

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



ESCUELA DE DISEÑO Y COMUNICACIÓN VISUAL
INFORME DE PROYECTO DE GRADUACIÓN
PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
ANALISTA DE SISTEMAS

TEMA

UTILIZACIÓN DE LA TECNOLOGÍA JEE COMO PLATAFORMA DE
DESARROLLO EN UNA APLICACIÓN INFORMÁTICA PARA
ANÁLISIS ACUÍCOLA EN UNA CAMARONERA

AUTOR

LUIS AUGUSTO GÓMEZ MOSQUERA

DIRECTOR

MSIG. LUIS RODRÍGUEZ

AÑO

2012

AGRADECIMIENTO

Agradezco a la Escuela Superior Politécnica del Litoral por llenar las expectativas en la carrera Análisis de Sistemas, preparándonos e introduciéndonos con el adecuado balaje de conocimientos teórico practico que nos exigía el campo profesional y así poder ser parte del desarrollo de nuevas tecnologías en software libre y sistemas informáticos que ayuden definitivamente a nuestro país a competir con mercados que nos llevan mucha más ventaja en la implementación de estas, mas no, en la capacidad para poder diseñarlas y desarrollarlas..

Luis Gómez Mosquera

DEDICATORIA

Dedico este trabajo a mis padres (Luis Gómez Junco y Carmen Mosquera de Gómez) y familiares a sus esfuerzos por conseguir nuestro desarrollo profesional en un ámbito en donde la tecnología y la innovación serán el marco propicio para nuestro desempeño en la sociedad del siglo XXI.

DECLARACIÓN EXPRESA

La responsabilidad del contenido de este Trabajo Final de Graduación, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral.

**FIRMA DEL DIRECTOR DEL PROYECTO Y MIEMBROS DEL
TRIBUNAL DE GRADO**

**Msig. Luis Rodríguez
DIRECTOR DEL PROYECTO**

DELEGADO

**FIRMA DEL AUTOR DEL PROYECTO DE
GRADUACIÓN**

Luis Augusto Gómez Mosquera

RESUMEN

Java Platform Enterprise Edition (ex J2EE) se ha convertido en un estándar de la industria de desarrollo de software que permite implementar aplicaciones empresariales basadas en servidores que sean seguras, portables, robustas y escalables. Java EE provee API's que permiten administrar entre otros, web services, modelo de componentes, comunicación entre objetos, que hacen posible el uso de la arquitectura empresarial orientada a servicios (SOA) y aplicaciones de siguiente (esta) generación (Web 2.0).

La utilización de la tecnología JEE como plataforma de desarrollo en una aplicación informática para análisis acuícola en una camaronera tiene por objetivo ser una ayuda para personal técnico (Biólogo, Acuicultor, Zootecnista) responsable de las actividades y análisis periódicos realizados desde el momento mismo que inicia el proceso de la explotación, es decir cuando la larva o semilla llega desde los distintos laboratorios en donde es cultivada, para luego ser traspasada a las piscinas en donde se dará inicio al ciclo de explotación propiamente dicho.

Las herramientas open source creadas para el desarrollo de aplicativos informáticos orientados a un ambiente web y entre las cuales se encuentran Vaadin y EclipseLink ambos componentes plenamente sustentables en la plataforma JEE, la misma que permitirá brindar todo un marco de desarrollo de manera integral en un ambiente Java, en donde se requerirá contar con elementos tales como:

Desarrollo Java para la capa Web

La capa web es aquella donde se reciben los requerimientos de los clientes y donde se construye el código de presentación que se ejecutará en los browsers de estos. Para esta capa se disponen de tecnologías tales como:

- Desarrollo con AJAX para producir aplicaciones web, esto es debido a que Vaadin cuenta con código embebido AJAX del cual el desarrollador no se percatará al programar, debido a la tecnología con que Java provee a este nuevo Framework.

Desarrollo Java para la capa de Negocio

La capa de negocio es aquella donde se implementan las reglas de negocio. Para esta capa se disponen de tecnologías tales como:

- Desarrollo con framework Vaadin (J2EE sin EJBs) Enterprise Java Bean.

Desarrollo Java para la capa de Persistencia

Las mejores prácticas del desarrollo por componentes sugieren desacoplar completamente las componentes de la capa lógica y de negocio de las componentes que entregan el servicio de persistencia (problema mapeo Objeto-Relacional) que permiten implementar la persistencia de los objetos de tipo entity. Esto entrega un diseño de persistencia con mucha más flexibilidad e independencia del tipo de base de datos. Para esta capa se dispone de tecnologías tales como:

- Desarrollos utilizando el patrón JPA (Java Persistencia Aplicación).
- Desarrollo e implementación de framework de persistencia EclipseLink.

Nuestra aplicación será capaz de entregar información que será vital para el usuario encargado de operarlo como también para elementos de otros departamentos (gerencia, administración) que también intervienen en una explotación camaronesa y necesitan conocer de la información que nos proporcionarán los análisis con los que cuenta la aplicación.

Entre las opciones con las que cuenta nuestra aplicación están el análisis de Recepción de larva que se hace para hacer el chequeo de cómo llega la larva antes de comenzar la explotación. El análisis de Fitoplancton que tiene por objetivo ver si la alimentación compuesta de organismos microscópicos (algas) es la correcta y no exista alguna deficiencia o incremento dañino por parte de esta que afecte al camarón. Y el análisis de control Patológico busca revisar en los órganos internos del camarón alguna presencia anómala producto de la alimentación suministrada en el manejo semi-intensivo utilizado.

ÍNDICE GENERAL

1. ANTECEDENTES	18
1.1 INTRODUCCIÓN	18
1.2 ESQUEMA Y FUNCIONALIDAD VAADIN	20
1.2.1 LOS SERVICIOS DE PRESENTACIÓN:.....	21
1.2.2 PROTOCOLOS:.....	21
1.2.3 ENTERPRISE DECISION MANAGEMENT:.....	21
1.2.3 SERVICES AND COMPONENTS:	21
1.2.4 SISTEMAS DE NEGOCIOS:.....	21
1.2.5 SISTEMAS LEGADOS:.....	21
1.2.6 ESB:	22
1.2.7 REGISTRO:.....	22
1.2.8 SERVIDORES DE APLICACIÓN:	22
1.3 PROBLEMA A RESOLVER	22
1.4 VENTAJAS Y DESVENTAJAS DE LA TECNOLOGÍA	23
2. EXPLICACIÓN DE LA TECNOLOGÍA	26
2.1 INTRODUCCIÓN	26
2.2 ECLIPSELINK JPA	26
2.3 UTILIZACIÓN DE ECLIPSELINK.	27
2.4 COMPONENTES VAADIN	30
2.4.1 LA ARQUITECTURA GENERAL DE VAADIN.....	31
2.4.2 ARQUITECTURA VAADIN.....	32
2.4.2.1 El apoyo para el Eclipse IDE	33
2.4.2.2 Componentes de Interfaz de usuario:	33
2.4.2.3 El Artefacto del lado cliente:.....	33
2.4.2.4 El Adaptador terminal:	33
2.4.2.5 Los temas:	34
2.4.2.6 UIDL:	34
2.4.2.7 Eventos:.....	34
2.4.2.8 Modelo de datos:	34
2.5 TECNOLOGÍAS SUBYACENTES	34
2.6 CONTROLES VAADIN	34
2.6.1 MANAGING LAYOUT	34
2.6.2 WINDOW Y DISEÑO DE RAÍZ DE TABLERO.....	36
2.6.3 VERTICALLAYOUT Y HORIZONTALLAYOUT.....	36
2.6.4 DATA BINDING.....	37
2.6.5 LA FECHA Y TIME INPUT CON DATEFIELD	37
2.6.5.1 DateField (PopupDateField) para Seleccionar Datos y Tiempo.....	38
2.6.6 BUTTON.....	38
2.6.7 TABLE	39
3. ALCANCE DEL PRODUCTO:	42

3.1 INTRODUCCIÓN	42
3.2 OPCIONES DEL SISTEMA	42
3.3 OPCIÓN MANTENIMIENTO DEL MENÚ PRINCIPAL:	42
3.3.1 CAMARONERA:	42
3.3.2 PISCINA:	43
3.3.3 ETAPAS CULTIVO:	43
3.3.4 Tipo de especie:	43
3.3.5 LABORATORIO:	43
3.4 OPCIÓN CULTIVO DEL MENÚ PRINCIPAL	44
3.4.1 CULTIVO:	45
3.4.2 OPCIÓN REGISTRO DE CULTIVO.....	45
3.5 OPCIÓN TRANSACCIÓN DEL MENÚ PRINCIPAL	46
3.5.1 CARGAMENTO:	46
3.5.2 CONTROL PATOLÓGICO:	47
3.5.3 CONTROL RECEPCIÓN DE LARVA:	50
3.5.4 CONTROL FITOPLANCTON:	51
4. DISEÑO DEL SISTEMA (REQUERIMIENTOS, MODELO DE OBJETOS, CASOS DE USOS IMPORTANTES, DIAGRAMA DE IMPLEMENTACIÓN) ..	55
4.1 REQUERIMIENTOS	55
4.1.1 INTRODUCCIÓN.....	55
4.2 CASOS DE USOS	55
4.2.1 INTRODUCCIÓN.....	55
4.3 DIAGRAMA SUBCASOS DE USO.....	56
4.3.1 INTRODUCCIÓN.....	56
4.4 DESCRIPCIÓN DE CASOS DE USOS.....	58
4.4.1 INTRODUCCIÓN.....	58
4.4.2 DESCRIPCIÓN SUBCASOS DE USOS	59
4.5 EVENTOS	60
4.5.1 INTRODUCCIÓN.....	60
4.6 DIAGRAMA CONCEPTUAL.....	66
4.6.1 INTRODUCCIÓN.....	66
4.7 DIAGRAMA DE ESTRUCTURA ESTÁTICA (DE CLASES)	67
4.7.1 INTRODUCCIÓN.....	67
4.8 DIAGRAMA DE INTERACCIÓN Y COLABORACIÓN.....	68
4.8.1 INTRODUCCIÓN.....	68
4.8.2 DIAGRAMA INTERACCIÓN INGRESO DATOS ANÁLISIS FITOPLANCTON.....	69
4.8.2.1 DIAGRAMA COLABORACIÓN INGRESO DATOS ANÁLISIS FITOPLANCTON	70
4.8.3 DIAGRAMA INTERACCIÓN ACTUALIZACIÓN DATOS ANÁLISIS FITOPLANCTON.....	71
4.8.3.1 Diagrama colaboración actualización datos Análisis Fitoplancton	72

4.8.4	DIAGRAMA INTERACCIÓN ELIMINACIÓN DATOS ANÁLISIS FITOPLANCTON	73
4.8.4.1	Diagrama colaboración eliminación datos Análisis Fitoplancton	74
4.8.5	DIAGRAMA INTERACCIÓN INGRESO DATOS CONTROL PATOLÓGICO	75
4.8.5.1	Diagrama colaboración ingreso datos Control Patológico	76
4.8.6	DIAGRAMA INTERACCIÓN ACTUALIZACIÓN DATOS CONTROL PATOLÓGICO.....	77
4.8.6.1	Diagrama colaboración actualización datos Control Patológico	78
4.8.7	DIAGRAMA INTERACCIÓN ELIMINACIÓN DATOS CONTROL PATOLÓGICO.....	79
4.8.7.1	Diagrama colaboración eliminación datos Control Patológico	80
4.8.8	DIAGRAMA INTERACCIÓN INGRESO DATOS RECEPCIÓN DE LARVA.....	81
4.8.8.1	Diagrama colaboración ingreso datos Recepción de Larva.....	82
4.8.9	DIAGRAMA INTERACCIÓN ACTUALIZACIÓN DATOS RECEPCIÓN DE LARVA	83
4.8.9.1	Diagrama colaboración actualización datos Recepción de Larva	84
4.8.10	DIAGRAMA INTERACCIÓN ELIMINACIÓN DATOS RECEPCIÓN DE LARVA	85
4.8.10.1	Diagrama colaboración eliminación datos Recepción de Larva.....	86
4.9	DIAGRAMAS DE ESTADO Y ACTIVIDADES	87
4.9.1	INTRODUCCIÓN.....	87
4.9.1.1	Diagrama de Estado ingreso datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva	87
4.9.1.2	Diagrama de Estado actualización datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva.....	88
4.9.1.3	Diagrama de Estado eliminación datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva.....	89
4.9.2	DIAGRAMA ACTIVIDADES	89
4.9.2.1	Diagrama de Actividades ingreso datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva.....	90
4.9.2.2	Diagrama de Actividades actualización datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva.....	91
4.9.2.3	Diagrama de Actividades eliminación datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva.....	92
5.	USO DEL PRODUCTO	94
5.1	PANTALLA SEGURIDAD DEL SISTEMA	95
5.2	PANTALLA PRINCIPAL DEL SISTEMA	95
5.3	PANTALLA PRINCIPAL OPCIÓN MANTENIMIENTO	96
5.4	PANTALLA CAMARONERA: SUBOPCIÓN MANTENIMIENTO DEL MENÚ PRINCIPAL DEL SISTEMA.....	96
5.5	PANTALLA PISCINA: SUBOPCIÓN MANTENIMIENTO DEL MENÚ PRINCIPAL DEL SISTEMA	97
5.6	PANTALLA ETAPAS CULTIVO: SUBOPCIÓN MANTENIMIENTO DEL MENÚ PRINCIPAL	97
5.7	PANTALLA TIPO ESPECIE: SUBOPCIÓN MANTENIMIENTO DEL MENÚ PRINCIPAL	98
5.8	PANTALLA LABORATORIO: SUBOPCIÓN MANTENIMIENTO DEL MENÚ PRINCIPAL DEL SISTEMA.....	98

5.9 PANTALLA PRINCIPAL DEL SISTEMA: OPCIÓN CULTIVO DEL MENÚ PRINCIPAL	99
5.10 PANTALLA CULTIVO: SUBOPCIÓN CULTIVO DEL MENÚ PRINCIPAL DEL SISTEMA	99
5.11 PANTALLA REGISTRO DE CULTIVO: SUBOPCIÓN CULTIVO DEL MENÚ PRINCIPAL DEL SISTEMA.....	100
5.12 PANTALLA PRINCIPAL DEL SISTEMA: OPCIÓN TRANSACCIÓN DEL MENÚ PRINCIPAL	100
5.13 PANTALLA CARGAMENTO: SUBOPCIÓN TRANSACCIÓN DEL MENÚ PRINCIPAL SISTEMA	101
5.14 PANTALLA ANÁLISIS RECEPCIÓN DE LARVA: SUBOPCIÓN TRANSACCIÓN DEL MENÚ PRINCIPAL DEL SISTEMA.....	101
5.15 PANTALLA ANÁLISIS CONTROL PATOLÓGICO: SUBOPCIÓN TRANSACCIÓN DEL MENÚ PRINCIPAL DEL SISTEMA.....	102
5.16 PANTALLA ANÁLISIS DE FITOPLANCTON: SUBOPCIÓN TRANSACCIÓN DEL MENÚ PRINCIPAL DEL SISTEMA.....	102
5.17 PANTALLA CONSULTAR: ANÁLISIS DE FITOPLANCTON, CONTROL PATOLÓGICO Y RECEPCIÓN DE LARVA	103
5.18 PANTALLA REPORTE CONTROL PATOLÓGICO: OPCIÓN IMPRIMIR DEL FORMULARIO CONTROL PATOLÓGICO	103
5.19 PANTALLA REPORTE ANÁLISIS DE FITOPLANCTON: OPCIÓN IMPRIMIR DEL FORMULARIO ANÁLISIS DE FITOPLANCTON.....	104
5.20 PANTALLA REPORTE ANÁLISIS DE RECEPCIÓN DE LARVA: OPCIÓN IMPRIMIR DEL FORMULARIO ANÁLISIS RECEPCIÓN DE LARVA	104
6. ESPECIFICACIONES TÉCNICAS	106
7. SUGERENCIAS Y RECOMENDACIONES.....	108
8. ANEXOS:	110
8.1 CONFIGURACIÓN DEL JPA.....	110
8.2 INSTALANDO Y CONFIGURANDO ECLIPSELINK.....	110
8.3 ECLIPSELINK INSTALANDO (JAVASE & EE).....	110
8.4 LAS VARIABLES DE AMBIENTE DE SISTEMA	110
8.5 LA INSTALACIÓN.....	111
BIBLIOGRAFÍA	112

Índice de Figuras

CAPÍTULO 1.

FIGURA. 1.1 ESQUEMA ARQUITECTURA JEE.....	19
FIGURA. 1.2 ESQUEMA VAADIN TOMADO DE HTTP://WWW.VAADIN.COM/BOOK	20

CAPÍTULO 2.

FIGURA. 2.1 OPCIÓN JPA ENTITIES FROM TABLES	10
FIGURA. 2.2 SELECCION DE BASE DE DATOS	11
FIGURA. 2.3 CLAVE DE SEGURIDAD Y HOST	11
FIGURA. 2.4 ELECCIÓN DE TABLAS.....	12
FIGURA. 2.5 AGREGAR RELACIONES ENTRE LAS TABLAS	12
FIGURA. 2.6 CREAR NUEVAS RELACIONES ENTRE TABLAS	13
FIGURA. 2.7 ESQUEMA DE ARQUITECTURA VAADIN TOMADO DE BOOK OF VAADIN (VAADIN LTD. MARKO GRONROOS)	14
FIGURA. 2.8 ARQUITECTURA VAADIN TOMADO DE BOOK OF VAADIN (VAADIN LTD. MARKO GRONROOS)	15
FIGURA. 2.9 VERTICALLAYOUT.....	21
FIGURA. 2.10 HORIZONTALLAYOUT.....	21
FIGURA. 2.11 POPUPDATEFIELD.....	22
FIGURA. 2.12 BUTTON	23

CAPÍTULO 4

FIGURA. 4.1 DIAGRAMA CASOS DE USOS	38
FIGURA. 4.2 DIAGRAMA SUBCASOS DE USOS.....	39
FIGURA. 4.3 DIAGRAMA CONCEPTUAL	48
FIGURA. 4.4 DIAGRAMA DE CLASES	49
FIGURA. 4.5 DIAGRAMA INTERACCIÓN INGRESO DATOS ANÁLISIS FITOPLANCTON	51
FIGURA. 4.6 DIAGRAMA COLABORACIÓN INGRESO DATOS ANÁLISIS FITOPLANCTON.....	52
FIGURA. 4.7 DIAGRAMA INTERACCIÓN ACTUALIZACIÓN DATOS ANÁLISIS FITOPLANCTON	53
FIGURA. 4.8 DIAGRAMA COLABORACIÓN ACTUALIZACIÓN DATOS ANÁLISIS FITOPLANCTON	54
FIGURA. 4.9 DIAGRAMA INTERACCIÓN ELIMINACIÓN DATOS ANÁLISIS FITOPLANCTON.....	55
FIGURA. 4.10 DIAGRAMA COLABORACIÓN ELIMINACIÓN DATOS ANÁLISIS FITOPLANCTON	56
FIGURA. 4.11 DIAGRAMA INTERACCIÓN INGRESO DATOS CONTROL PATOLÓGICO	57
FIGURA. 4.12 DIAGRAMA COLABORACIÓN INGRESO DATOS CONTROL PATOLÓGICO.....	58
FIGURA. 4.13 DIAGRAMA INTERACCIÓN ACTUALIZACIÓN DATOS CONTROL PATOLÓGICO	59
FIGURA. 4.14 DIAGRAMA COLABORACIÓN ACTUALIZACIÓN DATOS CONTROL PATOLÓGICO	60
FIGURA. 4.15 DIAGRAMA INTERACCIÓN ELIMINACIÓN DATOS CONTROL PATOLÓGICO	61
FIGURA. 4.16 DIAGRAMA COLABORACIÓN ELIMINACIÓN DATOS CONTROL PATOLÓGICO	62
FIGURA. 4.17 DIAGRAMA INTERACCIÓN INGRESO DATOS RECEPCIÓN DE LARVA	63
FIGURA. 4.18 DIAGRAMA COLABORACIÓN INGRESO DATOS RECEPCIÓN DE LARVA	64
FIGURA. 4.19 DIAGRAMA INTERACCIÓN ACTUALIZACIÓN DATOS RECEPCIÓN DE LARVA.....	65
FIGURA. 4.20 DIAGRAMA COLABORACIÓN ACTUALIZACIÓN DATOS RECEPCIÓN DE LARVA	66
FIGURA. 4.21 DIAGRAMA INTERACCIÓN ELIMINACIÓN DATOS RECEPCIÓN DE LARVA	67
FIGURA. 4.22 DIAGRAMA COLABORACIÓN ELIMINACIÓN DATOS RECEPCIÓN DE LARVA.....	68

FIGURA. 4.23 DIAGRAMA DE ESTADO INGRESO DATOS ANÁLISIS: FITOPLANCTON, CONTROL PATOLÓGICO, RECEPCIÓN DE LARVA	69
FIGURA. 4.24 DIAGRAMA DE ESTADO ACTUALIZACIÓN DATOS ANÁLISIS: FITOPLANCTON, CONTROL PATOLÓGICO RECEPCIÓN DE LARVA	70
FIGURA. 4.25 DIAGRAMA DE ESTADO ELIMINACIÓN DATOS ANÁLISIS: FITOPLANCTON, CONTROL PATOLÓGICO, RECEPCIÓN DE LARVA.....	71
FIGURA. 4.26 DIAGRAMA DE ACTIVIDADES INGRESO DATOS ANÁLISIS: FITOPLANCTON, CONTROL PATOLÓGICO, RECEPCIÓN DE LARVA.....	72
FIGURA. 4.27 DIAGRAMA DE ACTIVIDADES ACTUALIZACIÓN DATOS ANÁLISIS: FITOPLANCTON, CONTROL PATOLÓGICO, RECEPCIÓN DE LARVA.....	73
FIGURA. 4.28 DIAGRAMA DE ACTIVIDADES ELIMINACIÓN DATOS ANÁLISIS: FITOPLANCTON, CONTROL PATOLÓGICO, RECEPCIÓN DE LARVA.....	74

