nueva especie creada.
Condiciones:	El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Especie.
Observaciones:	No existen observaciones.
Procedimiento:	
<ul style="list-style-type: none"> • Ingresar al formulario Piscina. • Elegir Nuevo. 	

<ul style="list-style-type: none"> • Ingresar los datos para la nueva Piscina: Nombre Especie:" NORMAL". Descripción:" Producto de buena calidad". • Oprimir el botón (Crear)
<p>Resultado Real:</p> <p>El sistema despliega la interfaz con la lista de las especies creadas, la cual contiene la nueva especie que se acabo de ingresar.</p>

Fuente: Los Autores

2.4.1.4 Caso de Prueba Modificar Especie

Tabla 2. 40 Prueba Modificar Especie

Caso de Prueba	Modificar Especie
Entrada:	<p>Cambio de Datos:</p> <p>Nombre Especie:" NORMAL". Descripción:" Producto de buena calidad".</p> <p>Presionar Modificar.</p>
Resultado Esperado:	El sistema despliega la lista de todas las especies entre los cuales se encuentra la nueva especie creada.
Condiciones:	El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Especie.
Observaciones:	No existen observaciones.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Especie. • Seleccionar una especie existente. • Elegir modificar. • Cambiar los datos presentados por los nuevos en Piscina: Nombre Especie:" NORMAL". Descripción:" Producto de alta calidad". • Oprimir el botón (Modificar)

Resultado Real:

El sistema despliega la interfaz con la lista de especies creadas, la cual contiene la nueva especie que se acabo de modificar.

Fuente: Los Autores

2.4.1.5 Caso de Prueba Eliminar Especie (Datos Correctos)

Tabla 2. 41 Prueba Eliminar Especie

Caso de Prueba	Eliminar Especie
Entrada:	Presionar el Botón Eliminar para la Especie="NORMAL"
Resultado Esperado:	El sistema despliega la lista de todas las especies sin la especie que se ha eliminado.
Condiciones:	<ul style="list-style-type: none"> • El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Especie. • La especie debe existir en el sistema y no debe estar asignada a ninguna despacho
Observaciones:	Si la especie puede ser eliminada no se desplegara ningún mensaje de error en la pantalla del sistema.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Especie. • Seleccionar una especie existente. • Presionar el Botón (Eliminar) • El listado se refresca.
Resultado Real:	La especie es borrada del sistema por lo que despliega la interfaz con la lista de especies sin la especie borrada anteriormente.

Fuente: Los Autores

2.4.1.6 Caso de Prueba Crear Piscina

Tabla 2. 42 Prueba Crear Piscina

Caso de Prueba	Crear Piscina
Entrada:	<p>Ingreso de Datos:</p> <p>Nombre:"PiscinaA".</p> <p>Fecha Siembra:" 17/07/2008".</p> <p>Fecha Cosecha:" 24/10/2008".</p> <p>Días de Producción:"99".</p> <p>Superficie:" 9".</p> <p>Estado Actual:"Productiva".</p> <p>Presionar Crear.</p>
Resultado Esperado:	El sistema despliega la lista de todas las piscinas entre los cuales se encuentra la nueva piscina creada.
Condiciones:	El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Piscina.
Observaciones:	La piscina se la puede crear con el estado Productiva o No Productiva si se crea con el estado Productiva se puede realizar una cosecha sobre dicha piscina.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Piscina. • Elegir Nuevo. • Ingresar los datos para la nueva Piscina: Nombre:"PiscinaA". Fecha Siembra:" 17/07/2008". Fecha Cosecha:" 24/10/2008". Días de Producción:"99". Superficie:" 9". Estado Actual:"Productiva". • Oprimir el botón (Crear)
Resultado Real:	El sistema despliega la interfaz con la lista de las piscinas creadas, la cual contiene la nueva piscina que se acabo de ingresar.

Fuente: Los Autores

2.4.1.7 Caso de Prueba Modificar Piscina

Tabla 2. 43 Prueba Modificar Piscina

Caso de Prueba	Modificar Piscina
Entrada:	<p>Cambio de Datos:</p> <p>Nombre:"PiscinaA".</p> <p>Fecha Siembra:" 17/07/2008".</p> <p>Fecha Cosecha:" 24/10/2008".</p> <p>Días de Producción:"99".</p> <p>Superficie:" 9".</p> <p>Estado Actual:"No Productiva".</p> <p>Presionar Modificar.</p>
Resultado Esperado:	El sistema despliega la lista de todas las piscinas entre los cuales se encuentra la nueva piscina creada.
Condiciones:	El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Piscina.
Observaciones:	La piscina se la puede crear con el estado Productiva o No Productiva si se crea con el estado Productiva se puede realizar una cosecha sobre dicha piscina.
Procedimiento:	
<ul style="list-style-type: none"> • Ingresar al formulario Piscina. • Seleccionar una piscina existente. • Elegir Modificar. • Cambiar los datos presentados por los nuevos en Piscina: Nombre:"PiscinaA". Fecha Siembra:" 17/07/2008". Fecha Cosecha:" 24/10/2008". Días de Producción:"99". Superficie:" 9". Estado Actual:"No Productiva". • Oprimir el botón (Modificar) 	