CAPÍTULO 5

FIGURA. 5.1 SEGURIDAD.....	77
FIGURA. 5.2 MENÚ PRINCIPAL.....	77
FIGURA. 5.3 OPCIÓN MANTENIMIENTO	78
FIGURA. 5.4 FORMULARIO CAMARONERA	78
FIGURA. 5.5 FORMULARIO PISCINA	79
FIGURA. 5.6 FORMULARIO DE ETAPAS.....	79
FIGURA. 5.7 FORMULARIO DE ESPECIES	80
FIGURA. 5.8 FORMULARIO LABORATORIO.....	80
FIGURA. 5.9 OPCIÓN PRINCIPAL CULTIVO	81
FIGURA. 5.10 FORMULARIO APERTURA DE CULTIVO	81
FIGURA. 5.11 FORMULARIO PROCESO DE CULTIVO	82
FIGURA. 5.12 OPCIÓN PRINCIPAL TRANSACCIÓN.....	82
FIGURA. 5.13 FORMULARIO CARGAMENTO	83
FIGURA. 5.14 FORMULARIO DE RECEPCIÓN	83
FIGURA. 5.15 FORMULARIO DE CONTROL PATOLÓGICO.....	84
FIGURA. 5.16 FORMULARIO ANÁLISIS DE FITOPLANCTON.....	84
FIGURA. 5.17 PANTALLA CONSULTAR.....	85
FIGURA. 5.18 REPORTE DE CONTROL PATOLÓGICO.....	85
FIGURA. 5.19 REPORTE ANÁLISIS DE FITOPLANCTON.....	86
FIGURA. 5.20 REPORTE ANÁLISIS DE RECEPCIÓN DE LARVA.....	86

Índice de Tablas

CAPÍTULO 2

TABLA 2.1 METODOS ENTITYMANAGER	27
---------------------------------------	----

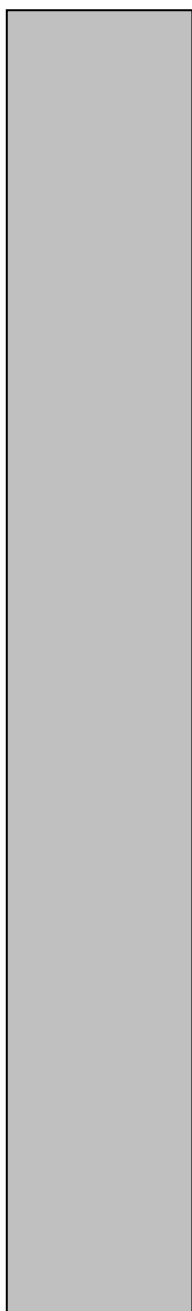
CAPÍTULO 3

TABLA 3.1 DESCRIPCIÓN DE OPCIONES CAMARONERA.	43
TABLA 3.2 DESCRIPCIÓN DE OPCIONES PISCINA.....	43
TABLA 3.3 DESCRIPCIÓN DE OPCIONES LABORATORIO.....	44
TABLA 3.4 DESCRIPCIÓN DE OPCIONES GENERALES DEL MENÚ MANTENIMIENTO.	44
TABLA 3.5 DESCRIPCIÓN DE OPCIONES CULTIVO.....	45
TABLA 3.6 DESCRIPCIÓN DE OPCIONES REGISTRO DE CULTIVO.....	46
TABLA 3.7 DESCRIPCIÓN DE OPCIONES CARGAMENTO.	46
TABLA 3.8 DESCRIPCIÓN DE OPCIONES CONTROL PATOLÓGICO.	50
TABLA 3.9 DESCRIPCIÓN DE OPCIONES CONTROL DE RECEPCIÓN DE LARVA.	51
TABLA 3.10 DESCRIPCIÓN DE OPCIONES CONTROL FITOPLANCTON.	53

CAPÍTULO 4

TABLA 4.1 REQUERIMIENTOS.	55
TABLA 4.2 CASO DE USO CONTROL ANÁLISIS DE FITOPLANCTON.	58
TABLA 4.3 CASO DE USO CONTROL PATOLÓGICO.....	58
TABLA 4.4 CASO DE USO ANÁLISIS RECEPCIÓN DE LARVA.	58
TABLA 4.5 SUBCASO DE USO INGRESO CONTROL PATOLÓGICO.....	59
TABLA 4.6 SUBCASO DE USO MODIFICACIÓN CONTROL PATOLÓGICO.....	59
TABLA 4.7 SUBCASO DE USO ELIMINACIÓN CONTROL PATOLÓGICO.	59
TABLA 4.8 SUBCASO DE USO INGRESO ANÁLISIS RECEPCIÓN DE LARVA.	59
TABLA 4.9 SUBCASO DE USO MODIFICACIÓN ANÁLISIS DE RECEPCIÓN DE LARVA.	59
TABLA 4.10 SUBCASO DE USO ELIMINACIÓN ANÁLISIS DE RECEPCIÓN DE LARVA.....	60
TABLA 4.11 SUBCASO DE USO INGRESO ANÁLISIS DE FITOPLANCTON.	60
TABLA 4.12 SUBCASO DE USO MODIFICACIÓN ANÁLISIS DE FITOPLANCTON.	60
TABLA 4.13 SUBCASO DE USO ELIMINACIÓN ANÁLISIS DE FITOPLANCTON.....	60
TABLA 4.14 DESCRIPCIÓN DE EVENTO INGRESO ANÁLISIS FITOPLANCTON.....	61
TABLA 4.15 DESCRIPCIÓN DE EVENTOS ALTERNOS INGRESO ANÁLISIS FITOPLANCTON.	61
TABLA 4.16 DESCRIPCIÓN DE EVENTO MODIFICACIÓN ANÁLISIS FITOPLANCTON.	62
TABLA 4.17 DESCRIPCIÓN DE EVENTOS ALTERNOS MODIFICACIÓN ANÁLISIS FITOPLANCTON.....	62
TABLA 4.18 DESCRIPCIÓN DE EVENTO ELIMINACIÓN ANÁLISIS FITOPLANCTON	62
TABLA 4.19 DESCRIPCIÓN DE EVENTOS ALTERNOS ELIMINACIÓN ANÁLISIS FITOPLANCTON	62
TABLA 4.20 DESCRIPCIÓN DE EVENTO INGRESO ANÁLISIS CONTROL PATOLÓGICO	63
TABLA 4.21 DESCRIPCIÓN DE EVENTOS ALTERNOS INGRESO ANÁLISIS CONTROL PATOLÓGICO.....	63
TABLA 4.22 DESCRIPCIÓN DE EVENTO MODIFICACIÓN ANÁLISIS CONTROL PATOLÓGICO	63
TABLA 4.23 DESCRIPCIÓN DE EVENTOS ALTERNOS MODIFICACIÓN ANÁLISIS CONTROL PATOLÓGICO	63
TABLA 4.24 DESCRIPCIÓN DE EVENTO ELIMINACIÓN ANÁLISIS CONTROL PATOLÓGIC	64
TABLA 4.25 DESCRIPCIÓN DE EVENTOS ALTERNOS ELIMINACIÓN ANÁLISIS CONTROL PATOLÓGICO	64
TABLA 4.26 DESCRIPCIÓN DE EVENTO INGRESO ANÁLISIS RECEPCIÓN DE LARVA.....	64
TABLA 4.27 DESCRIPCIÓN DE EVENTOS ALTERNOS INGRESO ANÁLISIS RECEPCIÓN DE LARVA	64
TABLA 4.28 DESCRIPCIÓN DE EVENTO MODIFICACIÓN ANÁLISIS RECEPCIÓN DE LARVA.....	65
TABLA 4.29 DESCRIPCIÓN DE EVENTOS ALTERNOS MODIFICACIÓN ANÁLISIS RECEPCIÓN DE LARVA.....	65

TABLA 4.30 DESCRIPCIÓN DE EVENTO ELIMINACIÓN ANÁLISIS RECEPCIÓN DE LARVA	65
TABLA 4.31 DESCRIPCIÓN DE EVENTOS ALTERNOS ELIMINACIÓN ANÁLISIS RECEPCIÓN DE LARVA	65



CAPÍTULO 1
ANTECEDENTES

1. Antecedentes

1.1 Introducción

Con el fin de crear aplicaciones que sean mucho más económicas, pero con igual desempeño que las desarrolladas con software comercial, en donde intervienen aspectos de tipo legal tales como uso de derechos de autor y licencias, es que cada vez tiene una mayor acogida el desarrollo de aplicaciones Open Source (código abierto) que tuvieron sus inicios allá por 1998 cuando un grupo de desarrolladores en la web pensó en encontrar una alternativa al software de autor que pudiera ser válida y eficaz. Intuyendo con una visión a futuro que lo que vendría con la evolución de sistemas en la web ayudaría tanto a las personas, negocios o empresas a emprender futuros proyectos de desarrollo, bien sea para PC o web que repercutirían en su consecuente crecimiento.

Pero para este desarrollo Open Source necesitamos de una plataforma confiable y segura que además, se encuentre lo suficientemente difundida y asegure contar con el adecuado respaldo tecnológico para emprender futuros proyectos, tanto en las herramientas que nos permitirán crear la interfaz gráfica con la cual terminará interactuado el usuario así también con el producto que por su compatibilidad permitirá el adecuado almacenamiento de la información que le suministrará dicha interfaz gráfica.

Java EE (Java Enterprise Edition) anteriormente conocida como J2EE provee servicios que podemos implementar en las aplicaciones Web. Pero antes de llegar a este tipo de aplicaciones tuvo que transcurrir una necesaria evolución que partió desde utilizar la estrategia tradicional que consistía en aplicaciones compactas, las cuales por su diseño cerrado, en donde todo estaba concebido en una sola codificación o capa (controles de interfaz gráfica, conexión a base de datos, métodos y procedimiento además de las reglas del negocio) sin la adecuada separación de componentes, trayendo consigo problemas tales como escalabilidad, disponibilidad, seguridad e integración.

Con el fin de resolver estos problemas se generalizó la división en las aplicaciones utilizando el modelo de capas, el cual busca simplificar el desarrollo y clarificar las responsabilidades de cada uno de los componentes de un aplicativo. Con el tiempo este modelo paso a convertirse en un estándar de desarrollo en las aplicaciones web, diferenciándose claramente por su diseño en tres capas muy bien definidas por capa cliente, capa servidor y de datos, el cual poco a poco fue mejorándose y con la evolución de servidores de aplicaciones surgió la necesidad de crear una cuarta capa, para separar la lógica de negocio de la transaccional.

Este tipo de arquitectura proporciona mayor control sobre los elementos que entran en juego en el desarrollo de aplicaciones web.

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

Dentro de la plataforma Java EE disponemos de varios contenedores. Un contenedor es un entorno de ejecución específico para un conjunto de objetos de un determinado tipo y con unos fines concretos.

Los contenedores son la interface entre un componente y la funcionalidad de bajo nivel específica de la plataforma que soporta el componente. Por ejemplo, antes de poder ejecutar un componente web, un bean enterprise o un componente de una aplicación cliente, debe ensamblarse dentro de una aplicación J2EE y desplegarse dentro de su contenedor.

El proceso de ensamble implica especificar las configuraciones del servidor para cada componente de la aplicación J2EE y para la propia aplicación J2EE. Estas configuraciones personalizan el soporte subyacente proporcionado por el servidor J2EE, que incluye servicios como, seguridad, control de transacciones, etc. en la gráfica (Figura 1.1 Esquema Arquitectura JEE) podemos ver un esquema del uso de contenedores. En la aplicación sobre la cual tratará este documento, el aspecto de control transaccional estaría dado por JPA y EclipseLink.

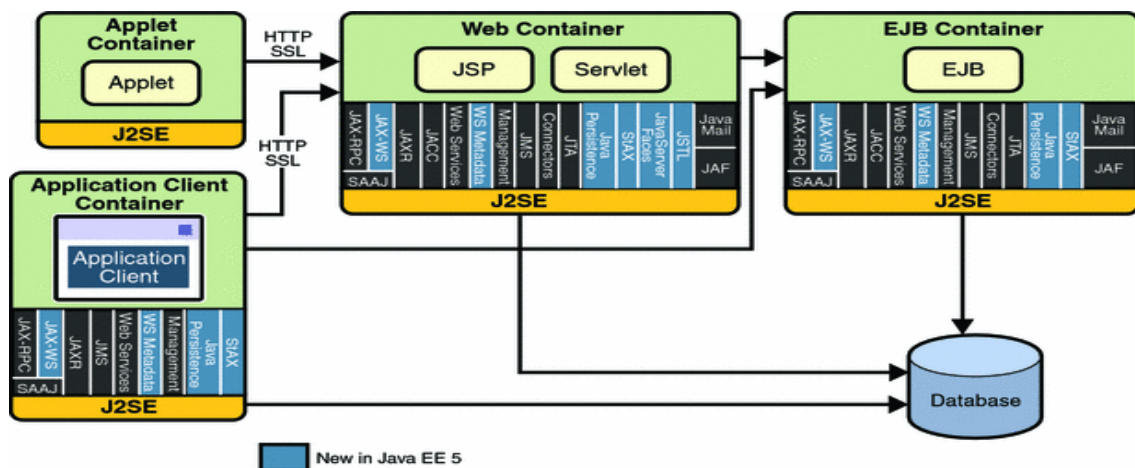


Figura. 1.1 Esquema arquitectura JEE

La plataforma JEE nos permitirá alcanzar la necesaria compatibilidad para hacer uso de Vaadin, que es un framework de aplicaciones web AJAX del lado del servidor que nos permitirá crear aplicaciones web, como si se programase soluciones convencionales usando el lenguaje Java, esto es, de la misma forma que se lo haría en Swing o AWT para diseñar la interfaz gráfica, aquella con la que interactuará finalmente el usuario. En la Figura 1.1 (Esquema arquitectura JEE) esta funcionalidad estaría representada por Contenedor de Clientes de Aplicación (Application Client Container).

Por medio de la tecnología JPA (Java Persistence Application) podemos crear objetos (tablas) en una base de datos valiéndonos de la programación orientada a objetos (POO) que nos brinda el lenguaje Java y su paradigma de diseño basado en clases.

JPA facilita la programación proveyéndole al desarrollador la creación de objetos sofisticados (tablas, relaciones, secuencias) a través de codificación en lenguaje Java ejecutándose dentro de una clase. De ahí que aunque no se haya creado físicamente la tabla en la base de datos, esto ya no tiene importancia porque JPA se encarga de crear la tabla del modelo de datos, una innovación que es realizada gracias a su mecanismo de persistencia de objetos, que no es otra cosa que la capacidad que tienen estos objetos para guardarse y recuperarse desde un medio de almacenamiento (Base de Datos).

Es decir si no existiera la tabla, JPA la crea y si ya existiera la usa. Además de ser lo de menos la base de datos que se utilice, en la aplicación sobre la cual tratará este documento será MySQL aunque podría haber sido ORACLE y el modelo creado en JPA sería igualmente valido con unos ligeros cambios.

1.2 Esquema y funcionalidad Vaadin

Pero no solo alcanzar una correcta interfaz gráfica que sea agradable y sencilla al usuario es lo único que nos ofrece Vaadin en un entorno JEE, sino además que, la información suministrada por el usuario será convenientemente tratada para que nos brinde no solo el clásico repositorio de información en una base de datos, tal como lo muestra el gráfico Esquema Vaadin (fig. 1.2) a continuación.



Figura. 1.2 Esquema Vaadin tomado de <http://www.vaadin.com/book>

En el esquema Vaadin tanto la funcionalidad como los servicios que provee su framework se hayan representados en la Figura 1.2 a continuación se realizara una

descripción de cada uno de ellos para tener una visión general de toda la tecnología que es capaz de suministrar Vaadin en un entorno de desarrollo.

1.2.1 Los servicios de presentación: Otorgados por interfaces de usuario que brindan una presentación de alta calidad visual, que facilitan el trabajo a diseñadores a través de una serie de controles y componentes gráficos fáciles de programar en código Java.

1.2.2 Protocolos: Los protocolos son reglas de comunicación que permiten el flujo de información entre equipos que manejan lenguajes distintos, por ejemplo, dos computadores conectados en ambiente de red o web pero con protocolos diferentes no podrían comunicarse jamás, es debido a esto, que es primordial contar con un estándar que permitirá la comunicación entre ambos. Entre los protocolos de comunicación con que cuenta Vaadin están SOAP, HTTP, JMS.

1.2.3 Enterprise Decision Management: Por medio de los cuales Vaadin está en la capacidad de crear aplicaciones robustas en organizaciones que manejan planes de decisiones automatizadas que buscan la interacción entre los diferentes entes que intervienen en sus operaciones. En el aplicativo estarían claramente identificados por el usuario responsable (técnico) de la información a procesar y almacenar (Análisis Acuícolas). La computarización ha cambiado las organizaciones y a través de Sistemas Manejadores de Base de Datos Relaciones (BRMS) les han otorgado una capacidad cada vez mayor de manejo de información vital para su desarrollo y expansión.

1.2.3 Services and Components: Vaadin cuenta por un lado un gran conjunto de componentes OUT-OF-THE-BOX (fuera de la caja) que permiten desarrollar una aplicación hasta en un 95% y por otro lado que se puedan crear nuevos componentes de así ser requeridos.

1.2.4 Sistemas de Negocios: con Vaadin podemos crear aplicaciones como si se estuviera programando aplicaciones para escritorio en Swing o AWT utilizando el modelo Orientado a Objeto que nos brinda el lenguaje Java para desarrollar sistemas en los ámbitos más diversos como el financiero, comercial (RRHH) o pecuario (acuícola).

1.2.5 Sistemas legados: Hace 10 años muchos pronosticaban la muerte de los dinosaurios tecnológicos (sistemas legados corriendo en mainframes), los pronósticos eran para los próximos 3 o 4 años el tiempo ha transcurrido pero aun sigue siendo inevitable su desaparición.

El problema va mucho más allá de los altos costos, que significaría su reemplazo, esta también otros aspectos como la falta de personal especializado, documentación y flexibilidad. Durante los últimos años han bajado tanto los costos como el desempeño de los mainframes debido al auge de los middleware que son los que proporcionan un enlace entre aplicaciones de software independiente (Vaadin). Los middleware establecen vías de comunicación que conectan dos aplicaciones y permiten pasar los

datos entre ellas, estos también permiten que los datos contenidos en una base de datos puedan ser accedidos por otra. Con buenas aplicaciones de middleware es posible actuar como si el sistema legado no existiera, construir nuevas funcionalidades alrededor del sistema legado minimizando al máximo la necesidad de modificar el sistema legado.

1.2.6 ESB: el término ESB (Enterprise Service Bus) es un término de marketing que inventaron las empresas que fabrican soluciones puras de middleware (establecer una vía de comunicación entre dos aplicaciones) para entornos empresariales dentro de los esquemas de arquitecturas.

Son soluciones creadas para un rol específico, en general deberían dar más soluciones a temas que están fuera del alcance SOA y Webservice en términos de integración. Permiten interactuar de mejor forma y más rápida a los componentes técnicos y de información con aquellos relacionados a los procesos de negocio de las capas superiores o más altas como las de tipo presentación expuestas en la (Fig. 1.2).

1.2.7 Registro: Vaadin cuenta con componentes tales como Tables que muestran registros de la información que requiera ser consultada de dispositivos tales como bases de Datos o archivos.

1.2.8 Servidores de aplicación: Java EE provee estándares que permiten a un servidor de aplicaciones servir como "contenedor" de los componentes que conforman dichas aplicaciones. Estos componentes, escritos en lenguaje Java, usualmente se conocen como Servlets, Java Server Pages (JSPs) y Enterprise JavaBeans (EJBs) y permiten implementar diferentes capas de la aplicación, como la interfaz de usuario, la lógica de negocio, la gestión de sesiones de usuario o el acceso a bases de datos remotas.

Entre estos están componentes que permiten incluir contenido Flash, Web Content que permiten incluir contenido de otras webs. Componentes para entrada de datos como por ejemplo Twin column select (list buider) Text y Label entre los más utilizados.

1.3 Problema a resolver

Nuestra aplicación está orientada a resolver el manejo de la información proveniente de los análisis diarios realizados en el laboratorio de una camaronera tiene por consiguiente el fin de prestar una ayuda vital en la toma de decisiones en base a dichos análisis tales como el análisis de Fitoplancton que registrará y procesará la información que tiene que ver con cuál ha sido la producción de algas en porcentajes estimativos en una piscina, esta información es preponderante dado el tipo de manejo emprendido en una explotación acuícola.

Si el manejo establecido es de tipo semi-intensivo el régimen alimenticio en una piscina en producción estará dado tanto por insumos balanceados, (alimentación artificial) como de forma natural por medio de los microorganismos, que se encuentran

suspendidos (algas) en el agua estuarina de la piscina, provenientes del bombeo diario del estero más cercano. Es por esto, fundamental el manejo de la información para un eficiente control de estas algas en la producción de una piscina.

Esta también el análisis de Control Patológico que se ocupará de registrar parámetros tales como peso promedio, aspecto general, piscina en la cual ha sido tomada la muestra, estado de branquias, estado de hepatopáncreas y estado del intestino, información que será revisada luego semanalmente buscado conocer cuál ha sido el desarrollo del camarón.

Por último está el análisis de Recepción de Larva que busca captar la información de la semilla (larva) que proviene de los diferentes laboratorios proveedores que suministran a la camaronera de este primordial recurso de explotación, es por esto, que la aplicación estará en la capacidad de hacer un seguimiento desde que está llega a la camaronera, buscando registrar y procesar parámetros como supervivencia, mortalidad, lugar de procedencia de dicha semilla.

1.4 Ventajas y desventajas de la tecnología

Ventajas:

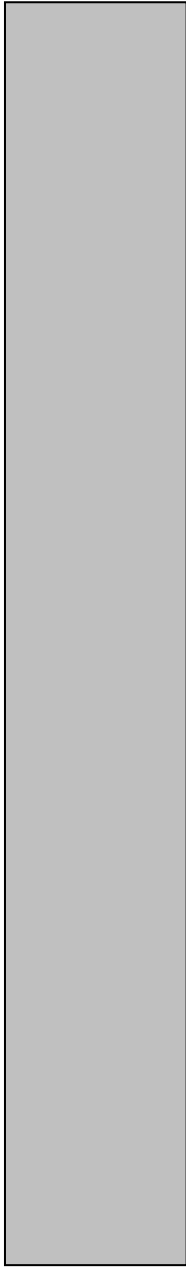
Notoria y sustancial baja en costos de desarrollo al estar utilizando herramientas Open Source de manera íntegra en todos los componentes utilizados.

Contar con el suficiente respaldo en material actualizado y documentación publicada en la web para las empresas y desarrolladores que estimulan y promueven las tecnologías Open Source a través de productos como Vaadin, JPA, Birt, MySQL y EclipseLink que serán justamente los escogidos para desarrollar la aplicación.

Desventajas:

Una vez escogida como plataforma de desarrollo las tecnologías en código abierto, JEE (Java Enterprise Edition) en particular provee de una serie de herramientas compatibles a su entorno como el IDE (Entorno de desarrollo Integrado) de Eclipse, en el cual se tendrá que configurar cada uno de los componentes, que se utilicen, comenzando desde el servidor web que en el caso de la aplicación será Apache Tomcat, luego Vaadin con el cual se desarrollará la interfaz con el cliente seguido del JPA para crear el modelo de objetos en la base y de manera simultánea tocara configurar el EclipseLink que permitirá realizar las relaciones entre las tablas por medio de su propia interfaz gráfica y por último el BIRT que es el componente que se encargará de realizar los reportes con los que mostraremos la información al usuario. De ahí que toda esta serie de pasos en la configuración de cada uno de estos componentes, además de plugins adicionales que son requeridos en la configuración, deben ser tomados en cuenta, el no hacerlo registrará problemas y errores en la compilación final que se realice. He aquí, la

importancia de que todas estas configuraciones deben realizarse previamente a ponernos a desarrollar la aplicación en sí, esto puede resultar molesto en comparación con otros productos comerciales, como .NET de Visual Studio en donde toda esta etapa de configuración de componentes ya viene establecida en su entorno de desarrollo facilitándole y a la vez ayudando al profesional a ocuparse en otros aspectos de la aplicación.



CAPÍTULO 2
**EXPLICACIÓN DE
LA TECNOLOGÍA**

2. Explicación de la tecnología

2.1 Introducción

Para la implementación de la plataforma JEE, IBM creó Eclipse, actualmente en manos Fundación Eclipse la cual ha continuado en su desarrollo hasta conseguir brindar una herramienta segura en la cual se tienen que configurar cada uno de los componentes a utilizar con el fin de tener una óptima integración entre estos. Además este nuevo marco tecnológico de desarrollo cuenta con un entorno que facilita la labor del programador al proveerle contar, por ejemplo, con herramientas tales como la **Interfaz de programación de aplicaciones API** (Application Programming Interface) que son un conjunto de funciones, procedimientos o métodos utilizados en la programación orientada a objetos conocidas comúnmente con el nombre de librerías por su capacidad de aglutinarlas en nombre específicos.

Entre los componentes a configurar dentro del entorno JEE esta Vaadin que será quien nos proporcionara el crear nuestra interfaz gráfica mediante una codificación sencilla y fácil de entender aun para un recién iniciado en campo del desarrollo en web.

De igual forma dentro del entorno Eclipse tenemos que configurar JPA que es el framework que nos permitirá crear nuestro modelo de Objetos (tablas) único con el que crearemos las tablas en nuestra base de Datos. Junto a JPA interactuará de forma directa EclipseLink que una vez configurado en el entorno JEE, mediante su propia interfaz gráfica nos ayudará a crear las relaciones entre las tablas u objetos creados de manera previa por JPA.

Por último la configuración de BIR en Eclipse nos permitirá manejar toda la reportería que será la que nos permitirá mostrar la información de una manera visual al usuario en un entorno web.

2.2 EclipseLink JPA

Una vez creadas las tablas en la base de datos mediante la Api (Interfaz de programación de aplicaciones) de Java Persistencia Aplicación (JPA) que es un framework para el trabajo con base de datos relacionales compuesto por tres parte:

- El paquete: javax.persistence.
- El lenguaje de consultas de persistencia.
- Los metadatos para los objetos y sus relaciones.

EclipseLink provee de una interfaz gráfica con la cual se crea de una manera sencilla las relaciones entre las tablas sin preocuparse mayormente por la codificación pues esta es realizada por EclipseLink en procesos internos que escapan a la vista del usuario.

Además de ser un estándar basado en Objecto-Relacion “OR” para java que facilita la implementación de Java Persistence Application “JPA” gracias a su interfaz gráfica de creación de persistencia la cual implementa los EntityManager API que manejan un

conjunto de entidades, determinadas por la instancia asociada, la cual permite ejecutar peticiones a la base de datos, como si se estuviera utilizando las sentencias básicas crear, modificar, eliminar, insertar de SQL.

Todo gracias al uso de los EntityManager métodos los cuales proveen anotaciones que manejan el aspecto transaccional en la base de datos. Entre las anotaciones más importantes, contamos con las mostradas en la tabla 2.1 Método EntityManager.

Métodos	Definición
EntityManager merge()	Permite actualizar un registro de la base de datos eliminándolo y luego creándolo
EntityManager remove()	Remueve el objeto de la base de datos
EntityManager find()	Busca un registro por medio de su clave primaria.

Tabla 2.1 Métodos EntityManager

2.3 Utilización de EclipseLink.

Creado el proyecto, se sitúa en la raíz del mismo, y dando clic derecho se elige la opción **New** se desplegará un submenú de opciones de las cuales señalamos **JPA Entities from Tables** tal como se muestra en el gráfico a continuación (Fig. 2.1).

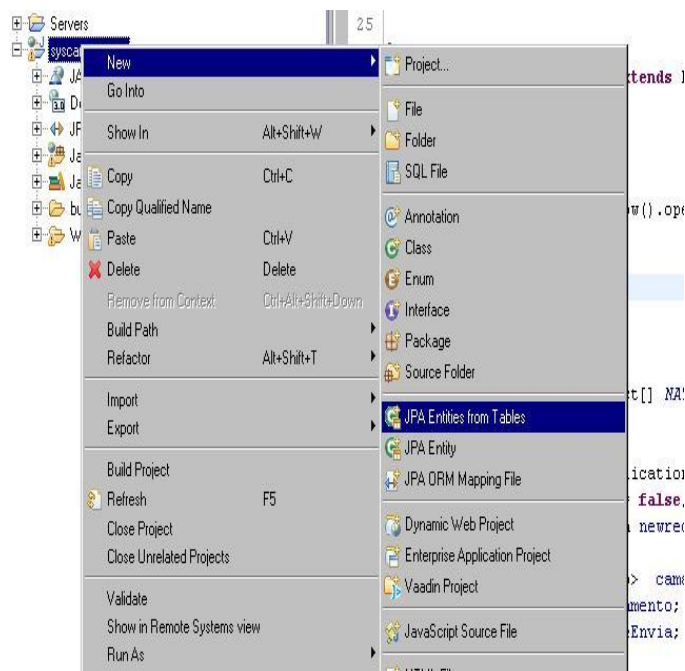


Figura. 2.1 Opción JPA entities from tables

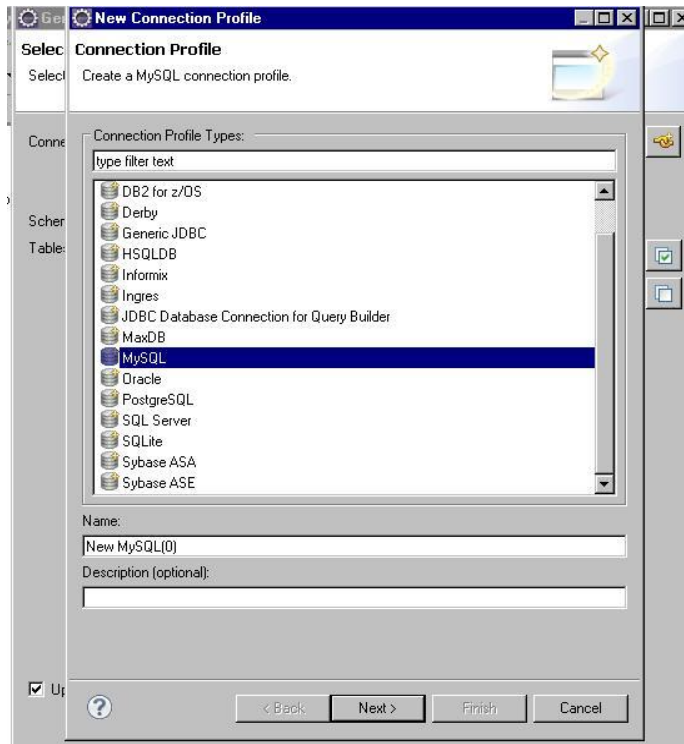


Figura. 2.2 Selección de Base de Datos

Se seleccionará la base de datos MySQL (Fig. 2.2) dando clic en botón Next el cual trasladará a otra pantalla en la que se introducirán los campos de acceso: usuario, clave, host como lo muestra la figura 2.3.

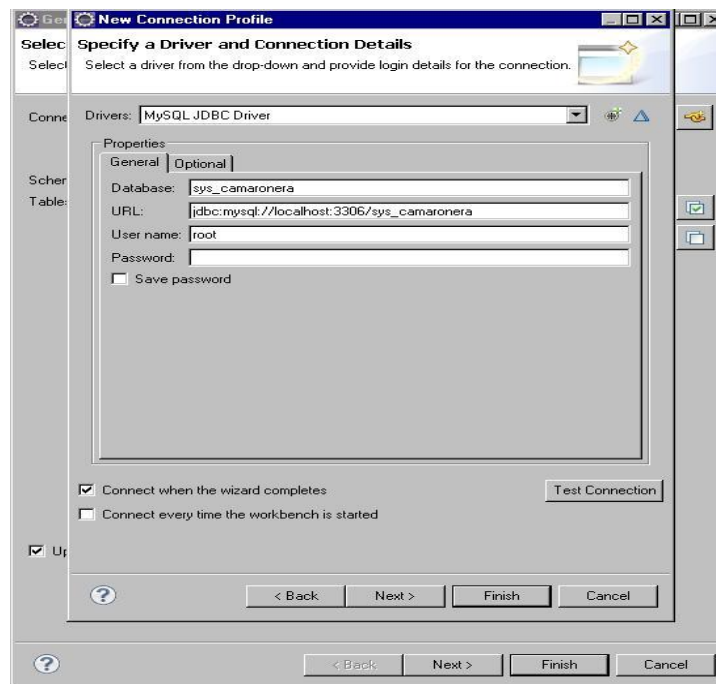


Figura. 2.3 Clave de Seguridad y host

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

A continuación se elije, la o las tablas con las que el sistema interactuará, para así crear los modelos.

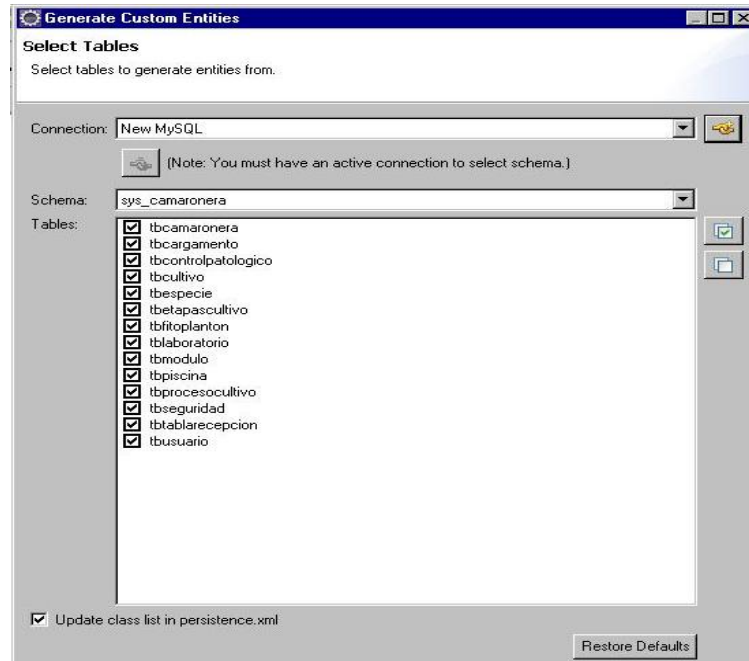


Figura. 2.4 Elección de Tablas

El siguiente paso, es empezar a agregar las relaciones de manera gráfica, ya que la herramienta lo permite. Presionando el botón con el símbolo + que se encuentra en extremo derecho superior tal como lo muestra la gráfica (Figura 2.5) siguiente.

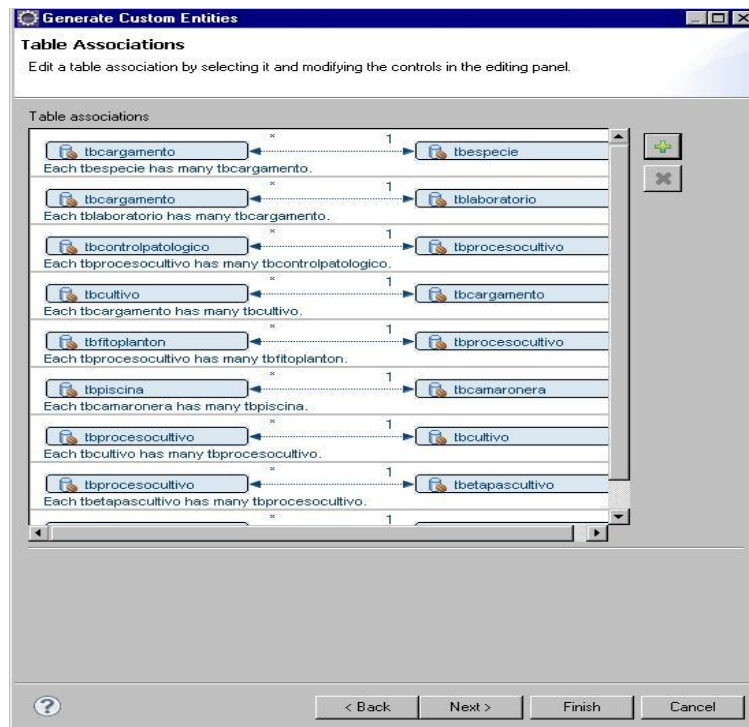


Figura. 2.5 Agregar relaciones entre las tablas

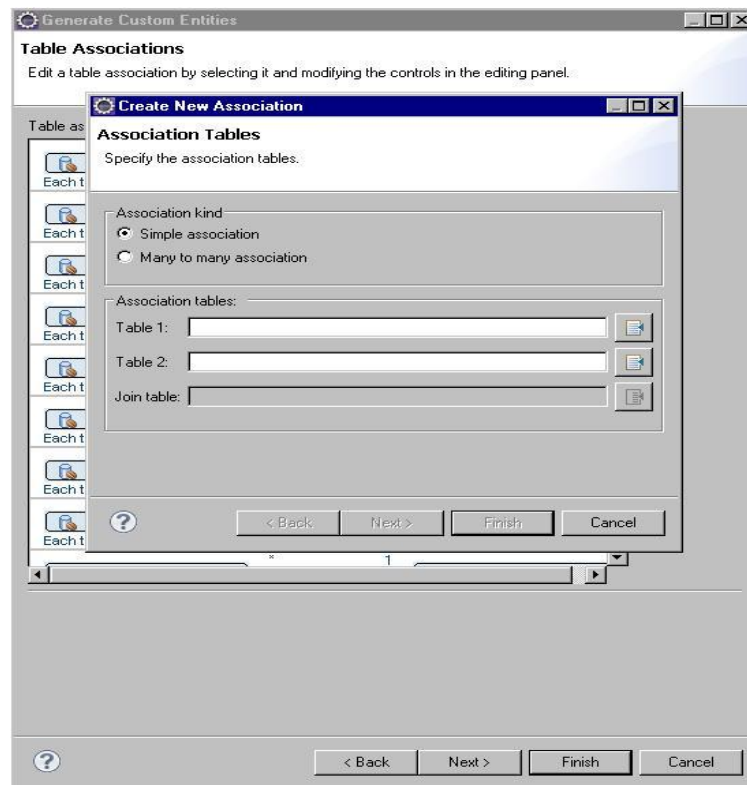


Figura. 2.6 Crear nuevas relaciones entre tablas

Es mediante, esta manera intuitiva y similar a como se haría en el administrador de base de datos (SQL SERVER) que se relacionan los campos uno a uno, buscando de esta forma que el Api Java Persistencia (JPA), cree las clases modelos con sus relaciones, facilitando el desarrollo y permitiendo concentrarse en el desarrollo de la aplicación.

Cuando se ha terminado, presionamos siguiente y nos permitirá darle un paquete a nuestras clases. De ahí siguiente y finalizar y se podrá utilizar las clases en el proyecto.

2.4 Componentes Vaadin

Lo fundamental de Vaadin es la biblioteca de Java que ha sido diseñada para la creación y mantenimiento de alta calidad con una interface web fácil. La idea importante en la programación modelo server-driven de Vaadin es que le permite olvidarse del web y programar como cualquier aplicación Java de escritorio con el toolkits convencional como serian AWT, Gire, o SWT. Pero más fácil.

Mientras el programar en web tradicional requiere de consumir mucho tiempo en el aprendizaje de las nuevas tecnologías web, si requerimos concentrarnos en la lógica de la aplicación, con el modelo de programación server-driven, Vaadin cuida de manejar la interfaz del usuario en el navegador y las comunicaciones de AJAX (Asynchronous JavaScript and XML) entre el navegador y el servidor. Con Vaadin no es necesario aprender y ponerse a punto en las tecnologías del navegador, como podrían ser por ejemplo HTML o JavaScript.

2.4.1 La Arquitectura general de Vaadin

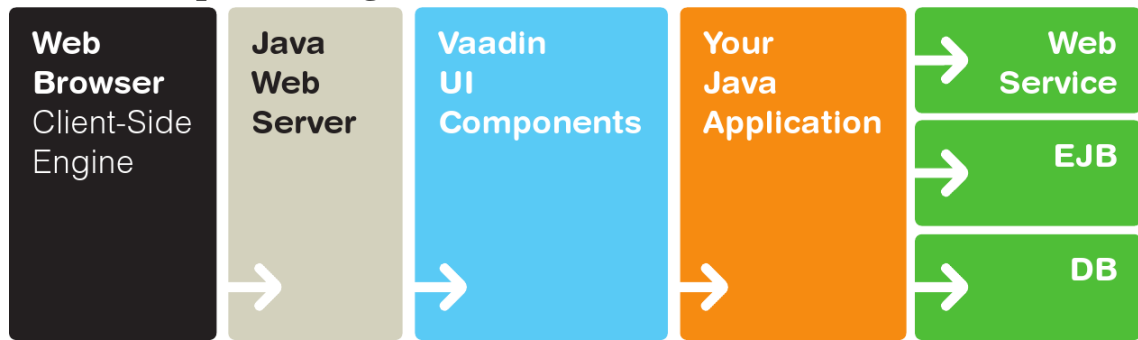


Figura. 2.7 Esquema de arquitectura Vaadin Tomado de Book of Vaadin (Vaadin Ltd. Marko Gronroos)

Vaadin consiste en un Framework del lado servidor y un artefacto del lado cliente que corre en el navegador como un programa de JavaScript, dando la interfaz de usuario y el usuario entregando la interacción al servidor (ver grafica 2.7 Esquema de Arquitectura Vaadin). Como la aplicación corre como una sesión de persistencia Java Servlet en una aplicación servidor, usted puede ligar su aplicación fácilmente a los datos y capa lógica de negocios.

Porque HTML, JavaScript, y otras tecnologías del navegador son esencialmente invisibles a la lógica de la aplicación, puede pensarse en el navegador web como sólo una plataforma de cliente delgada. Un cliente delgado despliega la interfaz del usuario y comunica los eventos del usuario al servidor a un nivel bajo. La lógica de mando de la interfaz de usuario corre en un servidor web basado en Java, junto con su lógica de negocios. Por contraste, una arquitectura del cliente-servidor normal con una aplicación del cliente especializada incluiría la porción de aplicación y las comunicaciones específicas entre el cliente y el servidor. Quitando esencialmente la capa de interfaz de usuario de la arquitectura de la aplicación hace a nuestra aplicación mucho más eficaz.

Cuando el Artefacto del lado Cliente se ejecuta como JavaScript en el navegador, ningún plugins del navegador es necesario para usar aplicaciones hechas con Vaadin. Esto le da una ventaja por encima de los Frameworks basado en Flash, Java Applets, u otro plugins. Vaadin cuenta con el apoyo de GWT para manejar un ancho rango de navegadores, para que el diseñador no necesite preocuparse por el apoyo del navegador.

Detrás del modelo de desarrollo server-driven, Vaadin hace un uso mejor de AJAX (Asíncrono JavaScript y XML) por medio de técnicas que hacen posible crear la Rich Internet Applications (RIA) que lo hacen tan sensible e interactivo como las aplicaciones de escritorio.

Por debajo, Vaadin usa GWT (Google Web Toolkit), para mostrar la interfaz de usuario en el navegador. Los programas de GWT son escritos en Java, pero compilados en JavaScript, librando al diseñador de aprender JavaScript y otras tecnologías del navegador. GWT es ideal para los componentes de interfaz de usuario avanzados (o

widgets en la terminología de GWT) e interacción de la lógica en el navegador, mientras Vaadin se ocupa de la lógica de la aplicación real en el servidor.

La biblioteca de Vaadin define una separación clara entre el usuario la presentación de la interfaz y la lógica y le permite desarrollar ellos separadamente como podemos apreciar en grafico (Fig. 2.7 Esquema de Arquitectura Vaadin). Nuestro acercamiento a estos temas se ven en la apariencia visual de las aplicaciones.

La arquitectura Vaadin consiste en un API de aplicación web, con una serie de componentes de interfaz de usuario, los temas para controlar la apariencia, y un modelo de datos que permiten a los componentes de interfaz de usuario la encuadración directamente a los datos. Por debajo emplea también un adaptador terminal para recibir las demandas de los navegadores web y contestaciones de la hechura dando las páginas.

Una aplicación usa las carreras (peticiones) de Vaadin como un servlet en un servidor Java web, sirviendo las demandas de HTTP. El adaptador terminal recibe las demandas del cliente a través del Java Servlet API del servidor web, e interpreta los eventos del usuario para una sesión particular. Un evento es asociado con un componente de UI y es entregado a la aplicación.

La cima de una aplicación de usuario consiste en una aplicación de clases que hereda com.vaadin.Application crea los componentes de UI (vea figura 2.8 Arquitectura Vaadin) que necesita, recibe los eventos con respecto a ellos.