Resultado Real:

El sistema despliega la interfaz con la lista de piscinas creadas, la cual contiene la nueva piscina que se acabo de modificar.

Fuente: Los Autores

2.4.1.8 Caso de Prueba Eliminar Piscina (Datos Correctos)

Tabla 2. 44 Prueba Eliminar Piscina (Datos Correctos)

Caso de Prueba	Eliminar Piscina
Entrada:	Presionar el Botón Eliminar para la Piscina="PiscinaA"
Resultado Esperado:	El sistema despliega la lista de todas las piscinas sin la piscina que se ha eliminado.
Condiciones:	<ul style="list-style-type: none"> • El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Piscinas. • La piscina debe existir en el sistema y no debe estar asignada a ninguna cosecha
Observaciones:	Si la piscina puede ser eliminada no se desplegara ningún mensaje de error en la pantalla del sistema.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Piscina. • Seleccionar una piscina existente. • Presionar el Botón (Eliminar) • El listado se refresca.
Resultado Real:	La piscina es borrada del sistema por lo que despliega la interfaz con la lista de piscinas sin la piscina borrada anteriormente.

Fuente: Los Autores

2.4.1.9 Caso de Prueba Crear Empacadora

Tabla 2. 45 Prueba Crear Empacadora

Caso de Prueba	Crear Empacadora
Entrada:	Ingreso de Datos: RUC:"0790094557001"

	<p>Nombre:"REVINCOR S.A.". Dirección:" Puerto Hualtaco". Provincia:"El Oro". Cantón:"Huaquillas". Ciudad:"Huaquillas". Teléfono:"072944637", Email:"revincor@gmail.com", Nombre:"Edwin Alcívar".</p> <p>Presionar Crear.</p>
Resultado Esperado:	El sistema despliega la lista de todas las empacadoras entre los cuales se encuentra la nueva empacadora creada.
Condiciones:	El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Empacadora.
Observaciones:	No existen observaciones.
Procedimiento:	
<ul style="list-style-type: none"> • Ingresar al formulario Empacadora. • Elegir Nuevo. • Ingresar los datos para la nueva Empacadora: RUC:"0790094557001" Nombre:"REVINCOR S.A.". Dirección:" Puerto Hualtaco". Provincia:"El Oro". Cantón:"Huaquillas". Ciudad:"Huaquillas". Teléfono:"072944637", Email:"revincor@gmail.com", Nombre:"Edwin Alcívar". • Oprimir el botón (Crear) 	
Resultado Real:	
El sistema despliega la interfaz con la lista de las empacadoras creadas, la cual contiene la nueva empacadora que se acabo de ingresar.	

2.4.1.10 Caso de Prueba Modificar Empacadora

Tabla 2. 46 Prueba Modificar Empacadora

Caso de Prueba	Modificar Empacadora
Entrada:	<p>Cambio de Datos:</p> <p>RUC:"0790094557001"</p> <p>Nombre:"REVINCOR S.A."</p> <p>Dirección:" Puerto Hualtaco".</p> <p>Provincia:"El Oro".</p> <p>Cantón:"Huaquillas".</p> <p>Ciudad:"Huaquillas".</p> <p>Teléfono:"072944637",</p> <p>Email:"revincor@gmail.com",</p> <p>Nombre:"Edwin Alcívar".</p> <p>Presionar Modificar.</p>
Resultado Esperado:	El sistema despliega la lista de todas las empacadoras entre los cuales se encuentra la nueva empacadora creada.
Condiciones:	El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Empacadora.
Observaciones:	No existen observaciones.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Empacadora. • Seleccionar una empacadora existente. • Elegir Modificar. • Cambiar los datos presentados por los nuevos en Empacadora: RUC:"0790094557001" Nombre:"REVINCOR S.A." Dirección:" Puerto Hualtaco". Provincia:"El Oro". Cantón:"Huaquillas". Ciudad:"Huaquillas". Teléfono:"072944637",