2.4.2 Arquitectura Vaadin

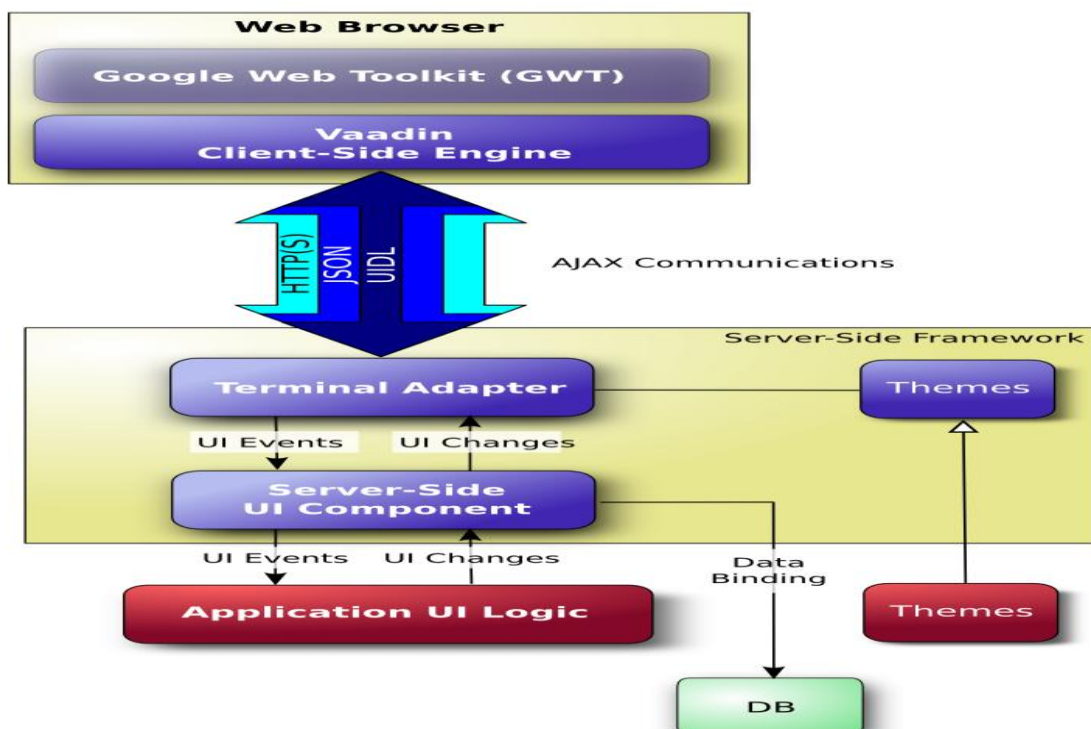


Figura. 2.8 Arquitectura Vaadin Tomado de Book of Vaadin (Vaadin Ltd. Marko Gronroos)

2.4.2.1 El apoyo para el Eclipse IDE

Mientras Vaadin no se liga a cualquier IDE específico, y se pudo haber concebido en cualquier IDE, se ha escogido el desarrollo de la aplicación sobre la cual tratará este documento a Eclipse IDE, que se ha vuelto el ambiente normal para el desarrollo de Java. El apoyo incluye:

El Vaadin Plugin para Eclipse nos permite:

- Crear los nuevos proyectos de Vaadin
- Crear los temas personalizados
- Crear los widgets del lado cliente personalizados y juegos del widget
- Crear los componentes compuestos personalizados con un diseñador visual
- Actualizar a una nueva versión de la biblioteca de Vaadin
- Importar el paquete de la instalación como un QuickStart, demostración de proyecto en el Eclipse

El Vaadin Plugin para el Eclipse es la manera recomendada de instalar Vaadin; la instalación real y el paquete contienen las demostraciones y documentación que también están disponibles en la website.

La mayor parte de la arquitectura y su función son como sigue:

2.4.2.2 Componentes de Interfaz de usuario: La interfaz del usuario consiste en componentes UI que son creados y puestos por la aplicación. Cada componente del lado del servidor tiene un colega del lado cliente, con qué el usuario actúa recíprocamente. Los componentes del lado servidor pueden realizarse sobre la conexión del cliente usando un adaptador terminal. Los componentes del lado cliente, a su vez, pueden realizar en serie la interacción del usuario volviendo a la aplicación, qué es recibida en los componentes del lado servidor como eventos. Los componentes relevan estos eventos y la lógica de la aplicación. La mayoría de los componentes se ligan a los datos fuentes.

2.4.2.3 El Artefacto del lado cliente: El Artefacto del lado Cliente de Vaadin maneja el navegador web que usa Google Web Toolkit (GWT). Comunica la interacción del usuario y los cambios UI con el Adaptador terminal del lado del servidor usando el lenguaje de definición de Interfaz de Usuario (UIDL), un lenguaje basado en JSON. Las comunicaciones de lenguaje son usando HTTP asíncronos o requerimientos HTTPS.

2.4.2.4 El Adaptador terminal: Los componentes de UI no se dan directamente como una página web, porque usan un Adaptador Terminal. Esta abstracción de esta

capa les permite a los usuarios usar las aplicaciones de Vaadin con prácticamente cualquier navegador web.

2.4.2.5 Los temas: La interfaz de usuario se separa entre la presentación y la lógica. Mientras que la lógica UI se ocupa del código Java, la presentación está definida en temas como CSS. Vaadin proporciona unos temas predefinidos. Los temas del usuario pueden, añadir un estilo a las hojas, incluida plantillas de HTML que definen los diseños personalizados y otros recursos, como las imágenes.

2.4.2.6 UIDL: El Adaptador Terminal dibuja la interfaz de usuario en la página web y cualquier cambio usando un especial lenguaje de definición de Interfaz de Usuario (UIDL). Las comunicaciones de UIDL se hace usando JSON (Objeto de JavaScript Notación) que es un ligero intercambio de datos en el formato, esto es especialmente eficaz para unir código basado en JavaScript con el código de AJAX en el navegador.

2.4.2.7 Eventos: La interacción de Usuario de eventos con los componentes de UI crea los eventos, qué se procesa primero en el lado cliente con JavaScript y entonces es pasado todo el camino a través del servidor de HTTP, adaptador terminal, y capas de componentes de usuario a la aplicación.

2.4.2.8 Modelo de datos: al utilizar el modelo de interfaz de usuario, Vaadin proporciona un modelo de datos para unir datos presentados en los componentes UI. Usando el modelo de datos, la interfaz de componentes de usuarios pueden poner al día los datos de la aplicación directamente, sin la necesidad de un código de control. Todos los componentes de UI usan este modelo de datos internamente, pero ellos pueden ligarse a una fuente de datos separada. Por ejemplo, usted puede ligar un componente de una tabla a una consulta SQL.

2.5 Tecnologías Subyacentes

Vaadin cuenta con el apoyo de tecnologías que permanecen ocultas al desarrollador, su framework crea la interfaz de usuario en el navegador web además de ocuparse de las tareas de bajo nivel que se dan producto de la interacción del usuario en el navegador. Para realizar toda esta interacción usuario, navegador, servidor utiliza AJAX, JSON y Google Web Toolkit de una forma totalmente invisible en el desarrollo de una aplicación.

2.6 Controles Vaadin

2.6.1 Managing Layout

Desarrolladores han querido siempre realizar GUI que se programen fácilmente. Las soluciones empezaron simples. Cuando GUIs aparecía en los escritorios de PC, prácticamente todas pantallas eran del tipo de VGA y fijo en 640x480 de tamaño. Mac o X Window System en UNIX no eran muy diferentes. Todos estaban contentos con tales

resoluciones de los gráficos que nunca pensaron que una aplicación tendría que trabajar en un tamaño de la pantalla sumamente diferente. En el peor de los casos, las pantallas sólo podrían crecer, fue lo que pensaron, dando más espacio para más ventanas. En los años ochenta, la idea de tener una pantalla de la computadora en su bolsillo simplemente no era realista. Los API de GUI permitieron poner componentes UI que usaban las coordenadas de la pantalla.

Algunos otros sistemas mantuvieron una manera fácil para el diseñador poder trabajar como lo era arrastrar y colocar componentes en una ventana fija clasificada según su tamaño. Se podría haber pensado que por lo menos los traductores se habrían quejado de la torpeza de tal solución, pero al parecer ellos no eran muy consientes de aquello.

Después que el web nació, el plan del diseño fue condenado para cambiar para siempre. Al principio, el diseño no fue muy importe, como todos estaba contento con los títulos llanos, párrafos, y unos hipervínculo aquí y allí. Diseñadores de HTML quisieron las páginas para correr en cualquier tamaño de la pantalla.

Esto significó que el plan de diseño tenía que ser fácil no sólo para programadores, sino también le permitiera al diseñador gráfico hacer su trabajo sin tener que saber mucho de programación.

Las aplicaciones reales, sin embargo, necesitan tener más mando, porque requieren poder cambiar el estado de componentes de interfaz de usuario e incluso su diseño a la carrera. Por aquello se creará la necesidad de separar la presentación del volumen en exactamente el nivel correcto.

Sun Microsystems había entrado en 1995 con una nueva programación utilizando el lenguaje, Java, para la creación de aplicaciones de escritorio. El usuario gráfico original de Java toolkit, utilizaba AWT (Windowing Toolkit Abstracto), que fue diseñado para trabajar en el múltiples sistemas operativos e incluido en los navegadores de web. Uno de los aspectos especiales de AWT era, el gerente de diseño, que permitió a los componentes de interfaz de usuario ser flexibles es decir, ir creciendo o encogiéndose según se deseara.

Se hizo posible incluso en el resize cambiar el tamaño de fuentes en pixeles, y el resto del diseño adaptado para el nuevo tamaño. La creación en el diseño Vaadin es una sucesión directa del concepto basado en web que tiene que ver con lo que es la separación entre satisfacción y apariencia que Java introdujo en aplicaciones con AWT las cuales buscaban ligar el diseño e interfaz de los componentes de usuario en sus programas. Vaadin diseño un framework que permiten posicionar los componentes UI en la pantalla en un modelo jerárquico, esto es del agrado de muchos desarrolladores que sienten preferencia por el toolkits de UI del Java convencional como AWT o SWT. Además, Vaadin permite acercarse al diseño web con los componentes de CustomLayout que pueden usarse para escribir su diseño como si se tratara de una plantilla en XHTML.

La moral de la historia es que, en Vaadin se piensa en aplicaciones web, y la apariencia es de mucha importancia. Las soluciones tienen que ser las mejores de ambos mundos (diseño/desarrollo) y satisfacer a profesionales involucrados en el código y diseño gráfico. En el lado del API, el diseño se controla por los componentes UI, particularmente, los componentes de diseño. En el lado visual, se controla por los temas. Los temas pueden contener cualquiera HTML, CSS, y JavaScript que usted o los desarrolladores de web crean como software.

Los componentes de interfaz de usuario en Vaadin pueden ser divididos en dos grupos: los componentes con que el usuario puede actuar recíprocamente y los componentes del diseño para poner los otros componentes a especificar en los lugares en donde el usuario los unen. Los componentes del diseño son idénticos en su propósito al manejador de diseño en los armazones de escritorio regulares para Java y usted podría usar Java básico para lograr el layouting de un componente sofisticado.

Se empezará creando un diseño de la raíz para la ventana principal, a menos que se esté usando el valor predeterminado, y entonces se agrega los otros componentes del diseño jerárquicamente, y finalmente los componentes de la interacción como las hojas del árbol de componentes.

2.6.2 Window y Diseño de Raíz de Tablero

La Ventana y sus superclass forman tableros que tienen un solo componente de diseño de raíz. El componente es normalmente un diseño, pero cualquier ComponentContainer se permite. Cuando usted crea los componentes, ellos crearán un diseño de raíz predefinido, normalmente VerticalLayout, pero este puede ser cambiado con el método setContent ()

El tamaño del diseño de la raíz es el tamaño predefinido del componente del diseño particular, por ejemplo, un VerticalLayout tiene 100% anchura y la altura indefinida por defecto.

2.6.3 VerticalLayout y HorizontalLayout

VerticalLayout y HorizontalLayout son los recipientes para poner los componentes verticalmente u horizontalmente, respectivamente. Éstos son los dos componentes del diseño más importantes en Vaadin y algunos componentes, como la Window y Panel, tienen un VerticalLayout como el diseño de la raíz que usted puede poner con el setContent().

En estos diseños, se ponen los subtítulos del componente sobre el componente:

A vertical stack of three text input fields. The top field is labeled 'Ciudad', the middle field is labeled 'Laboratorio', and the bottom field is labeled 'Responsable'.

Figura. 2.9 VerticalLayout

Usando HorizontalLayout da el diseño siguiente:

Three text input fields arranged horizontally. The left field is labeled 'Ciudad', the middle field is labeled 'Laboratorio', and the right field is labeled 'Responsable'.

Figura. 2.10 HorizontalLayout

2.6.4 Data Binding

Usted puede asociar un TextField a cualquier tipo de propiedad que permita conversión del valor de String con los métodos `getValue()` y `setValue()`.

```
//Tiene un modelo del datos inicial. Como Double es el inmodificable y
// no soporta la asignación String, el objeto es
// reconstruido en la envoltura cuando el valor se cambia.
Double trouble = 42.0;
// Envuélvalo en una fuente de datos de propiedad
```

```
final ObjectProperty<Double> property =
new ObjectProperty<Double>(trouble);
```

```
// Cree un campo del texto limitado a él
TextField tf = new TextField("The Answer", property);
tf.setImmediate(true);
// La muestra que el valor regresa escrito a
// la fuente de los datos cuando es revisó por el usuario.
Label feedback = new Label(property);
feedback.setCaption("The Value");
```

Cuando se colocan los valores en un Formulario o Table, el `DefaultFieldFactory` casi crea un `TextField` para cada tipo de propiedad por defecto. Sin embargo, si se necesitara crear una forma personalizada en donde se tomaran aspectos tales como la descripción de campo de texto, la proximidad, la aprobación, el formato, y así sucesivamente.

2.6.5 La Fecha y Time Input con DateField

El componente de `DateField` proporciona los medios para desplegar la fecha de la entrada y tiempo. El campo entra en dos variaciones: `PopupDateField`, con una caja de la entrada numérica y una vista de calendario de popup, e `InlineDateField`, con la vista

del calendario siempre visible. El DateField de los valores predeterminados de la clase están sujetos bajo la variación del popup.

El ejemplo debajo ilustra el uso del baseclass de DateField al que es equivalente el PopupDateField. Se ha puesto el tiempo inicial del campo de la fecha al tiempo actual usando el valor predeterminado constructor de la clase de java.util.Date.

```
// Crear un DateField con el estilo por default
DateField date = new DateField();
// toma un dato y tiempo presente
date.setValue(new Date());
```

2.6.5.1 DateField (PopupDateField) para Seleccionar Datos y Tiempo

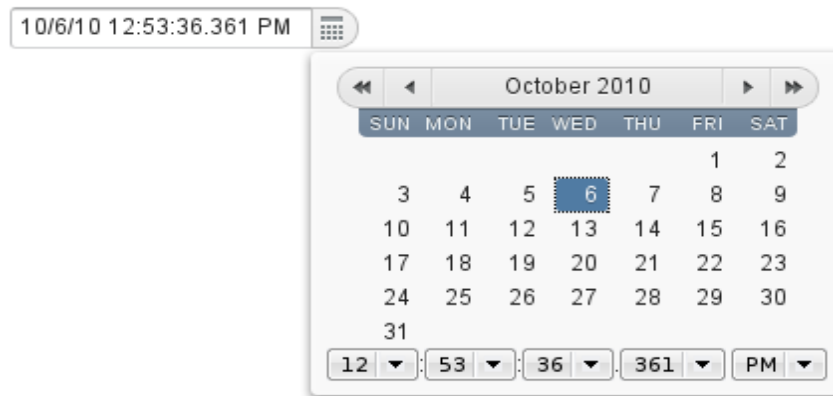


Figura. 2.11 PopupDateField

El PopupDateField proporciona que la fecha entró usando una caja del texto donde se muestra la fecha y tiempo escogidos tal como lo indica la figura 2.11. Con el DateField los valores predeterminados a este componente y su uso son exactamente igual que el descrito antes. Pulsando el botón derecho de la fecha se abre una vista del popup para seleccionar el año, mes, y día, así como tiempo. También al bajar la llave abre el popup. Una vez abierto, el usuario puede navegar en el calendario que usa las llaves del cursor.

Se desplegaran la fecha y tiempo seleccionados del popup en la caja del texto según el valor predeterminado. La fecha y formato de tiempo del sitio actual, o como se halla especificado con el setDateFormat(). En el mismo se usan las definiciones del formato que sean requeridas por parte del usuario.

2.6.6 Button

El Botón es un componente de interfaz de usuario que normalmente se usa por finalizar la entrada y comenzar alguna acción. Cuando el usuario pulsa el botón, un Button.ClickEvent se emite. Un oyente que hereda la interfaz de Button.ClickListener puede manejarse haciendo clic con el método buttonClick ().

```
public class TheButton extends CustomComponent
implements Button.ClickListener {
Button thebutton;
public TheButton() {
// Cree un Botón con un mensaje escrito.
thebutton = new Button ("Este es un Boton");
// Escucha por ClickEvents.
thebutton.addListener(this);
setCompositionRoot(thebutton);
}
/** Los eventos de click para el botón. */
public void buttonClick (Button.ClickEvent event) {
thebutton.setCaption ("Este es un Botón");
}
}
```

Ejemplo of a Button:

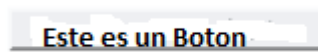


Figura. 2.12 Button

Como una interfaz del usuario a menudo tiene varios botones, usted o puede diferenciar entre ellos comparando la referencia de objeto de Botón devuelta por el método `getButton()` de `Button.Click` - Evento a una referencia guardada o usando un método del oyente separado para cada botón.

2.6.7 Table

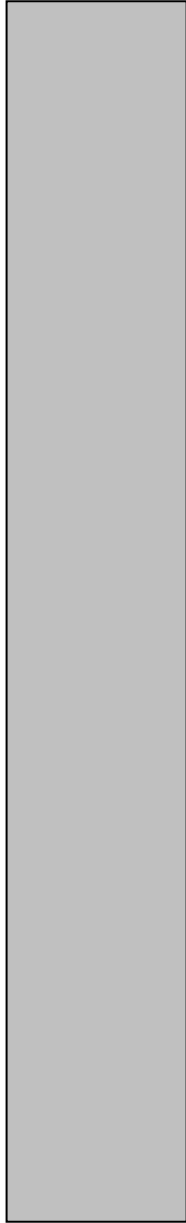
Los componentes de la Tabla han sido pensados para presentar datos en una forma tabular y organizados en las filas y columnas. La Table es uno de los componentes más versátiles en Vaadin. Las células de la Table pueden incluir el texto arbitrario gracias a los componentes de UI. Se pueden llevar a cabo corrección de los datos de la tabla fácilmente, mientras haciendo clic por ejemplo en una célula podría cambiarlo a un campo del texto por revisar.

El datos de contenido en una Table están manejados por el modelo de datos de Vaadin, a través de la interfaz del recipiente del Table. Esto lo hace posible ligar una Table directamente a una fuente de datos, que podría ser una base de datos (MySQL). Sólo la parte visible de la Table está cargada en el navegador y moviendo la ventana visible con el scrollbar del servidor. Mientras el dato está cargado, un tooltip se desplegará para mostrar el rango y el número total de artículos en la Table.

Las filas de la Table son los artículos en el recipiente y las columnas son las propiedades. Cada fila de la Table (el artículo) se identifica con un identificador del artículo (IID), y cada columna (la propiedad) con un identificador de propiedad (PID).

Al crear una Table, usted necesita definir las columnas con el `addContainerProperty()` primero. Este método presenta dos opciones. La más simple toma la propiedad ID de la columna y también el subtítulo de la columna. El más complejo permite diferir PID y título para la columna. Por ejemplo, esto puede permitir una internacionalización de

títulos de la Table más fácil, porque si un PID se internacionaliza, la internacionalización tiene que ser usada en todas las partes donde el PID sea usado. En un formulario complejo existen métodos que también permite definir un icono para la columna como un recurso.



CAPÍTULO 3
ALCANCE DEL
PRODUCTO

3. Alcance del Producto:

3.1 Introducción

En su etapa inicial de desarrollo contará con los tres más importantes análisis que se realizan en el laboratorio de una camaronera que son:

1. Análisis de Fitoplancton en cual el usuario será capaz de guardar la información de los análisis semanales que tienen que ver con la cantidad estimativa de la producción fitoplancton en las piscinas, el cual por el manejo semi-intensivo en el que se desenvuelve la explotación en las piscina es primordial contar con una herramienta que nos permita conocer de una manera rápida y sistematizada.

Así también los datos podrán ser reportados a la conveniencia del usuario responsable si este quisiera conocer pormenores de la producción de algas (fitoplancton) y cuál ha sido el porcentaje en que estas se encuentran en un análisis determinado.

2. Análisis de control Patológico el cual busca además del ingreso, actualización y almacenamiento de la información que busca reportar información de aspectos tales peso, longitud de la muestra, fecha en que es realizado análisis, además de un análisis intestinal buscando parásitos que incidan en salud y por consecuencia en su producción, de la misma manera que el usuario responsable podrá realizar consultas de algún análisis patológico específico y de su reporte además.
3. Análisis Recepción de Larva tendrá la capacidad de realizar el ingreso, actualización y almacenamiento de la información proveniente de el análisis primario que realiza el usuario responsable al llegar la larva para dar inicio con el proceso de cultivo del camarón y procesando esta información con el fin de reportar un porcentaje de supervivencia y mortalidad de este primario análisis realizado a una muestra tomada de forma aleatoria de entre los embarques de larvas provenientes de los distintos laboratorios de procedencia.

3.2 Opciones del sistema

3.3 Opción Mantenimiento del menú principal: Cuenta con las opciones Camaronera, Piscina, Etapas Cultivo, Tipo Especie y Laboratorio.

3.3.1 Camaronera: Esta opción permitirá guardar la información concerniente de la camaronera, en campos como: ciudad, dirección, nombre de la camaronera y teléfono.

Campo	Descripción
Ciudad	Se ingresará en este campo el nombre de la ciudad donde reside la camaronera.
Dirección	Registrará la dirección o localidad donde se encuentra situada la camaronera.

Nombre	Indica que se registrará el nombre de la camaronera.
Teléfono	En este campo se ingresará el teléfono de la camaronera

Tabla 3.1 Descripción de Opciones Camaronera.

3.3.2 Piscina: En esta opción el usuario podrá ingresar los datos de información de la piscina en donde se realizará físicamente la explotación acuícola, está compuesta por un formulario con los siguientes campos:

Campo	Descripción
Elija una camaronera	El cual está conformado por ComboBox en donde el usuario puede seleccionar la camaronera a la pertenecerá dicha piscina.
Volumen	Viene dado por largo (L), ancho (A) y profundidad (P) de la piscina $V=L*A*P$ medida representada en metros cúbicos (m ³)
Área	Dada por el largo (L) y ancho (A) de la piscina $A = L*A$ medida representada en Hectáreas (ha).
Profundidad	Esta dada por la cantidad de metros tomados desde el borde de la piscina hasta el interior, en su parte más profunda.
Sector	Medida representada por la división física de la extensión de la camaronera en dos partes sur y norte.

Tabla 2.2 Descripción de Opciones Piscina.

3.3.3 Etapas Cultivo: opción del menú Mantenimiento que nos conduce al Formulario Etapas cuya finalidad es registrar en que etapa se encuentra la explotación. Entendiéndose que estas podrían ser: cuarentena, muda, transferencia, cosecha entre los más importantes.

3.3.4 Tipo de especie: esta opción nos conduce al Formulario de Especies el cual consta con el campo Descripción en cual podría el usuario hacer una descripción científica de la especie a explotar, entendiéndose que podría ser no solo camarón sino cualquiera que quisiese emprenderse.

3.3.5 Laboratorio: la opción Laboratorio nos conduce al Formulario Laboratorio el cual tiene como objetivo el ingreso de la información del laboratorio del cual procede la larva a utilizar en la explotación, cuenta con los siguientes campos:

Campo	Descripción
Fecha	Campo que indica en qué fecha fue receptada la larva procedente del laboratorio.

Laboratorio	En este campo ira introducido el nombre del laboratorio del cual procede la larva a cultivar.
Responsable	Indica el nombre del técnico encargado de recibir la larva y hacer su respectivo análisis de verificación.