<p>Email:"revincor@gmail.com", Nombre:"Luís Solano".</p> <ul style="list-style-type: none"> • Oprimir el botón (Modificar)
<p>Resultado Real:</p> <p>El sistema despliega la interfaz con la lista de empacadoras creadas, la cual contiene la nueva empacadora que se acabo de modificar.</p>

Fuente: Los Autores

2.4.1.11 Caso de Prueba Eliminar Empacadora (Datos Correctos)

Tabla 2. 47 Prueba Eliminar Empacadora

Caso de Prueba	Eliminar Empacadora
Entrada:	Presionar el Botón Eliminar para la Empacadora="REVINCOR S.A"
Resultado Esperado:	El sistema despliega la lista de todas las empacadoras sin la empacadora que se ha eliminado.
Condiciones:	<ul style="list-style-type: none"> • El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Empacadora. • La empacadora debe existir en el sistema y no debe estar asignada a ninguna cosecha tampoco a cotización alguna.
Observaciones:	Si la empacadora puede ser eliminada no se desplegara ningún mensaje de error en la pantalla del sistema.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Empacadora. • Seleccionar una empacadora existente. • Presionar el Botón (Eliminar) • El listado se refresca.
Resultado Real:	La empacadora es borrada del sistema por lo que despliega la interfaz con la lista de empacadoras sin la empacadora borrada anteriormente.

Fuente: Los Autores

2.4.1.12 Caso de Prueba Crear Cotización

Tabla 2. 48 Prueba Crear Cotización

Caso de Prueba	Crear Cotización
Entrada:	<p>Ingreso de Datos:</p> <p>Fecha:" 17/07/2008".</p> <p>Piscina:" PiscinaC".</p> <p>Empacadora:"REVINCOR S.A.".</p> <p>Peso:"3420".</p> <p>Precio:"1,16".</p> <p>Valor Cotizado:" 3967,20".</p> <p>Presionar Crear.</p>
Resultado Esperado:	El sistema despliega la lista de todas las cotizaciones entre los cuales se encuentra la nueva cotización creada.
Condiciones:	<p>El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Cotización.</p> <p>La piscina debe existir en el sistema.</p> <p>La empacadora debe existir en el sistema</p>
Observaciones:	No existen observaciones.
Procedimiento:	
<ul style="list-style-type: none"> • Ingresar al formulario Cotización. • Elegir Nuevo. • Ingresar los datos para la nueva Cotización: <ul style="list-style-type: none"> Fecha:" 17/07/2008". Piscina:" PiscinaC". Empacadora:"REVINCOR S.A.". Peso:"3420". Precio:"1,16". Valor Cotizado:" 3967,20". • Oprimir el botón (Cotizar) • Oprimir el botón (Crear) 	

Resultado Real:

El sistema despliega la interfaz con la lista de las cotizaciones creadas, la cual contiene la nueva cotización que se acabo de ingresar.

Fuente: Los Autores

2.4.1.13 Caso de Prueba Modificar Cotización

Tabla 2. 49 Prueba Modificar Cotización

Caso de Prueba	Modificar Cotización
Entrada:	<p>Cambio de Datos:</p> <p>Fecha:" 17/07/2008".</p> <p>Piscina:" PiscinaC".</p> <p>Empacadora:"REVINCOR S.A.".</p> <p>Peso:"3420".</p> <p>Precio:"1,16".</p> <p>Valor Cotizado:" 3967,20".</p> <p>Presionar Modificar.</p>
Resultado Esperado:	El sistema despliega la lista de todas las cotizaciones entre los cuales se encuentra la nueva cotización creada.
Condiciones:	<p>El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Cotización.</p> <p>La piscina debe existir en el sistema.</p> <p>La empacadora debe existir en el sistema.</p>
Observaciones:	No existen observaciones.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Cotización. • Seleccionar una cotización existente. • Elegir Modificar. • Cambiar los datos presentados por los nuevos en Cotización: Fecha:" 17/07/2008". Piscina:" PiscinaC". Empacadora:"REVINCOR S.A.". Peso:"3420".

<p>Precio:"1,260".</p> <p>Valor Cotizado:" 4309,20".</p> <ul style="list-style-type: none"> • Oprimir el botón (Modificar)
<p>Resultado Real:</p> <p>El sistema despliega la interfaz con la lista de cotizaciones creadas, la cual contiene la nueva cotización que se acabo de modificar.</p>

Fuente: Los Autores

2.4.1.14 Caso de Prueba Eliminar Cotización (Datos Correctos)}

Tabla 2. 50 Prueba Eliminar Cotización (Datos Correctos)

Caso de Prueba	Eliminar Cotización
Entrada:	Presionar el Botón Eliminar para la Cotización =" REVINCOR S.A."
Resultado Esperado:	El sistema despliega la lista de todas las cotizaciones sin la cotización que se ha eliminado.
Condiciones:	El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Cotización. La cotización debe existir en el sistema.
Observaciones:	Si la cotización puede ser eliminada no se desplegara ningún mensaje de error en la pantalla del sistema.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Cotización. • Seleccionar una cotización existente. • Presionar el Botón (Eliminar) • El listado se refresca.
Resultado Real:	La cotización es borrada del sistema por lo que despliega la interfaz con la lista de cotizaciones sin la cotización borrada anteriormente.

Fuente: Los Autores

2.4.1.15 Caso de Prueba Crear Cosecha

Tabla 2. 51 Prueba Crear Cosecha

Caso de Prueba	Crear Cosecha
Entrada:	<p>Ingreso de Datos:</p> <p>Fecha Inicio:"17/07/2008".</p> <p>Fecha Fin:"18/07/2008".</p> <p>Empacadora:"CAPASUR S.A."</p> <p>Nro Aguaje:"23".</p> <p>Piscina:"PISCINAB".</p> <p>Especie:"NORMAL".</p> <p>Valor: "1.20"</p> <p>Libras Cosechada:"1200".</p> <p>Especie:"POMADA".</p> <p>Valor: "0.60"</p> <p>Libras Cosechada:"300".</p> <p>Libras Total:"1500".</p> <p>Gramos Remitidos:"9.5"</p> <p>Presionar Crear.</p>
Resultado Esperado:	El sistema nos despliega el formulario de Ingreso de Despacho, indicando la respectiva cosecha y empacadora correspondiente.
Condiciones:	<p>El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de cosecha.</p> <p>La piscina debe existir en el sistema y debe estar en estado productiva.</p>
Observaciones:	La cosecha se la debe crear con los respectivos detalles de cosecha.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Cosecha. • Elegir Nuevo. • Ingresar los datos para la nueva Cosecha: Fecha Inicio:"17/07/2008".

<p>Fecha Fin:"18/07/2008".</p> <p>Empacadora:"CAPASUR S.A."</p> <p>Nro Aguaje:"23".</p> <p>Piscina:"PISCINAB".</p> <p>Especie:"NORMAL".</p> <p>Valor: "1.20"</p> <p>Libras Cosechada:"1200".</p> <p>Especie:"POMADA".</p> <p>Valor: "0.60"</p> <p>Libras Cosechada:"300".</p> <p>Libras Total:"1500".</p> <p>Gramos Remitidos:"9.5"</p> <ul style="list-style-type: none"> • Oprimir el botón (Crear)
<p>Resultado Real:</p> <p>El sistema despliega el formulario de ingreso de despacho, indicando la respectiva cosecha y empacadora correspondiente.</p>

Fuente: Los Autores

2.4.1.16 Caso de Prueba Modificar Cosecha

Tabla 2. 52 Prueba Modificar Cosecha

Caso de Prueba	Modificar Cosecha
Entrada:	<p>Cambio de Datos:</p> <p>Fecha Inicio:"17/07/2008".</p> <p>Fecha Fin:"18/07/2008".</p> <p>Empacadora:"CAPASUR S.A."</p> <p>Nro Aguaje:"22".</p> <p>Piscina:"PISCINAB".</p> <p>Especie:"NORMAL".</p> <p>Valor: "1.20"</p> <p>Libras Cosechada:"1200".</p>

	<p>Especie:"POMADA". Valor: "0.60" Libras Cosechada:"300". Libras Total:"1500". Gramos Remitidos:"9.5"</p> <p>Presionar Modificar.</p>
Resultado Esperado:	El sistema despliega la lista de todas las cosecha con los cambios realizados en la cosecha modificada.
Condiciones:	El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Cosecha.
Observaciones:	La cosecha se la puede crear con el estado Productiva o No Productiva si se crea con el estado Productiva se puede realizar una cosecha sobre dicha piscina.
<p>Procedimiento:</p> <ul style="list-style-type: none"> • Ingresar al formulario Cosecha. • Seleccionar una cosecha existente. • Elegir modificar. • Cambiar los datos presentados por los nuevos en Cosecha: Fecha Inicio:"17/07/2008". Fecha Fin:"18/07/2008". Empacadora:"REVINCOR S.A." Nro Aguaje:"23". Piscina:"PISCINAB". Especie:"NORMAL". Valor: "1.20" Libras Cosechada:"1200". Especie:"POMADA". Valor: "0.60" Libras Cosechada:"300". Libras Total:"1500". Gramos Remitidos:"9.5" • Oprimir el botón (Modificar) 	

Resultado Real:

El sistema despliega la interfaz con la lista de cosechas creadas, la cual contiene la nueva cosecha que se acabo de modificar.