Tabla 3.3 Descripción de opciones Laboratorio

Cada uno de los formularios de Mantenimiento cuenta con un menú con las siguientes opciones:

Opciones	Descripción
Nuevo	Crea un nuevo registro listo para ser llenado por el usuario (Técnico Responsable)
Guardar	Guarda la información en la base de datos de los datos ingresados por el usuario.
Consultar	Consulta de una tabla los datos previamente almacenados dando clic en el registro seleccionado, los datos serán mostrados en sus respectivos campos en espera de algún requerimiento del usuario tal como modificar o eliminar.
Eliminar	Realizará la eliminación física en la base de datos del registro seleccionado.
Cerrar	Por medio de esta opción el usuario cierra el formulario activado y vuelve al menú principal.

Tabla 3.4 Descripción de opciones generales del menú Mantenimiento.

3.4 Opción Cultivo del Menú Principal

La opción Cultivo tiene por objetivo la sistematización y registro de información del proceso de cultivo, entendiéndose por proceso a todas las fases que se dan desde el mismo momento que llega el cargamento con la larva de los distintos laboratorios (desembarque) hasta que el camarón ya se encuentra listo para su extracción de la piscina para ser llevado a la empacadora para ser exportado.

Cultivo en el menú principal se encuentra compuesto por las siguientes opciones:

3.4.1 Cultivo: tiene por finalidad mediante el Formulario Apertura de Cultivo registrar el inicio mismo de nuestra explotación acuícola consta de los siguientes campos:

Campo	Descripción
Elija un cargamento	Mediante el cual el usuario (técnico responsable) puede registrar y conocer que la larva recibida de un determinado laboratorio pertenece a un determinado cargamento.
Fecha Ingreso	Mediante este campo el usuario ingresa la fecha de recepción de la larva
Fecha Fin	Mediante este campo el usuario registra la fecha de cuando se dio por concluido el desembarque de un cargamento específico de larvas, esto es porque en ocasiones un cargamento puede ser grande y requerirá de más de 24 horas en ser cubierto debido básicamente a problemas tales como mareas, o mal clima que hacen que las embarcaciones en la cual se envía la larva, tengan que hacer varios embarques para cubrir un cargamento.
Técnico responsable	Este campo registrará el nombre del técnico responsable que recibió la larva en de un determinado cargamento.
Costo promedio Cultivo	Registrará el valor promedio del cargamento recibido para la explotación en las piscinas.

Tabla 3.5 Descripción de opciones Cultivo.

3.4.2 Opción Registro de Cultivo

Tiene por finalidad de llevar un registro sistematizado a través del Formulario Proceso de Cultivo de las etapas por la que va pasando una determinada selección de población conformada por las larvas que llegaron en un determinado cargamento con fin de conocer su evolución tanto fisiológica como reproductiva una vez que alcance su estado de madurez sexual. Posee los siguientes campos:

Campo	Descripción
Elija una Etapa	Este campo sirve para que el usuario, ingrese la etapa de desarrollo en la que se encuentra la población escogida.
Elija una Piscina	Indica en cual piscina se encuentra la población a explotar.
Elija un Cultivo	Se indica la clave por la cual se conocerá a una determinada población a explotar

	conformada por las larvas de un determinado cargamento.
Fecha de Recepción	Indica cual fue la fecha en que se llevo a cabo el desembarque o la transferencia a otra piscina para continuar la explotación.
Nivel	Se halla dado por la profundidad de la piscina.
Otros	Campo en el cual el usuario podrá realizar la anotación de cómo ha encontrado la explotación hasta esa fecha.
Población recibida	Busca registrar la cantidad aproximada de larvas a explotar en una determinada piscina.

Tabla 3.6 Descripción de opciones Registro de Cultivo.

Tanto los formularios Cultivo como Registro de Cultivo cuentan con un menú con las siguientes opciones descritas en la siguiente **Tabla 4**

3.5 Opción Transacción del Menú Principal

Esta opción del menú principal tiene por objetivo ocuparse del proceso de análisis y reportes que el usuario tendrá a disposición para facilitar su trabajo en el laboratorio. Dichos análisis son los más importantes que se hacen periódicamente buscando conocer cuál es el estado de salud del camarón así como su desarrollo semana tras semana.

Entre las opciones que maneja transacción están:

3.5.1 Cargamento: mediante esta opción nos muestra el formulario de cargamento en el cual se registrara información del lugar de procedencia de larva, su especie, y cantidad.

Consta de los siguientes campos:

Campos	Descripción
Elija un Laboratorio	Este campo registrará el lugar de procedencia de la larva.
Elija una Especie	El usuario podrá ingresar la especie a la cual pertenece larva a cosechar (explotar).
Población	Se registrará el número de individuos en una cifra estimativa.
Número de Gavetas	En este campo se ingresará el número de embases en donde vendrá alojada la larva convenientemente aclimatada.

Tabla 3.7 Descripción de opciones Cargamento.

3.5.2 Control Patológico: el control patológico busca registrar el estado fisiológico del camarón mediante un análisis de control patológico el cual es realizado en el laboratorio de una camaronera por el técnico responsable a una muestra aleatoria de doce individuos diseccionándolos en busca de conocer aspectos como: peso, estado de branquias, intestinos y hepatopáncreas.

Estos parámetros encontrados en el análisis serán registrados en el aplicativo mediante el formulario de control patológico que consta de los siguientes campos:

Campo	Descripción
Elija un Cultivo	Registrará el cultivo al que se le realizará el análisis seleccionándolo de un ComboBook.
Peso promedio	En este campo se registrará el peso promedio tomados a los 10 individuos seleccionados para la muestra.
Aspecto General	Descripción física de cómo se encuentra (la muestra) en el momento de hacer el análisis buscando ver si no presenta manchas, o alguna otra anomalía en su anatomía externa.
Observación	Mediante este campo el usuario podrá anotar alguna descripción que él considere oportuna y relevante en relación al Análisis de control Patológico.
Fecha	En este campo se anotará la fecha en que fue realizado el análisis.
Branquias Detritus	Se registrará en este campo la cantidad de Detritus (microorganismos) que se encuentran en las branquias en el momento del análisis a microscopio a 10 individuos, dato en donde se buscara una cantidad porcentual que represente la cantidad de Detritus encontrada.
Branquias Ectoparásitos:	Registrará la cantidad de ectoparásitos que se encuentran en el análisis microscópico de las branquias de la muestra escogida, valor anotado en porcentaje.
Branquias Filamentosas	Se registrará la cantidad de filamentosas (microorganismos) detectadas en el microscopio durante el análisis de las branquias realizadas a 10 individuos (camarones).
Branquias Fusarium	Registrará la cantidad de Fusarium (microorganismos) detectados en el análisis microscópico realizado en 10 muestras (camarones).

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

Branquias Algas	En este campo será anotado la presencia de algas detectadas en el análisis microscópico hecho a 10 muestras durante el análisis de control patológico el dato será una representación porcentual de el número de algas detectadas en la muestra escogida para analizar.
Otros	En este campo se registrará otros tipos de parásitos o microorganismo encontrados durante el análisis microscópico a las muestras conformada por 10 individuos.
Hepatopáncreas Incidencia Baculovirus	Registrará la información obtenida durante el análisis de control patológico realizado en los hepatopáncreas de los 10 individuos bajo el microscopio buscando la presencia del Baculovirus (microorganismos) que infectan al camarón llegando a producir una mortandad masiva en la producción si no es detectada a tiempo. De ahí su importancia.
Hepatopáncreas Lípidos	Se Registrará la presencia de Lípidos (microorganismo) detectados durante el análisis microscópico. Una abundancia de estos indicará un estado de salud optimo en la muestra seleccionada la deficiencia en cambio será motivo para prestar atención en el manejo alimenticio de la muestra seleccionada en los 10 individuos.
Hepatopáncreas Túbulos	Registrará la presencia de los Túbulos (microorganismos) encontrados durante el análisis microscópico buscando evaluar su aspecto el mismo que será representado porcentualmente de acuerdo a parámetros manejados por el técnico responsable del análisis. El valor estimativo que encuentre el técnico será anotado en el campo de la aplicación.
Intestino Gregarina Adulta	Se anotará el número de gregarina adulta (microorganismo) encontradas en el análisis microscópico a los 10 individuos en la muestra aleatoria durante el análisis de control patológico.
Intestino Gregarina Inmadura	Se anotará el número de gregarina Inmadura (microorganismo) encontradas en el análisis realizado en el microscopio

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

	a los 10 individuos que se seleccionan en la muestra.
Intestino Incidencia Baculovirus	Se registrará en este campo, el valor correspondiente producto del análisis de los intestinos de los 10 individuos tomados como muestra para ser examinados mediante microscopio, buscando encontrar la presencia del Baculovirus, en dichos análisis, que de no detectarse de manera oportuna sería el responsable de una infección masiva, en la piscina donde haya sido sacada la muestra, causando una mortandad que podría expandirse en toda la explotación. La información será una representación porcentual representativa de los intestinos de los 10 individuos analizados.
Intestino algas diatomea	Registrará el número de diatomeas (algas) encontradas en el análisis microscópico a los intestinos de los 10 individuos tomados para el análisis. El dato ingresado en el campo de la aplicación corresponderá a la representación porcentual de algas de esa especie del gran total (100%) de algas encontradas en el análisis.
Intestino algas Cianofitas	Registrará el número de cianofitas (algas) encontradas en el análisis microscópico a los intestinos de los 10 individuos tomados para el análisis. El dato ingresado en el campo de la aplicación corresponderá a la representación porcentual de algas de esa especie del gran total (100%) de algas encontradas en el análisis.
Intestino Vacío	En este campo se registrará el dato correspondiente a el examen de no encontrar elemento alguno (alimento) en los intestinos de las muestra escogidas para el análisis en el microscopio.
Intestino Lleno	En este campo se registrará el dato correspondiente a el examen de haber encontrado completamente lleno (alimento) los intestinos de las muestras escogidas para el análisis en el microscopio.

Intestino Semilleno	En este campo se registrará el dato correspondiente a el examen de haber encontrado a medio llenar (alimento) los intestinos de las muestras escogidas para el análisis en el microscopio.
----------------------------	--

Tabla 3.8 Descripción de opciones Control Patológico.

3.5.3 Control Recepción de Larva: mediante esta opción podrá llevarse a cabo la sistematización en el manejo de la información del proceso (análisis de laboratorio) que se da en la recepción de la larva proveniente de los laboratorios mediante el formulario de recepción el cual consta de los siguientes campos:

Campos	Descripción
Elija cargamento	En este campo el usuario podrá seleccionar de un ComboBox el cargamento al cual pertenece el muestreo aleatorio de la selección de individuos (larvas) que se ha tomado para el análisis con el fin de encontrar el estado en que han llegado del laboratorio de procedencia.
Valor muestra	Se registrará el número de individuos escogidos para realizar la prueba de aclimatación de la larva.
Salinidad	Registrará el valor tomado por el salinometro el cual mide la salinidad del agua de donde se saco la muestra.
PH	En este campo se registrará el valor arrojado por el peachimetro el cual mide el PH (potencial de hidrógeno) del agua de la muestra que fue escogida para analizar.
Fecha	Registrará la fecha en que fue tomada la muestra para ser sometida a el análisis de recepción de larva.
Valor mortandad	Se Registrará el número de individuos muertos (larvas) durante la prueba de aclimatación.
% de Mortandad	En este campo el aplicativo procesará la información de mortandad y supervivencia para luego realizar un cálculo (dependiendo del técnico) y arrojará un valor automáticamente.
Valor Supervivencia	Se registrará el número de individuos vivos (larvas) que fueron capaces de sobrevivir a la prueba de aclimatación.
% de Supervivencia	En este campo el aplicativo procesará la información de mortandad y

	supervivencia para luego realizar un cálculo (dependiendo del técnico) y arrojará un valor automáticamente.
Necrosis	En este campo se anotará en una cantidad porcentual la apreciación del técnico encargado de realizar el análisis de cuanto tejido muerto encontrará en las larvas escogidas para el muestreo.
Baculovirus	Se registrará en este campo el valor de la apreciación que realice el técnico encargado la cual se hará en una cantidad estimativa dependiendo del número de individuos escogidos para el análisis.
Elija donde envía	Registrará en este campo mediante la selección en un comboBox el lugar donde será enviada la larva del cargamento del cual se seleccionaron las muestras para analizar.

Tabla 3.9 Descripción de opciones Control de Recepción de Larva.

3.5.4 Control Fitoplancton: mediante esta opción el aplicativo estará en la capacidad de sistematizar el análisis de control de fitoplancton mediante el formulario de Fitoplancton el cual tendrá por objetivo reportar la cantidad de algas que son encontradas al hacer una muestra del agua en donde se halla delectado algún problema, por parte del técnico o por simple chequeo diario de la misma.

Entre los campos que se encuentran en formulario Fitoplancton están los siguientes:

Campos	Descripción
Elija un cultivo	En este campo se registrará el cultivo al que pertenece el análisis de fitoplancton a realizar mediante la selección por parte del usuario en un ComboBox donde estarán los registros de todos los cultivos que se encuentren hasta la actualidad.
Responsable	Se registrará en este campo el nombre del técnico responsable encargado de realizar el análisis.
Fecha de creación	En este campo se registrará la fecha en que fue realizado el análisis de control de fitoplancton.
Muestra	Se registrará en este campo el número de la piscina o pre criadero en donde será extraída la muestra de agua para el respectivo análisis.
# Total de células	Registrará en este campo el número total de células encontradas durante el análisis a microscopio de la muestra de agua examinada.

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

Valor diatomea	En este campo se anotará el valor del total de algas diatomeas encontrado durante al análisis de microscopio realizado a la muestra.
Diatomea	Se registrará en este campo la cantidad que procese la aplicación una vez que el usuario termine de ingresar los datos en los campos % Cianofitas y % Clorofita para poder realizar el cálculo respectivo, resultado que representará su cantidad porcentual (número diatomeas) del número total de células que equivaldría al 100%.
Valor Cianofitas	En este campo se anotará el número de algas Cianofitas encontrados durante el análisis a microscopio de la muestra de agua escogida para el análisis.
% Cianofitas	Se registrará en este campo la cantidad que procese la aplicación una vez que el usuario termine de ingresar los datos en los campos % Diatomea y % Clorofita para poder realizar el cálculo respectivo, resultado que representará su cantidad porcentual (número cianofitas) del número total de células que equivaldría el 100%.
Valor Clorofitas	En este campo se anotará el número de algas Clorofitas encontrados durante el análisis a microscopio de la muestra de agua escogida para el análisis.
% Clorofitas	Se registrará en este campo la cantidad que procese la aplicación una vez que el usuario termine de ingresar los datos en los campos % Diatomea y % Clorofita para poder realizar el cálculo respectivo, resultado que representará su cantidad porcentual (número clorofitas) del número total de células que equivaldría el 100%.
Valor PH	Se anotará en este campo el valor registrado por el peachimetro instrumento que mide el PH (potencial de hidrógeno). Cantidad que mide la alcalinidad o acidez de la muestra de agua a analizar.
Salinidad	El dato registrado en este campo será el que nos proporciones el salinometro que es el instrumento que mide la salinidad del agua de la muestra que será analizada.
Valor de Ciliados	El valor registrado en este campo será el valor de los microorganismos ciliados (algas) encontrados durante el análisis en el microscopio de la muestra tomada para

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

	el análisis.
Conteo por mililitro	Cantidad registrada durante el conteo de algas mediante la utilización de una placa Neubauer en el microscopio, durante el análisis a la muestra de agua seleccionada.
Observación	Se anotará en este campo cualquier apreciación técnica que considere relevante por parte del usuario detectada durante el análisis.

Tabla 3.10 Descripción de opciones Control Fitoplancton.

Los formularios pertenecientes a Transacción en el menú principal cuentan con el menú de opciones expuestos en la **Tabla 3.4 Opciones generales del menú Mantenimiento**



CAPÍTULO 4

**DISEÑO DEL SISTEMA
(REQUERIMIENTOS, MODELO DE
OBJETOS, CASOS DE USOS
IMPORTANTES, DIAGRAMAS DE
IMPLEMENTACIÓN)**

4. Diseño del Sistema (Requerimientos, Modelo de Objetos, Casos de Usos Importantes, Diagrama de Implementación)

4.1 Requerimientos

4.1.1 Introducción

Clasificación de los requerimientos del proyecto, los cuales se identifican con una clave, a la cual se le hará referencia en los diagramas más detallados en adelante. Esta clasificación se hace en tres grandes grupos: Consultas e Informes, Almacenamiento y Procesamiento.

Numero de Requerimientos	Descripción
Consulta/Informes	
R1	Consulta e informes de Control Patológico
R2	Consulta e informes de Análisis de Fitoplancton
R3	Consulta e informes de Recepción de Larva
Almacenamiento	
R4	Datos de Control Patológico
R5	Datos de Análisis de Fitoplancton
R6	Datos de Recepción de Larva
Procesamiento	
R7	Porcentaje numérico de Diatomeas
R8	Porcentaje numérico de Cianofitas
R9	Porcentaje numérico de Clorofitas
R10	Porcentaje numérico de Supervivencia de larvas

Tabla 4.1 Requerimientos.

4.2 Casos de Usos

4.2.1 Introducción

Este diagrama representa la funcionalidad completa de una sistema (o una clase), mostrando su interacción con los agentes externos. Esta representación se hace a través de las relaciones entre los actores (agentes externos) y los casos de uso (acciones) dentro del sistema. Los diagramas de Casos de Uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos que tiene a su cargo. Esos conjuntos de funcionalidades son representados por los casos de uso. Se pueden visualizar como las funciones más importantes que la aplicación puede realizar o como las opciones presentes en el menú de la aplicación.

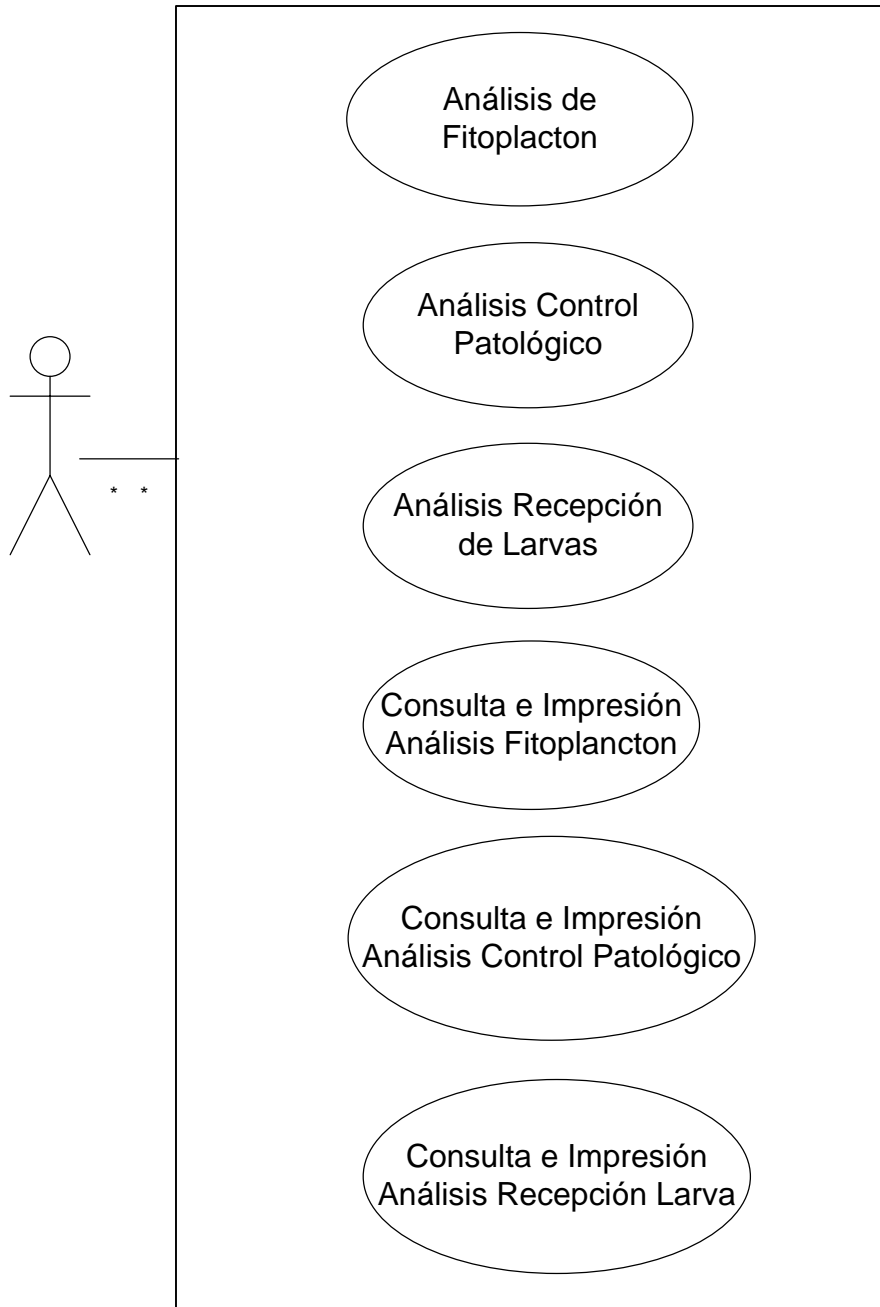


Figura. 4.1 Diagrama casos de usos

4.3 Diagrama Subcasos de Uso

4.3.1 Introducción

Hacen referencia a la descomposición de los DCU (Diagrama de Casos de Usos) del punto anterior. Se dan cuando existe una relación entre dos casos de uso. Dicha relación puede ser de extensión, que en términos de OO, esta relación es una relación de

herencia, donde el “subcaso” especializa al caso. También puede ser una relación de “uso”, donde el caso requiere que el subcaso se realice completamente para que él mismo se realice bien y completamente.