Fuente: Los Autores

2.4.1.17 Caso de Prueba Eliminar Cosecha (Datos Correctos)

Tabla 2. 53 Prueba Eliminar Cosecha (Datos Correctos)

Caso de Prueba	Eliminar Cosecha
Entrada:	Presionar el Botón Cosecha para la Cosecha="12"
Resultado Esperado:	El sistema despliega la lista de todas las cosechas sin la cosecha que se ha eliminado.
Condiciones:	El usuario debe existir dentro del sistema, y debe tener los privilegios sobre el manejo de Cosechas. La cosecha debe existir en el sistema y no debe estar ligada a ningún despacho.
Observaciones:	Si la cosecha puede ser eliminada no se desplegara ningún mensaje de error en la pantalla del sistema.
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario Cosecha. • Seleccionar una cosecha existente. • Presionar el Botón Eliminar • El listado se refresca.
Resultado Real:	La cosecha es borrada del sistema por lo que despliega la interfaz con la lista de cosechas sin la cosecha borrada anteriormente.

Fuente: Los Autores

2.4.2 EVALUACION DE RESULTADOS

2.4.2.1 Herramientas

Las herramientas permiten el adecuado manejo de los recursos disponibles para el desarrollo del sistema permitiendo así la construcción de un sistema confiable, mejorando los parámetros de tiempo y costos.

La utilización de una herramienta como en el caso de Microsoft Visual Studio .Net, permite un desarrollo rápido, mediante la utilización de asistentes que definen las plantillas principales a utilizarse durante el desarrollo de software, manteniendo así un estándar estructural del sistema.

2.4.2.2 Rendimiento

El rendimiento, evalúa los parámetros de eficacia y eficiencia del sistema en una situación determinada, la cual permite medir la fiabilidad desde el punto de vista de operadores y administradores del sistema.

Debido a que el manejo de las cosechas es la parte del sistema más concurrente y transaccional, se midió el rendimiento que involucra el manejo de los usuarios Gerente y Empleado.

2.4.2.3 Utilidad

La utilidad mide el nivel de aceptación que tiene el sistema con los usuarios. Los resultados del sistema mostraron la resolución de los problemas planteados en el diseño de este proyecto de titulación.

2.4.2.4 Manejo

El manejo es evaluado en cuanto a la facilidad de uso por parte del usuario, así como la simplicidad en la capacitación al usuario.

El aprendizaje de uso del sistema requirió de una breve inducción del proceso sobre todo del área de gerencia, debido al orden en que se deben ejecutar los procesos para la carga de parámetros de inicialización.

Los usuario que probaron el sistema, se acostumbraron de una manera rápida, teniendo un tiempo de capacitación de alrededor de 3 horas. Tiempo en el cual comprendieron el funcionamiento del sistema y se pudo realizar las pruebas satisfactoriamente.

2.4.2.5 Consistencia

Se evalúa la consistencia en función de la fiabilidad de los datos, para que el sistema pueda funcionar sin ningún problema evitando errores de integridad de datos.

El sistema muestra en los distintos casos de pruebas de validaciones de consistencia en los parámetros del sistema y en los datos, al momento de ingresar los mismos, evitando así futuras validaciones complicadas y pérdidas de tiempo injustificadas. Este proceso fue muy acertado, debido a que la depuración de errores se la realizo en muy corto plazo.

2.4.2.6 Controles Internos

Los controles internos ayudan a mantener el rendimiento y la consistencia de los datos; los mismos serán evaluados por los casos de prueba anteriormente mostrados.

Las pruebas obtenidas mediante la elaboración de los casos de prueba se realizaron con éxito. Para corroborar los resultados se accedió con cada uno de los perfiles, para verificar así la fiabilidad del perfil de un usuario asignado.

El establecimiento de perfiles permitió independizar los módulos para así lograr un adecuado control de seguridad que permita mantener los datos adecuadamente administrados.

Los resultados presentaron información lógica luego de realizar ingresos sobre las diferentes interfaces del sistema, debido a los controles establecidos para impedir el ingreso de datos incorrectos.

Los procesos de control de existencia de datos ayudaron al usuario a realizar un manejo mas intuitivo de las interfaces, que a su vez acelero el trabajo evitando mensajes innecesarios y retrasos por errores.

CAPÍTULO 3 CONCLUSIONES Y RECOMENDACIONES

3.1 CONCLUSIONES.

- En el presente proyecto se utilizó la metodología RUP, la cual permitió obtener los resultados esperados en un principio, las características como son la abstracción de requerimientos del sistema y la definición de una arquitectura estable y sólida, permitieron establecer de forma clara y concisa los requerimientos en el desarrollo del proyecto.
- La Camaronera “Pampas de Cayanca” con la automatización de los procesos de Cosecha y Post-Cosecha, obtiene un mejor manejo de la información, además de la generación de datos históricos, en base a los cuales se puede mejorar las estrategias que permitan aumentar la producción, asimismo con mayor seguridad y por ende agilizando el proceso de gestión lo que le permiten mejorar su productividad
- Una de las principales fuentes de ingresos económicos en el país son las Exportaciones de Camarón , por tal motivo existen camaroneras que buscan en los sistemas informáticos la automatización de sus procesos, permitiendo mejorarlos; proporcionando un buen tratamiento de la información y logrando optimizar el tiempo, evitar pérdidas económicas, logrando una administración más rápida, eficiente y efectiva, por lo que podemos concluir que es un nicho que se puede explotar ya que pocas son la camaroneras que poseen procesos automatizados.
- La documentación generada en el proceso de desarrollo del presente proyecto nos facilitará el mantenimiento adecuado del producto, ya que la misma es realizada utilizando estándares que ayudan a su utilización y son de suma importancia si se pretende realizar cambios o mejorar el producto.

- La utilización de Microsoft Visual Studio Net 2005 para la construcción de la aplicación, facilitó en gran medida su programación; ya que posee características como son: la de poder arrastrar y colocar cualquier control visual o componente, un editor de códigos enriquecido, un entorno de depuración de primera clase y una gran productividad adicional que hicieron más fácil y rápido el desarrollo de la aplicación Web.
- El haber realizado una aplicación Web nos permite que el presente proyecto pueda tener un mantenimiento de forma sencilla y sin riesgos de incompatibilidades debido a que existe solo una versión que se encuentra en el servidor lo que implica que no hay que distribuirla entre los demás computadores y solo se necesita de un browser para acceder a dicha aplicación.
- La utilización de frameworks alternativos como Ajax, consiguió un rendimiento significativo en la red lo cual brinda una interfaz más ágil y funcional para el usuario, demostrando que los nuevos frameworks no tienen un impacto negativo sobre un sistema.
- Uno de los detalles principales a tomar en cuenta en la implementación de un sistema es que este sea entendible y de fácil manejo para el usuario final, esto se logra con una interfase amigable basada en un criterio orientado a los conocimientos del usuario.

3.2 RECOMENDACIONES.

- Durante todo el proceso de desarrollo se debe mantener una buena comunicación con el cliente, de tal manera que se pueda tomar en cuenta su opinión y sugerencias las mismas que fueron plasmadas en el transcurso de todo el proyecto.
- Se recomienda escoger herramientas acorde al tipo de aplicación que se van a utilizar en el desarrollo de un sistema, para esto se debe tomar en cuenta criterios de selección como: el ambiente de programación, el nivel de conocimiento y la facilidad de aprendizaje así como el soporte que se tenga de la misma.
- Para el uso de sistemas de tamaño medio se recomienda realizar una verificación de los flujos de trabajo a ser utilizados en el proceso de desarrollo del sistema ya que RUP permite manejar la configuración de estos dependiendo del contexto en el que se desarrollen las actividades.
- Si se utiliza RUP para proyectos de corto alcance, se debe recordar que RUP es configurable y por lo tanto, al inicio de cada proyecto, se puede delimitar el número de diagramas y artefactos que se van a implementar a lo largo del proceso de desarrollo, dejando solo los básicos y útiles de acuerdo a los requerimientos del sistema, número de integrantes de equipo de desarrollo, entre otros.
- Siempre al momento de desarrollar un sistema, se debe tener en cuenta las habilidades del equipo de desarrollo y buscar las soluciones que requieran menos tiempo; pero siempre evaluando el impacto que dichas soluciones tendrán en el futuro en el sistema desarrollado.
- Para la realización de un sistema informático es importante establecer de una forma correcta los requerimientos establecidos por el cliente, para que los resultados lo satisfagan de forma exitosa y total.

BIBLIOGRAFÍA.

LIBROS

- JACOBSON, Iván; BOOCH, Grady; RUMBAUGH James. El Proceso Unificado de Desarrollo de Software. Primera edición. Addison Wesley - Person Educación S.A. 2000.
- PRESSMAN, S. Roger. Ingeniería del Software un enfoque practico. Sexta edición. McGraw-Hill Interamericana editores S.A. 2006.
- RUMBAUGH James; JACOBSON Iván; BOOCH Grady. El lenguaje unificado de modelado. Manual de referencia. Madrid. Addison Wesley Longman Inc.- Person Educación S.A.2000.