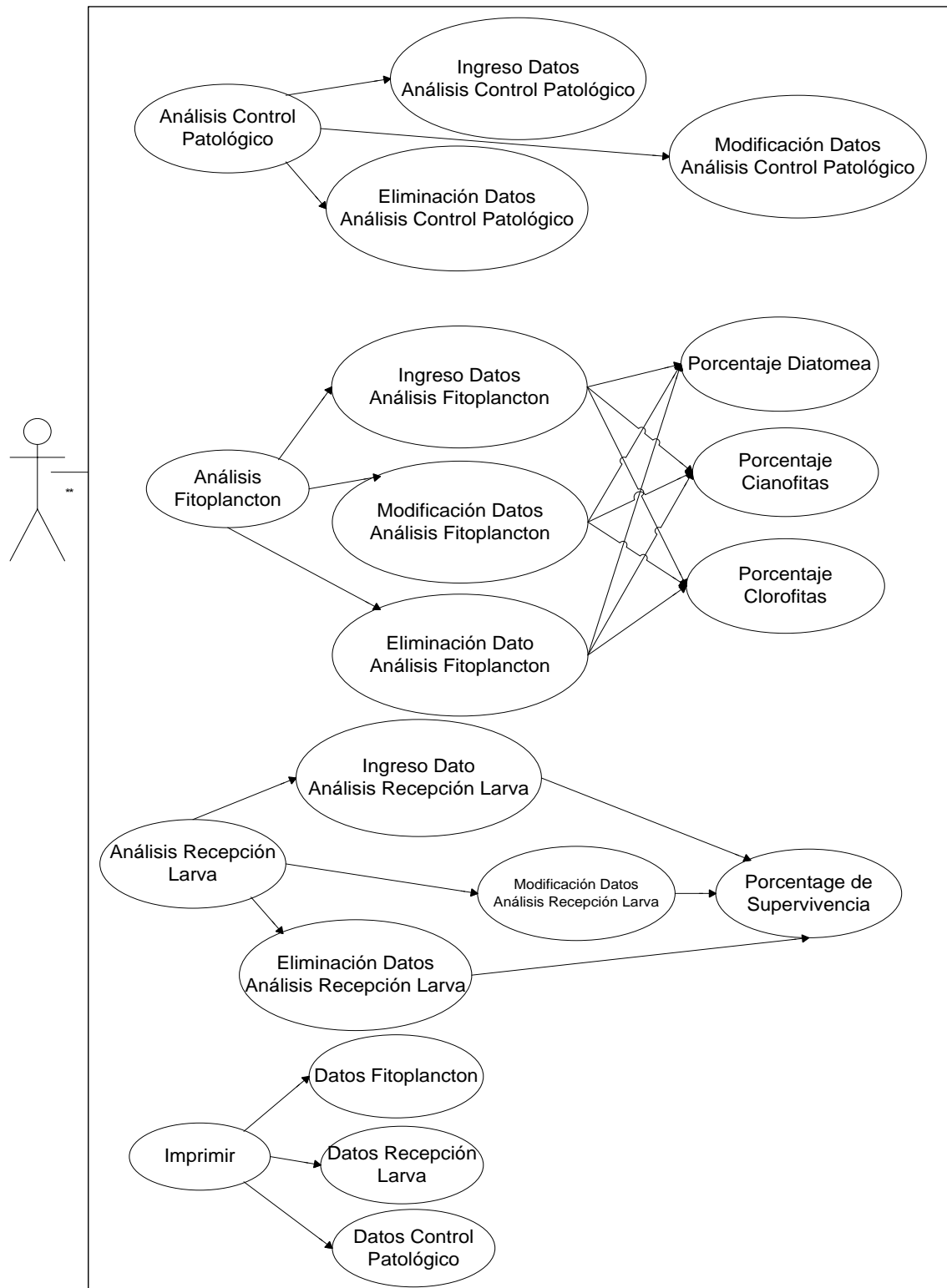


Figura. 4.2 Diagrama Subcasos de usos

4.4 Descripción de Casos de usos

4.4.1 Introducción

Este formato muestra una descripción para ayudar a comprender los Casos y SubCasos de Uso. También hace referencia a los requerimientos consignados en el documento de requerimientos, con los cuales tiene relación. Por causa del espacio, solo se muestran algunos a continuación.

Control de Análisis de Laboratorio Acuicultor

Nombre	Análisis de Fitoplancton
Alias	
Actores	Responsable
Función	Permitir mantenimiento de Registros de Fitoplancton
Descripción	El responsable puede registrar el control diario de los análisis de fitoplancton, modificación de dichos registros y eliminación de los mismos.
Referencias	De requerimientos: R2, R5, R7, R8, R9. De casos: Consulta e Informes Análisis de Fitoplancton, Porcentaje numérico de Diatomeas, Porcentaje numérico de Cianofitas, Porcentaje numérico de Clorofitas.

Tabla 4.2 Caso de uso control Análisis de Fitoplancton.

Nombre	Control Patológico
Alias	
Actores	Responsable
Función	Permitir el mantenimiento registros de Control de Patológico
Descripción	El responsable puede registrar el Control diario de Patologías del camarón, podrá además modificar dicho datos de Control Patológico y eliminarlos de así deseárselo.
Referencias	De requerimientos: R1, R4 De casos: Consultas e informes de Control Patológico.

Tabla 4.3 Caso de uso control Patológico.

Nombre	Análisis de Recepción de Larva
Alias	
Actores	Responsable
Función	Permitir el mantenimiento de Registro de Recepción de larva, que llega de diferentes laboratorios, en los cargamentos asignados, como paso primordial en el inicio de una nueva cosecha.
Descripción	El responsable puede registrar el control de recepción de larva de camarón, además de hacer cualquier modificación requerida en los datos de dichos registros, así como también su eliminación de ser requerido.
Referencias	De requerimientos: R3, R6, R10 De casos: Consultas e Informes Recepción de Larva, Porcentaje numérico de Supervivencia.

Tabla 4.4 Caso de uso Análisis Recepción de Larva.

4.4.2 Descripción Subcasos de Usos

Control de Análisis de Laboratorio Acuicultor

Nombre	Control Patológico, Ingreso Datos Patológicos
Alias	
Actores	Responsable
Función	Permitir la creación de nuevos registros de información sobre datos concernientes a control patológico del camarón. El sistema debe validar que el código esté disponible y que se valido para ser ingresado.
Referencias	De requerimientos: R4

Tabla 4.5 Subcaso de uso ingreso Control Patológico.

Nombre	Control Patológico: Modificación de datos Patológicos
Alias	
Actores	Responsable
Función	Permitir la modificación de registros de información de datos de control patológico del camarón. El sistema debe validar que el código exista, que no esté utilizado.
Referencias	De Requerimientos: R4

Tabla 4.6 Subcaso de uso modificación Control Patológico.

Nombre	Control Patológico: Eliminación de datos Patológicos
Alias	
Actores	
Función	Permitir la eliminación de registros de información de datos de Control patológico del camarón. El sistema debe eliminar el registro seleccionado por el responsable.

Tabla 4.7 Subcaso de uso eliminación Control Patológico.

Nombre	Análisis de Recepción de Larva, Ingreso datos Recepción de Larva
Alias	
Autores	Responsable
Función	Permite la creación de nuevo registro de información sobre datos concernientes a Análisis de Recepción de Larva como prueba fundamental para el inicio de una cosecha en las piscinas camaroneras. El sistema debe validar que el código esté disponible y que sea válido para ser ingresado.
Referencia	De requerimientos: R10

Tabla 4.8 Subcaso de uso ingreso Análisis Recepción de Larva.

Nombre	Análisis de Recepción de Larva, Modificación de datos de Recepción de Larva.
Alias	
Autores	Responsable
Función	Permitir la modificación de registros de información de datos de Análisis de recepción de Larva. El sistema debe validar que el código exista, que no esté terminado.
Referencias	De Requerimiento: R10

Tabla 4.9 Subcaso de uso modificación Análisis de Recepción de Larva.

Nombre	Análisis de Recepción de Larva, Eliminación de datos de Recepción de Larva
Alias	
Autores	Responsable
Función	Permitir la eliminación de registro de información de datos del formulario Recepción de Larva. El sistema debe eliminar el registro seleccionado por técnico responsable de manipulación y control.

Tabla 4.10 Subcaso de uso eliminación Análisis de Recepción de Larva.

Nombre	Análisis de Fitoplancton, Ingreso de datos de Fitoplancton
Alias	
Autores	Responsable
Función	Permitir la creación de nuevo registro de información sobre datos concernientes a los análisis de Fitoplancton diarios que se hacen a las piscinas y precriaderos. El sistema debe validar que el código esté disponible y que sea válido para ser ingresado.

Tabla 4.11 Subcaso de uso ingreso Análisis de Fitoplancton.

Nombre	Análisis de Fitoplancton, Modificación de datos de Fitoplancton
Alias	
Autores	Responsable
Función	Permitir La modificación de registros de información de datos de los análisis de Fitoplancton diarios que se hacen al el camarón provenientes de las distintas piscinas y precriaderos. El sistema debe validar que el código exista, y no esté ya ocupado.

Tabla 4.12 Subcaso de uso modificación Análisis de Fitoplancton.

Nombre	Análisis de Fitoplancton, Eliminación de datos de Fitoplancton
Alias	
Autores	Responsable
Función	Permitir la eliminación de registros de información de datos de Análisis de Fitoplancton que son realizados diariamente a las distintas piscinas y precriaderos. El sistema debe eliminar el registro seleccionado por el técnico o persona responsable de su control.

Tabla 4.13 Subcaso de uso eliminación Análisis de Fitoplancton.

4.5 Eventos

4.5.1 Introducción

En este formato se establecen los eventos que pueden ser generados por el actor y van a ser atendidos por cada Caso de Uso. Por evento entendemos la interacción que tiene un actor con la aplicación a través de la interfaz gráfica, como el clic de un ratón, el ingreso de un texto a un componente, el movimiento de un elemento de la interfaz. Todos los eventos van numerados en orden secuencial de acuerdo a la secuencia lógica como

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

ocurrirían en la aplicación (ciclo de vida del caso de uso). De este formato se obtiene la información para la creación de los diagramas de interacción, más específicamente el de secuencia. También se deben presentar los eventos alternos, los cuales permiten establecer las excepciones que se pueden presentar en la ejecución del programa.

Análisis de Fitoplancton	Formatos de Eventos
Nombre Caso de Uso Alias Actores	Análisis de Fitoplancton; subcaso Ingreso de Datos de Análisis de Fitoplancton. Responsable
Evento	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Selecciona opción Análisis de Fitoplancton en menú principal 3. Digitar datos nuevos de Análisis Fitoplancton. 5. Clic en grabar datos de Análisis de Fitoplancton 	<ol style="list-style-type: none"> 2. Mostrar pantalla de captura de datos de análisis de Fitoplancton 4. Capturar datos nuevos de Análisis de Fitoplancton. 6. Grabar datos en el sistema.

Tabla 4.14 Descripción de evento ingreso Análisis Fitoplancton.

Eventos Alternos

Nombre caso de uso Alias Actores	Análisis de Fitoplancton; Subcaso Ingreso de Análisis de Fitoplancton. Responsable
Numero de Línea	Respuesta del Sistema
Línea 4	Si el registro de análisis de fitoplancton ya existe no permitirá seguir y lo informará.
Línea 6	Si falla la grabación mostrar mensaje de error en el almacenamiento de información.

Tabla 4.15 Descripción de eventos alternos ingreso Análisis Fitoplancton.

Nombre caso de uso Alias Actores	Análisis de Fitoplancton; Subcaso Modificación de Análisis de fitoplancton Responsable
Evento	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Selecciona opción Análisis de Fitoplancton del menú principal. 3. Modificar datos del Análisis 	<ol style="list-style-type: none"> 2. Mostrar pantalla de modificación de datos Fitoplancton. 4. Capturar datos nuevos de Análisis de Fitoplancton. 6. Actualiza datos en el sistema

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

Fitoplancton. 5. Clic en Actualiza datos.	
--	--

Tabla 4.16 Descripción de evento modificación Análisis Fitoplancton.

Eventos Alternos

Nombre caso de uso	Análisis de Fitoplancton; Subcaso Modificación de Análisis de Fitoplancton.
Alias	
Actores	Responsable
Numero de Línea	Respuesta de Sistema
Línea 2	Si el registro a modificar no existe lo informa.
Línea 3	Si falla actualización lo informa al igual que si es exitosa.

Tabla 4.17 Descripción de eventos alternos modificación Análisis Fitoplancton

Nombre caso de uso	Análisis de Fitoplancton; Subcaso Eliminación de Análisis de Fitoplancton.
Alias	
Actores	Responsable
Evento	Respuesta del sistema
1. Seleccionar la opción Análisis de Fitoplancton del menú principal. 3. Click en Eliminar datos	2. Mostrar pantalla de Análisis de Fitoplancton con la opción de eliminar registros. 4. Elimina datos del sistema.

Tabla 4.18 Descripción de evento eliminación Análisis Fitoplancton

Eventos Alternos

Nombre caso de uso	Análisis de Fitoplancton; Subcaso Eliminación de Análisis de Fitoplancton.
Alias	
Actores	Responsable.
Numero de Línea	Respuesta de Sistema.
Línea 2	Si el registro a eliminar no existe lo informa.
Línea 3	Si falla la eliminación del registro lo informa al igual que si la eliminación es exitosa.

Tabla 4.19 Descripción de eventos alternos eliminación Análisis Fitoplancton

Nombre caso de uso	Control Patológico; Subcaso Ingreso datos Control Patológico
Alias	
Actores	Responsable
Evento	Respuesta del Sistema
1. Seleccionar la opción Control Patológico del menú principal. 3. Digitar datos	2. Mostrar pantalla de captura de datos de Control Patológico. 4. Capturar datos nuevos de Control Patológico, 6. Grabar datos en el Sistema.

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

nuevos de control patológico. 5. Clic en grabar datos,	
---	--

Tabla 4.20 Descripción de evento ingreso Análisis Control Patológico

Eventos Alternos

Nombre caso de uso	Control Patológico; Subcaso Ingreso datos Control Patológico
Alias	
Actores	Responsable
Numero de Línea	Respuesta del Sistema
Línea 4	Si el registro de Control Patológico ya existe no permitirá seguir y lo informará.
Línea 6	Si falla la grabación mostrar mensaje de error en el almacenamiento de información.

Tabla 4.21 Descripción de eventos alternos ingreso Análisis Control Patológico

Nombre caso de uso	Control Patológico; Subcaso Modificación de Control Patológico
Alias	
Actores	Responsable
Evento	Respuesta del Sistema
1. Selecciona opción Control Patológico del menú principal. 3. Modificar datos del Control Patológico. 5. Clic en Actualiza datos.	2. Mostrar pantalla de Control Patológico. 4. Capturar datos nuevos de Control Patológico. 6. Actualiza datos en el sistema

Tabla 4.22 Descripción de evento modificación Análisis Control Patológico

Eventos Alternos

Nombre caso de uso	Control Patológico; Subcaso Modificación de Control Patológico.
Alias	
Actores	Responsable
Numero de Línea	Respuesta de Sistema
Línea 2	Si el registro a modificar no existe lo informa.
Línea 3	Si falla actualización lo informa al igual que si es exitosa.

Tabla 4.23 Descripción de eventos alternos modificación Análisis Control Patológico

Nombre caso de uso	Control Patológico; Suceso Eliminación de registro de Control Patológico.
Alias	
Actores	Responsable
Evento	Respuesta del sistema
1. Seleccionar la	2. Mostrar pantalla de Control Patológico con la opción de

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

opción Control Patológico del menú principal. 3. Clic en Eliminar datos	eliminar registros. 4. Elimina datos del sistema.
--	--

Tabla 4.24 Descripción de evento eliminación Análisis Control Patológico

Eventos Alternos

Nombre caso de uso	Control Patológico; Subcaso Eliminación de registro de Control Patológico.
Alias	
Actores	Responsable.
Numero de Línea	Respuesta de Sistema.
Línea 2	Si el registro a eliminar no existe lo informa.
Línea 3	Si falla la eliminación del registro lo informa al igual que si la eliminación es exitosa.

Tabla 4.25 Descripción de eventos alternos eliminación Análisis Control Patológico

Nombre caso de uso	Recepción de Larva; Subcaso Ingreso datos Recepción de Larva
Alias	
Actores	Responsable
Evento	Respuesta del Sistema
1. Seleccionar la opción Recepción de Larva del menú principal. 3. Digitar datos nuevos de control patológico. 5. Clic en grabar datos,	2. Mostrar pantalla de captura de datos de Recepción de Larva. 4. Capturar datos nuevos de Recepción de Larva. 6. Grabar datos en el Sistema.

Tabla 4.26 Descripción de evento ingreso Análisis Recepción de Larva

Eventos Alternos

Nombre caso de uso	Recepción de Larva; Subcaso Ingreso datos Recepción de Larva
Alias	
Actores	Responsable
Numero de Línea	Respuesta del Sistema
Línea 4	Si el registro de Recepción de Larva ya existe no permitirá seguir y lo informará.
Línea 6	Si falla la grabación mostrar mensaje de error en el almacenamiento de información.

Tabla 4.27 Descripción de eventos alternos ingreso Análisis Recepción de Larva

Nombre caso de uso	Recepción de Larva; Subcaso Modificación de datos de Recepción de Larva
Alias	
Actores	Responsable
Evento	Respuesta del Sistema

Utilización de la tecnología JEE como entorno de desarrollo en una Aplicación Informática para Análisis Acuícola en una Camaronera

4	Selecciona opción Recepción de Larva del menú principal. 3. Modificar datos de la Recepción de Larva. 5. Clic en Actualiza datos.	5	Mostrar pantalla de Recepción de Larva. 4. Capturar datos nuevos de Recepción de Larva. 6. Actualiza datos en el sistema
---	---	---	--

Tabla 4.28 Descripción de evento modificación Análisis Recepción de Larva

Eventos Alternos

Nombre caso de uso	Recepción de Larva; Subcaso Modificación de datos de Recepción de Larva.
Alias	
Actores	Responsable
Numero de Línea	Respuesta de Sistema
Línea 2	Si el registro a modificar no existe lo informa.
Línea 3	Si falla actualización lo informa al igual que si es exitosa.

Tabla 4.29 Descripción de eventos alternos modificación Análisis Recepción de Larva

Nombre caso de uso	Recepción de Larva; Subcaso Eliminación de registro de Recepción de Larva.
Alias	
Actores	Responsable
Evento	Respuesta del sistema
1. Seleccionar la opción Recepción de Larva del menú principal. 6. Clic en Eliminar datos	2. Mostrar pantalla de Recepción de Larva con la opción de eliminar registros. 7. Elimina datos del sistema.

Tabla 4.30 Descripción de evento eliminación Análisis Recepción de Larva

Eventos Alternos

Nombre caso de uso	Recepción de Larva; Subcaso Eliminación de registro de Recepción de Larva.
Alias	
Actores	Responsable.
Numero de Línea	Respuesta de Sistema.
Línea 2	Si el registro a eliminar no existe lo informa.
Línea 3	Si falla la eliminación del registro lo informa al igual que si la eliminación es exitosa.

Tabla 4.31 Descripción de eventos alternos eliminación Análisis Recepción de Larva

4.6 Diagrama Conceptual

4.6.1 Introducción

Antes de definir el modelo estático o de clases, es necesario definir el modelo conceptual, el cual nos muestra los conceptos presentes en el dominio del problema. Un concepto para este caso, en términos de POO, es un objeto del mundo real, es decir, es la representación de cosas del mundo real y NO de componentes de software. En él no se definen operaciones (o métodos). En este modelo se pueden mostrar los conceptos, los atributos de los conceptos (opcionalmente) y la relación o asociación entre ellos. Informalmente podríamos decir que un concepto es una idea, cosa u objeto. Para descubrirlos debemos analizar los sustantivos en las descripciones textuales del dominio del problema, es decir, de la descripción del sistema, de los requerimientos y de los Casos de Uso.

Diagrama Conceptual

Análisis de control de Camaronera

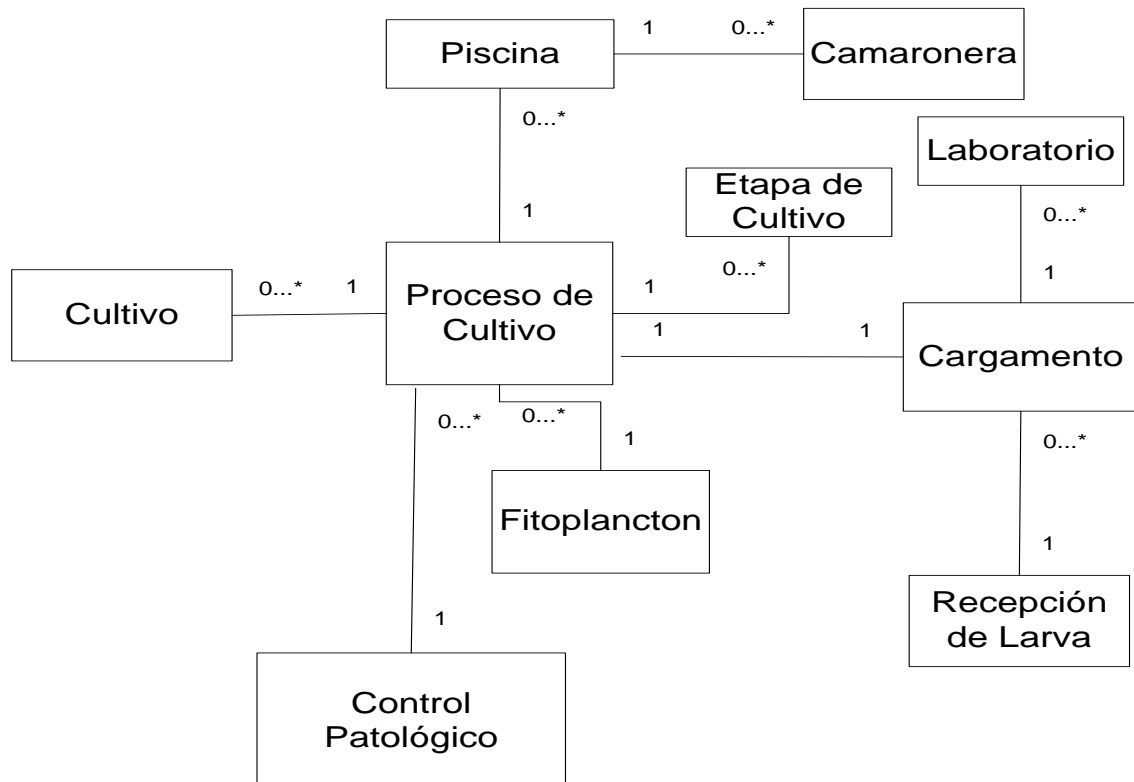


Figura. 4.3 Diagrama Conceptual

4.7 Diagrama de Estructura Estática (de Clases)

4.7.1 Introducción

Nos muestra una vista de la aplicación en un determinado momento, es decir, en un instante en que el sistema está detenido. Las clases son la plantilla de los objetos, y aquí podemos ver representados estos con sus atributos o características y su comportamiento o métodos, así como la relación entre ellas.