DIRECCIONES ELECTRÓNICAS

- WIKIMEDIA FOUNDATION, INC. Descripción de los diagramas que conforman el lenguaje UML. http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso.htm, 27 de Julio del 2007.
- WIKILEARNING. Metodologías de desarrollo de software. http://www.wikilearning.com/metodologias_de_desarrollo_de_software-wkc-3617.htm. 2004.
- Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. Descripción del Proceso Unificado de Desarrollo. <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducción%20a%20RUP.doc>.

- FIGUEROA, Pablo. Etapas y actividades en el desarrollo OO basado en UML. <http://www.cs.ualberta.ca/~pfiguero/soo/metod/uml-met.html>. 30 de Agosto del 2007.
- GUIAPESCA. Guía Técnica Para El Cultivo De “Camarón Marino”. <http://www.agronegocios.gob.sv/comoproducir/guias/GUIAPESCA%20CAMARON%20MARINO.pdf>. 16 de Enero del 2007.
- Microsoft Corporation.ASP.Net. <http://www.asp.net/ajax/>. 2008.
- Microsoft Corporation. AJAX Control Toolkit Project Page. <http://www.asp.net/ajax/ajaxcontroltoolkit/samples/>. 2006-2007.
- UNIVERSITAT DE LES ILLES BALEARS. Departament de Ciències Matemàtiques i Informàtica, Selección De Herramientas Case. <http://dmi.uib.es/%7EElabsoft/Labsg/trabajo%20herramientas%20CASE.pdf>. 9 de Mayo del 2005.
- FACULTAD DE CIENCIAS ECONÓMICA. Universidad Nacional De Jujuy. Metodología Estructurada De Desarrollo De Sistemas. <http://www.fce.unju.edu.ar/download/si/unidadvi.pdf>. 29 de Abril del 2004.
- EXTREME PROGRAMMING. Extreme Programming: A gentle introduction. <http://www.extremeprogramming.org/>.17 de Febrero del 2006.
- MENDOZA, María. Metodologías De Desarrollo De Software. http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf. 7 de Junio del 2004.
- FOWLER, Martin; The New Methodology. <http://martinfowler.com/articles/newMethodology.html>. 13 de Diciembre del 2005.

- LETELIER, Patricio; PENADES, Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
<http://www.willydev.net/descargas/masyxp.pdf>. 29 de Abril del 2008.
- WIKIMEDIA FOUNDATION, INC. Desarrollo rápido de aplicaciones.
http://es.wikipedia.org/wiki/Desarrollo_r%C3%A1pido_de_aplicaciones. 26 de Agosto del 2008.
- ORACLE. Oracle Enterprise Manager 10g Release 2.
http://www.oracle.com/lang/es/enterprise_manager/index.html.2007.
- WIKIMEDIA FOUNDATION, INC. Oracle.
<http://es.wikipedia.org/wiki/Oracle>. 20 Abril 2008.
- SAM Sistemas. JBuilder® 2007.
<http://www.samsistemas.com.ar/partners/jbuilder.html>. 1992-2007.
- WIKIMEDIA FOUNDATION, INC. Microsoft Visual Studio.
http://es.wikipedia.org/wiki/Microsoft_Visual_Studio. 12 de Enero 2008.
- WIKIMEDIA FOUNDATION, INC .Front-end_y_back-end.
http://es.wikipedia.org/wiki/Front-end_y_back-end. 2 de Septiembre del 2008.
- WIKIMEDIA FOUNDATION, INC. Eclipse (software).
[http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software)). 13 Julio del 2008.
- MYSQL HISPANO. <http://www.mysql-hispano.org/articles.php>. 26 Agosto del 2002.
- IBM, Rational Rose Enterprise.
http://www-950.ibm.com/ecatalog/Detail.wss?locale=es_ES&synkey=M221280M46834Z27. 9 de Septiembre del 2007.

- ARSYS.ES. Servidores Dedicados - SQL Server 2005 Express Edition.
<http://www.arsys.es/ayuda/directorio/productos/servidores-dedicados/sql-server-express-edition.htm>. 2007.

TESIS

- URÍA, Juan." Sistema De Producción Camaronera".1991.
- RUIZ, José; CASTRO, Robert; ENCALADA, Jorge; MENDIETA, Rossana.
ESPOL. "Plan De Negocios De La Camaronera CREMSA".1999.

ANEXOS

GLOSARIO

AJAX acrónimo de *Asynchronous JavaScript And XML* (Java Script asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

ASD siglas de “Adaptative Software Development” es un software de proceso de desarrollo, que surgió del trabajo de Jim Highsmith y Sam Bayer con el desarrollo rápido de aplicaciones. ASD encarna el principio de que la adaptación continua del proceso para el trabajo a mano es el estado normal de las cosas.

ASP.NET es un framework para aplicaciones Web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios Web dinámicos, aplicaciones Web y servicios Web XML. ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

BOOTSTRAP es otra de las iniciativas para resolver la crisis del desarrollo de software. Esta metodología mediante prácticas, herramientas y estándares de calidad internacional; mide, evalúa y propone mejoras al proceso de desarrollo de Software que siguen las Unidades de Producción de Software (UPS) de las empresas.

Este proyecto al igual que otros, tiene como principio el reducir costos y mejorar la calidad previendo problemas, en lugar de reaccionar a ellos. Su objetivo es desarrollar un método para la evaluación de procesos de desarrollo de SW. Inicialmente se baso en CMM añadiendo conceptos de calidad de ISO 9000 y el modelo de proceso de ESA (European Space Agency)

CMM siglas de “Capability Maturity Model”, modelo desarrollado por SEI (Software Engineering Institute) en 1990, para la evaluación y mejora de los procesos. El primer modelo desarrollado para evaluar y mejorar los procesos fue el SW-CMM, por lo que muchas veces se hace referencia a él coloquialmente como “CMM”.

CRYSTAL CLEAR no es una metodología en si misma sino una familia de metodologías con un “código genético” común. La idea es poder armar distintas metodologías para distintos tipos de proyectos. Cada proyecto y organización usará este código genético para generar su propia metodología. El nombre Crystal deriva de la caracterización de los proyectos según 2 dimensiones, tamaño y complejidad (como en los minerales, color y dureza).

DSDM siglas de “Dynamic Software Development Method” es un enfoque de desarrollo de software originalmente basada en el desarrollo rápido de aplicaciones (RAD). DSDM es iterativo e incremental con un enfoque que hace hincapié en la continua participación de los usuarios. Su objetivo es ofrecer sistemas de software en tiempo y en presupuesto, mientras que para ajustar las necesidades cambiantes a lo largo del proceso de desarrollo.

FDD siglas de “Feature Driven Development”, metodología ágil de desarrollo. No requiere un modelo específico de proceso y se complementa con otras metodologías. Enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso. FDD consiste en cinco procesos secuenciales durante los que se diseña y construye el sistema: Desarrollo del modelo general – Construcción de la lista de rasgos – Planeamiento por rasgo – Diseño por rasgo – Construcción por rasgo.

FRAMEWORK es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un *framework* puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

ISO 9000 son normas de "calidad" y "gestión continua de calidad", establecidas por la Organización Internacional para la Estandarización (ISO) que se pueden aplicar en cualquier tipo de organización o actividad sistemática, que esté orientada a la producción de bienes o servicios. Se componen de estándares y guías relacionados con sistemas de gestión y de herramientas específicas como los métodos de auditoría (el proceso de verificar que los sistemas de gestión cumplen con el estándar).

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información.

.NET Proyecto de Microsoft, nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones.

ORM es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos orientada a objetos virtual, por sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

RAD es un proceso de desarrollo de software, en inglés: software development process, desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering).

RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

SCRUM es un proceso de desarrollo de software iterativo e incremental utilizado comúnmente en entornos ágiles de desarrollo de software. Metodología ágil, aplicada originalmente por Jeff Sutherland y elaborada más formalmente por Ken Schwaber. Scrum aplica principios de control industrial, junto con experiencias metodológicas de Microsoft, Borland y Hewlet Packard.

SDCE la Capacidad de Desarrollo de Software de Evaluación (SDCE) es un método que aborda no solo la capacidad de ingeniería de software, sino también y más importante, los sistemas de ingeniería y disciplinas relacionadas con el desarrollo integral para el desarrollo con éxito de un programa intensivo de sistemas de defensa.

SPICE la metodología spice tienen sus orígenes en el creciente uso y dependencia de la Tecnología de Información, que en consecuencia dio el incremento de frustración e incumplimiento de expectativas por parte de los desarrolladores y los usuarios de software.

TICKIT la metodología TickIT está especialmente orientada a las empresas que desarrollan software. Abarca un campo muy amplio ya que tanto empresas como personas dependen de forma directa o indirecta de los sistemas de información.

TRILLIUM Basado en CMM e ISO 9000. Orientado a entornos relacionados con las telecomunicaciones, cubre todos los aspectos relativos al ciclo de vida del desarrollo de software

UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de

lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

XP Es la más destacada de los procesos ágiles de desarrollo de software. Al igual que éstos, la *Programación Extrema* se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.