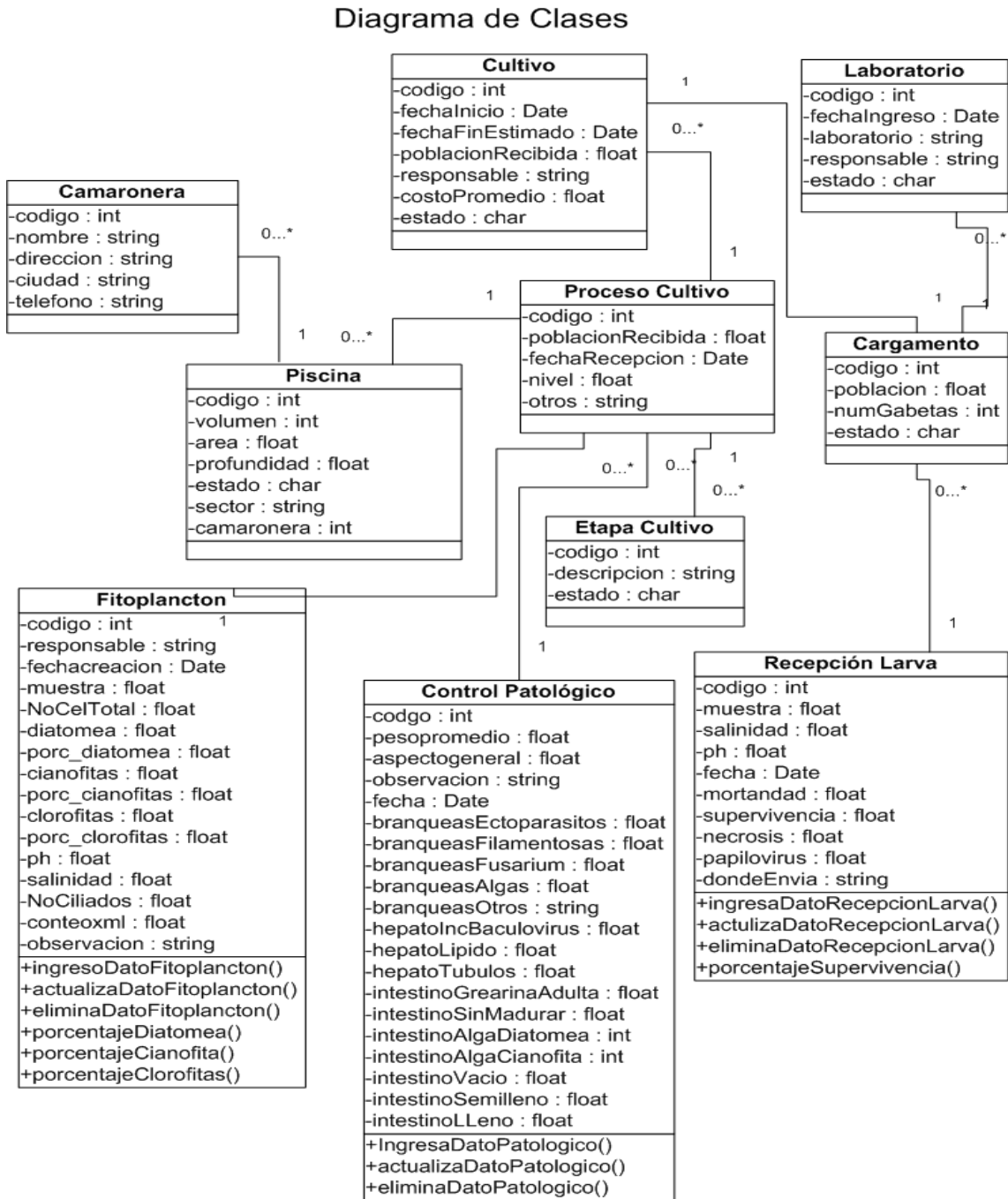


Figura. 4.4 Diagrama de Clases

4.8 Diagrama de Interacción y Colaboración

4.8.1 Introducción

Son aquellos que muestran las interacciones de un usuario con el sistema. Interacción es una cadena de mensajes enviados entre los objetos en respuesta a un evento generado por el usuario sobre la aplicación.

Los diagramas de interacción pueden ser Diagramas de Secuencia y Diagramas de Colaboración. Estos diagramas conforman la etapa del diseño de la aplicación, y se crean a partir de los diagramas de Casos de Uso y el Conceptual.

Los Diagramas de Secuencia representan una interacción entre objetos de manera secuencial en el tiempo. Muestra la participación de objetos en la interacción entre sus “líneas de vida”, (desde que se instancias) y los mensajes que ellos organizadamente intercambian en el tiempo. El responsable o ACTOR es quien inicia el ciclo interactuando inicialmente con la interfaz de usuario: GUI; en seguida se inician todos los objetos que intervienen en el funcionamiento del aplicativo. En este diagrama se comienza a observar el comportamiento del sistema a partir de los eventos generados por los actores. Aquí se interactúa con instancias, no con clases.

Los diagramas de Colaboración dan todas las especificaciones de los métodos. Estos permiten describir una operación específica incluyendo sus argumentos y variables locales creadas durante su ejecución. Se muestran los objetos y mensajes que son necesarios para cumplir con un requerimiento o propósito, o con un conjunto de ellos. Se puede elaborar para una operación o para un Caso de Uso, con el fin de describir el contexto en el cual su comportamiento ocurre.

4.8.2 Diagrama interacción ingreso datos Análisis Fitoplancton

Control de Análisis de Fitoplancton

Diagrama Interacción – Caso de uso Ingreso Datos Análisis de Fitoplancton

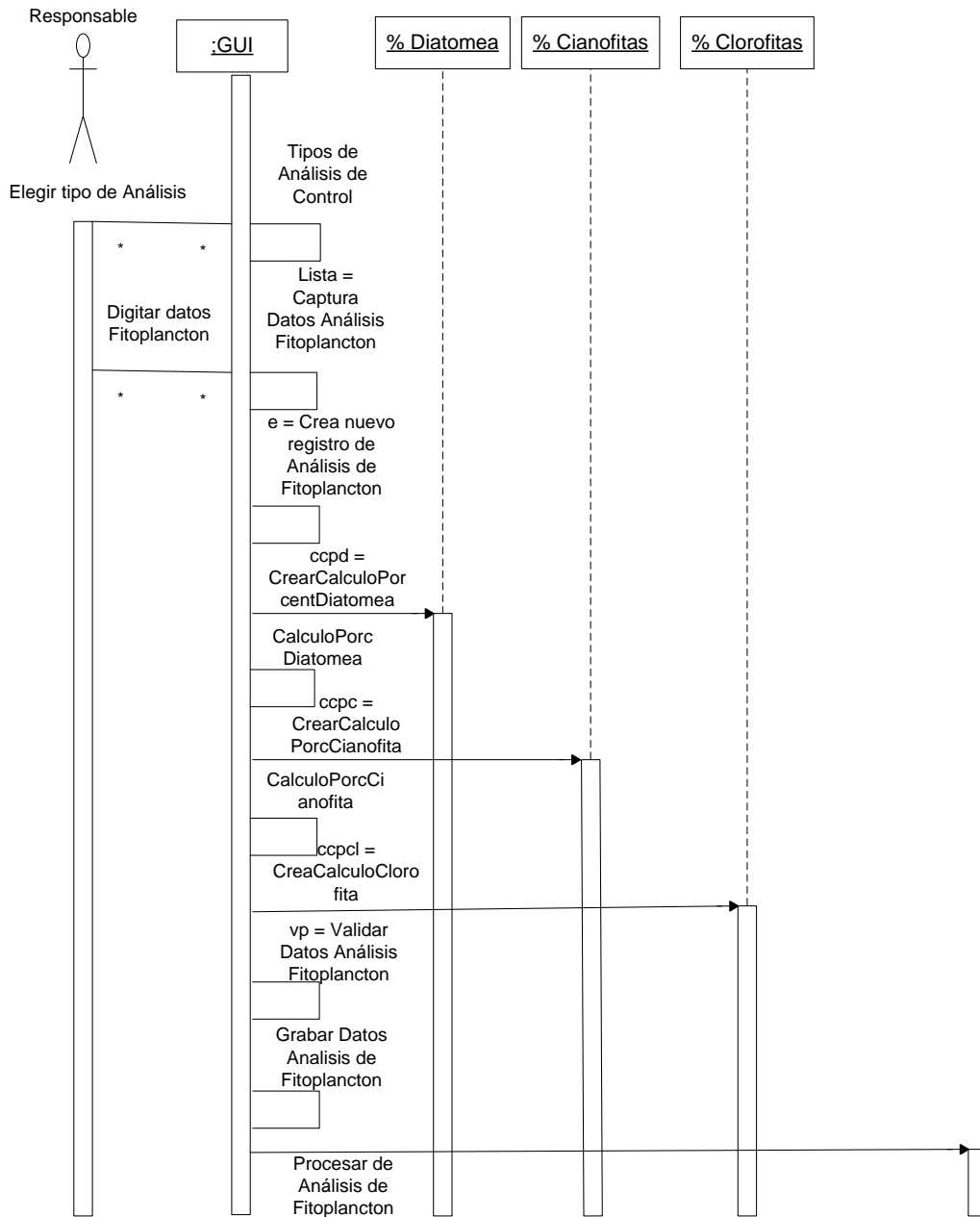


Figura. 4.5 Diagrama interacción ingreso datos Análisis Fitoplancton

4.8.2.1 Diagrama colaboración ingreso datos Análisis Fitoplancton

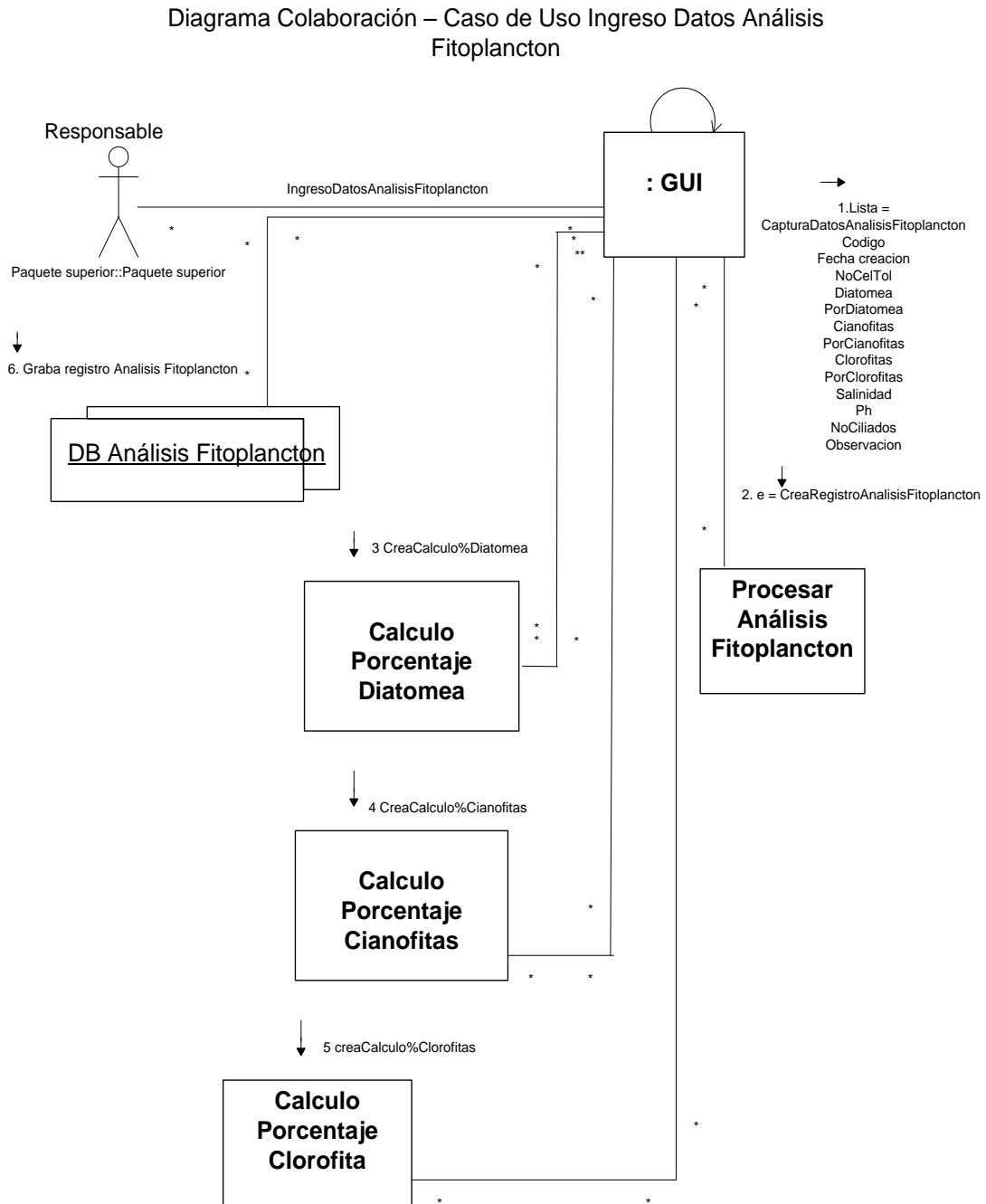


Figura. 4.6 Diagrama colaboración ingreso datos Análisis Fitoplancton

4.8.3 Diagrama interacción actualización datos Análisis Fitoplancton

Diagrama de Interacción

Control de Análisis de Fitoplancton
Diagrama Interacción – Caso de uso Actualización Datos Análisis de Fitoplancton

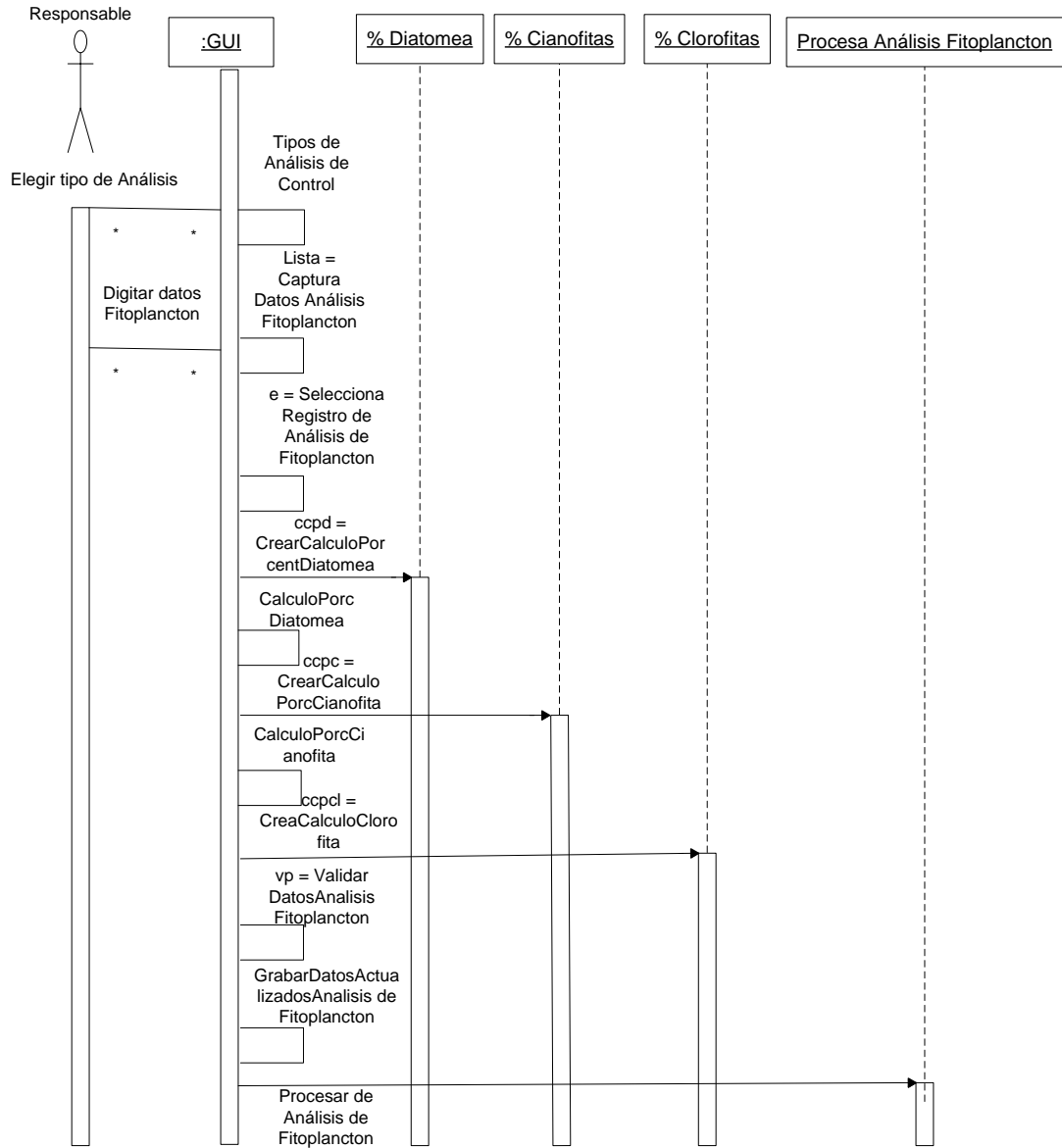


Figura. 4.7 Diagrama interacción actualización datos Análisis Fitoplancton

4.8.3.1 Diagrama colaboración actualización datos Análisis Fitoplancton

Diagrama Colaboración – Caso de Uso Actualización Datos Análisis Fitoplancton

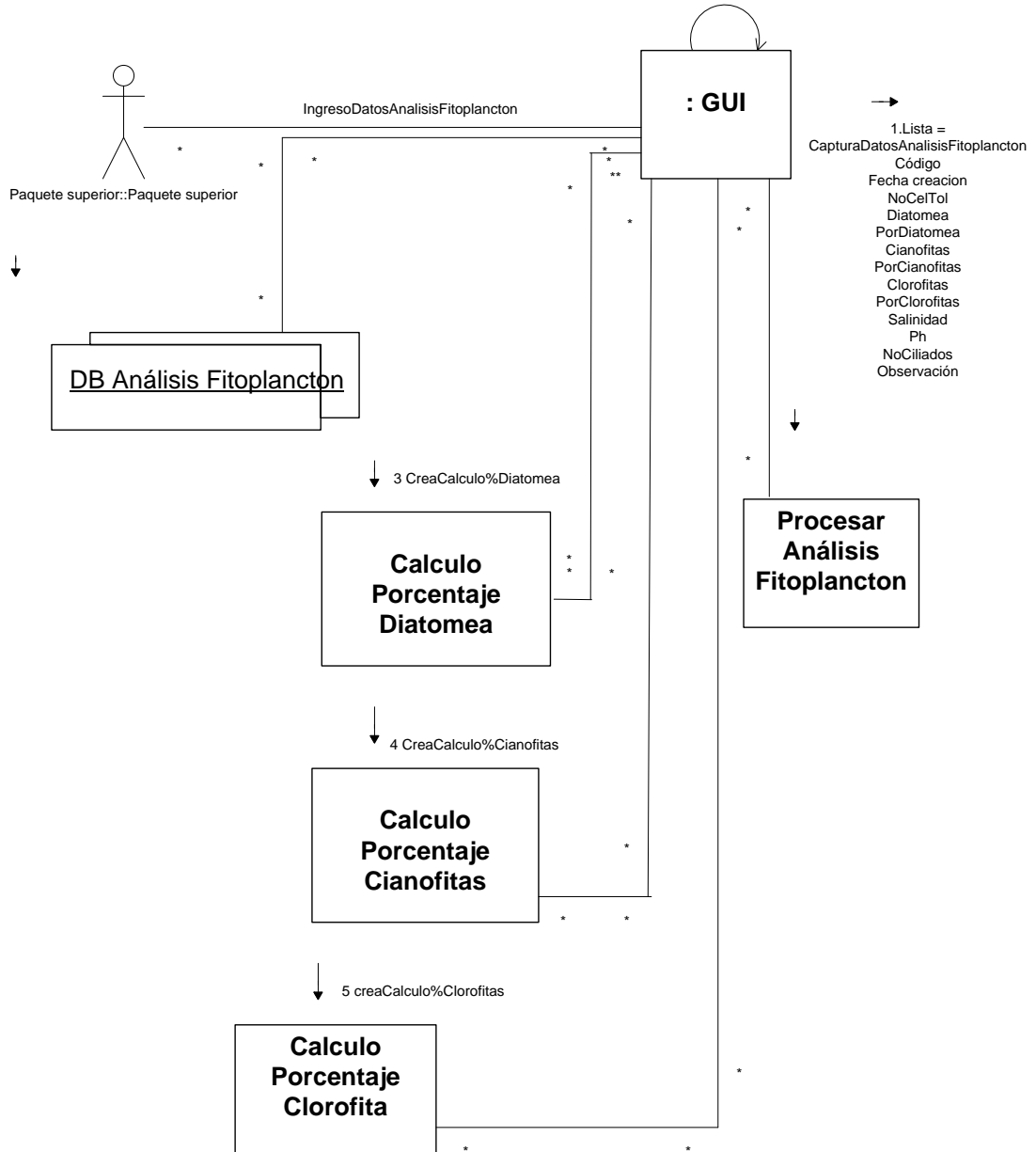


Figura. 3 Diagrama colaboración actualización datos Análisis Fitoplancton

4.8.4 Diagrama interacción eliminación datos Análisis Fitoplancton

Diagrama de Interacción

Control Análisis de Fitoplancton

Diagrama de Interacción – Caso de uso Eliminación datos Análisis de Fitoplancton

Responsable

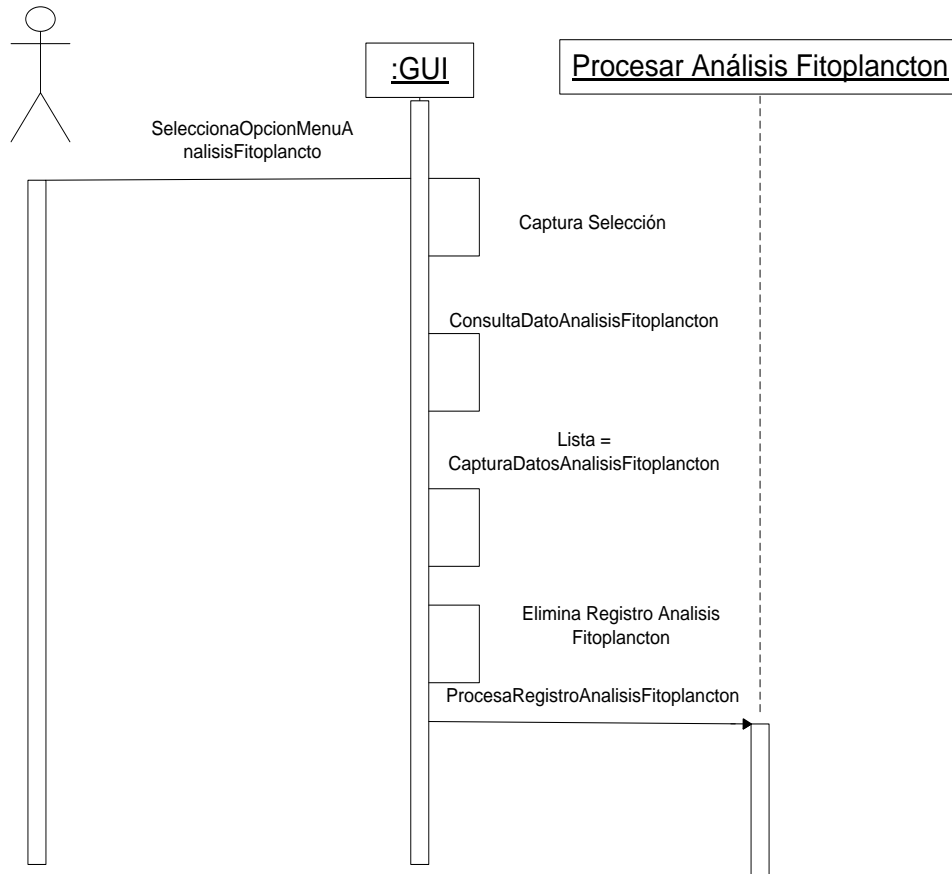


Figura. 4.9 Diagrama interacción eliminación datos Análisis Fitoplancton

4.8.4.1 Diagrama colaboración eliminación datos Análisis Fitoplancton

Casos de Usos Eliminación Datos Análisis de Fitoplancton

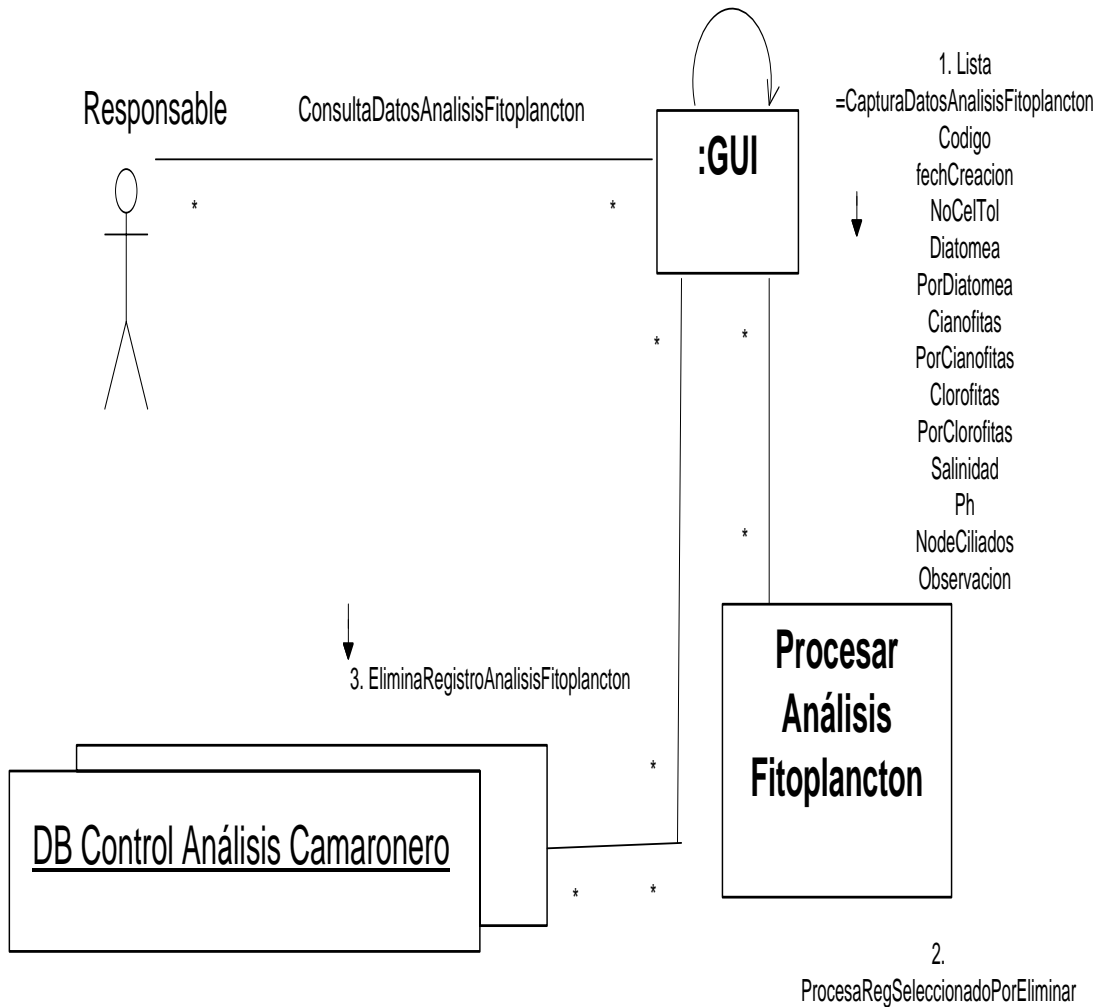


Figura. 4.10 Diagrama colaboración eliminación datos Análisis Fitoplancton

4.8.5 Diagrama interacción ingreso datos Control Patológico

Diagrama de Interacción

Análisis de Control Patológico Diagrama Interacción – Caso de uso Ingreso Datos Control Patológico

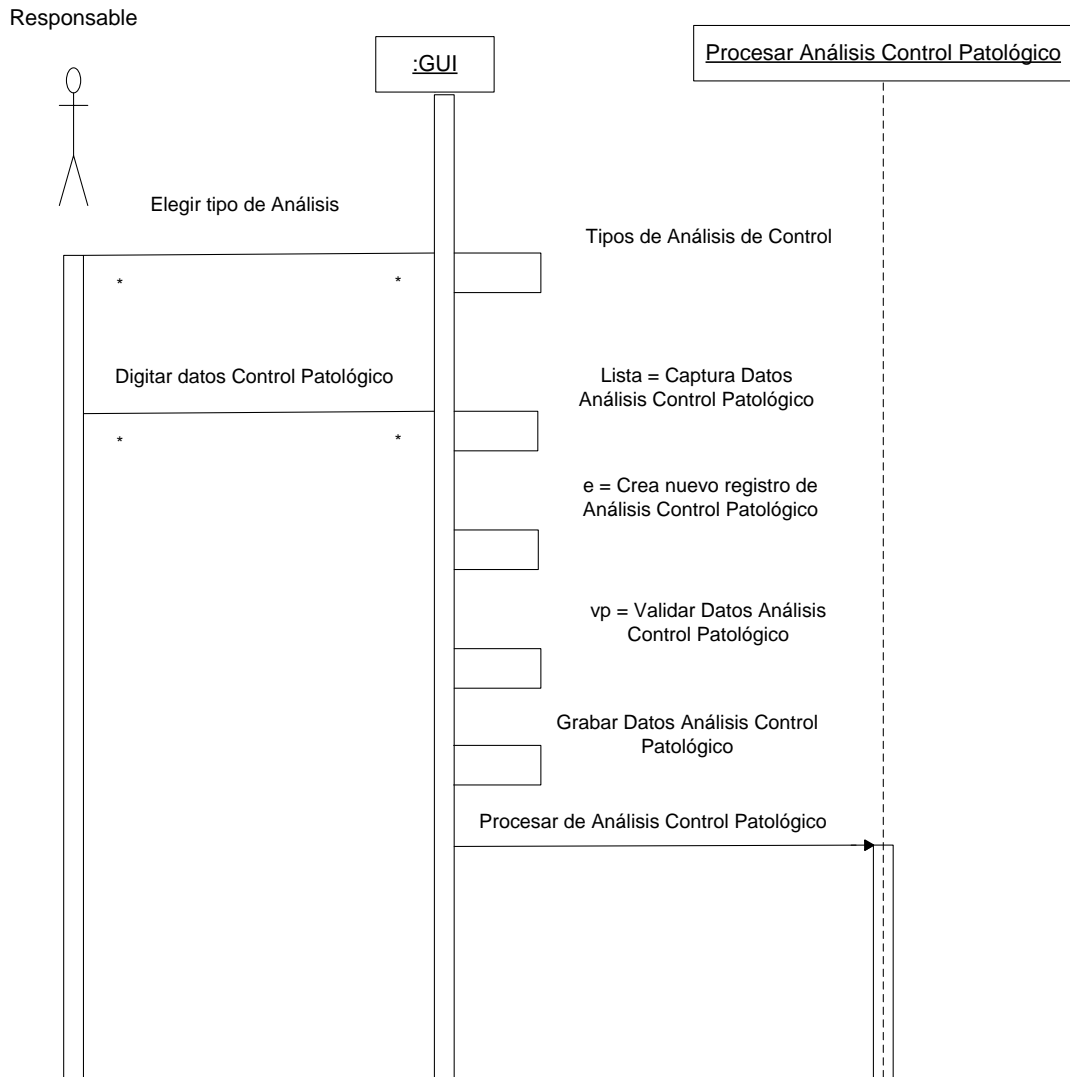


Figura. 4.11 Diagrama interacción ingreso datos Control Patológico

4.8.5.1 Diagrama colaboración ingreso datos Control Patológico

Diagrama Colaboración – Caso de Uso Ingreso Datos Análisis Control Patológico

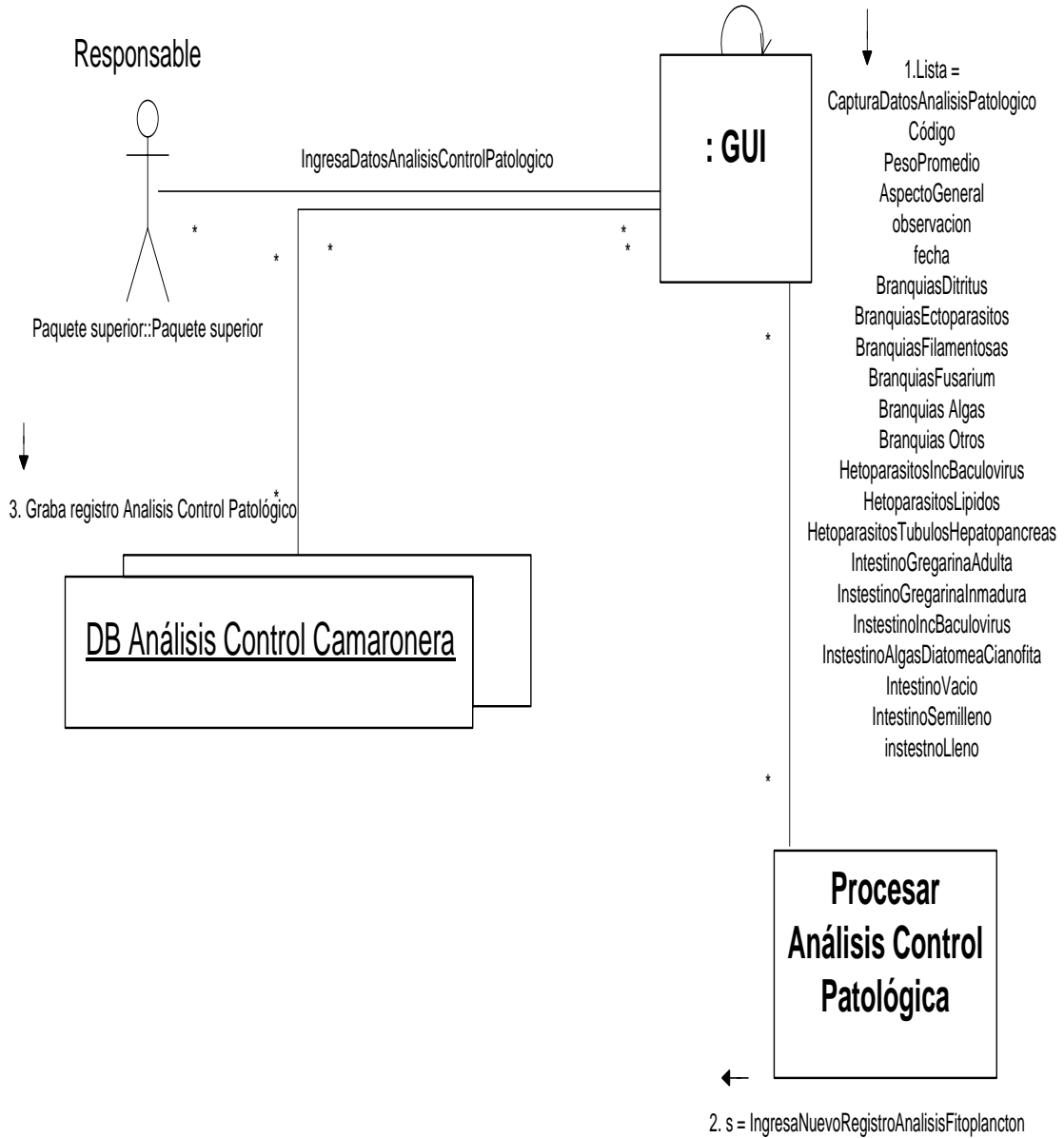


Figura. 4.12 Diagrama colaboración ingreso datos Control Patológico

4.8.6 Diagrama interacción actualización datos Control Patológico

Diagrama de Interacción

Análisis de Control Patológico

Diagrama Interacción – Caso de uso Actualización de Datos Control Patológico

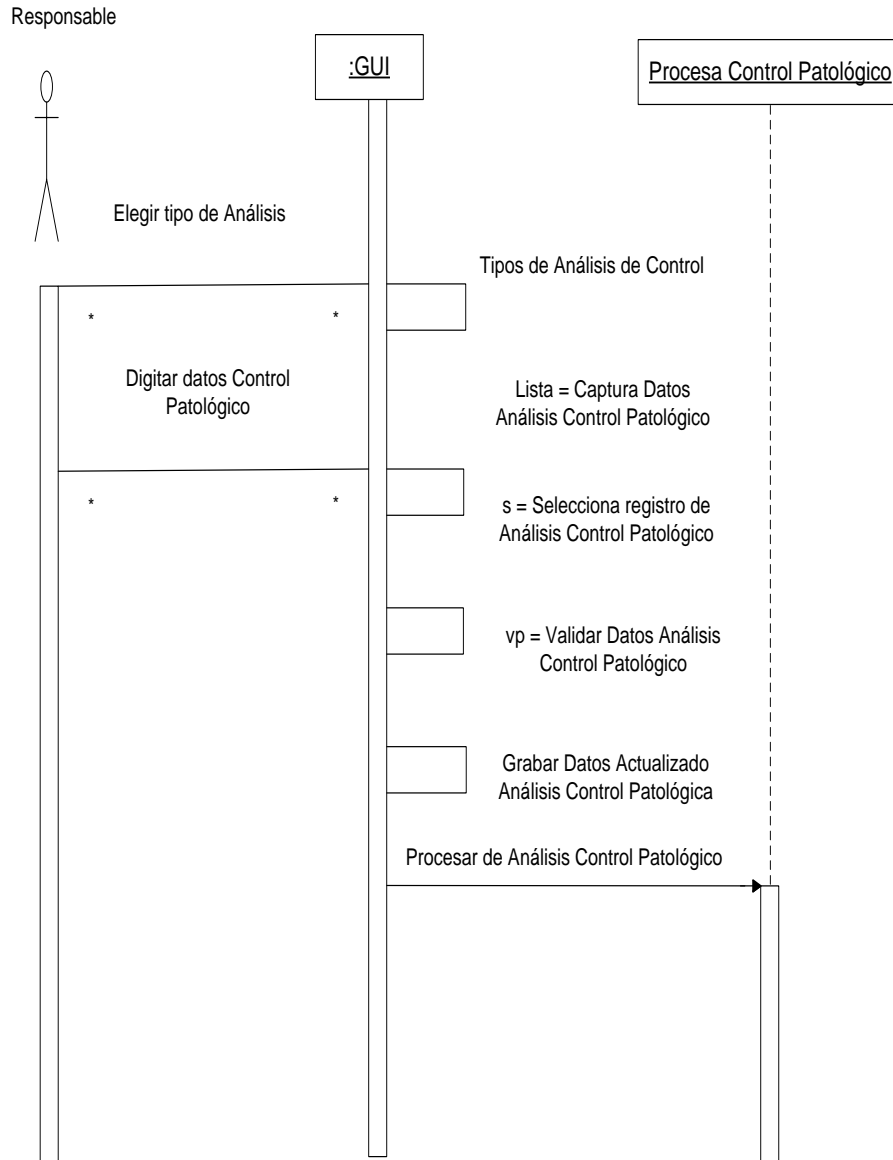


Figura. 4.13 Diagrama interacción actualización datos Control Patológico

4.8.6.1 Diagrama colaboración actualización datos Control Patológico

Diagrama Colaboración – Caso de Uso Actualización Datos Análisis Control Patológico

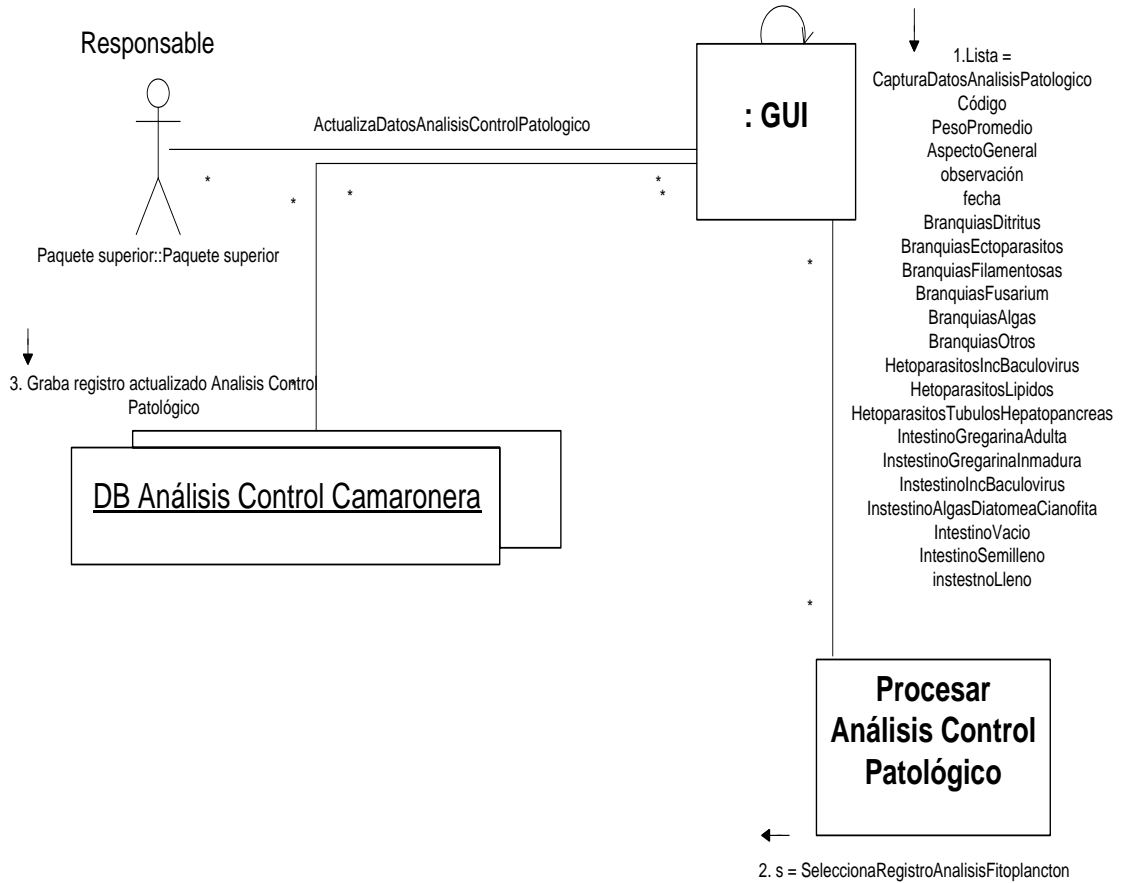


Figura. 4.14 Diagrama colaboración actualización datos Control Patológico

4.8.7 Diagrama interacción eliminación datos Control Patológico

Diagrama de Interacción

Control Análisis de Control Patológico

Diagrama de Interacción – Caso de uso Eliminación datos Análisis de Control Patológico

Responsable

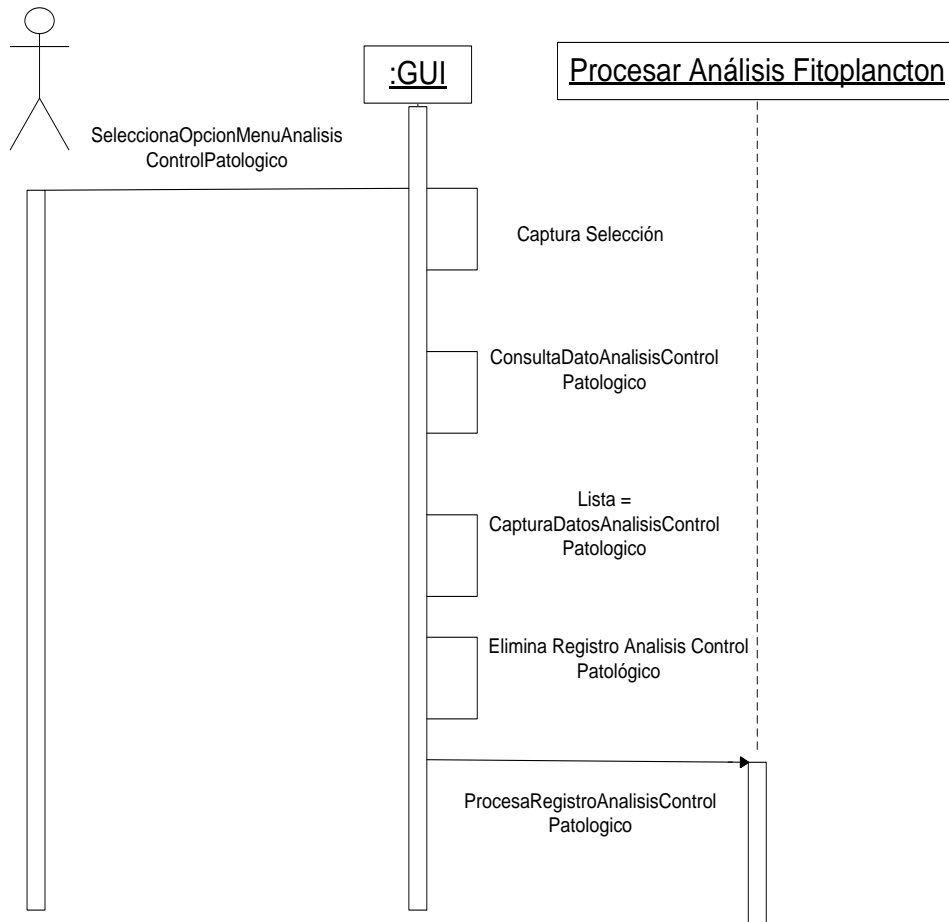


Figura. 4.15 Diagrama interacción eliminación datos Control Patológico

4.8.7.1 Diagrama colaboración eliminación datos Control Patológico

Diagrama Colaboración – Caso de Uso Eliminación Datos Análisis Control Patológico

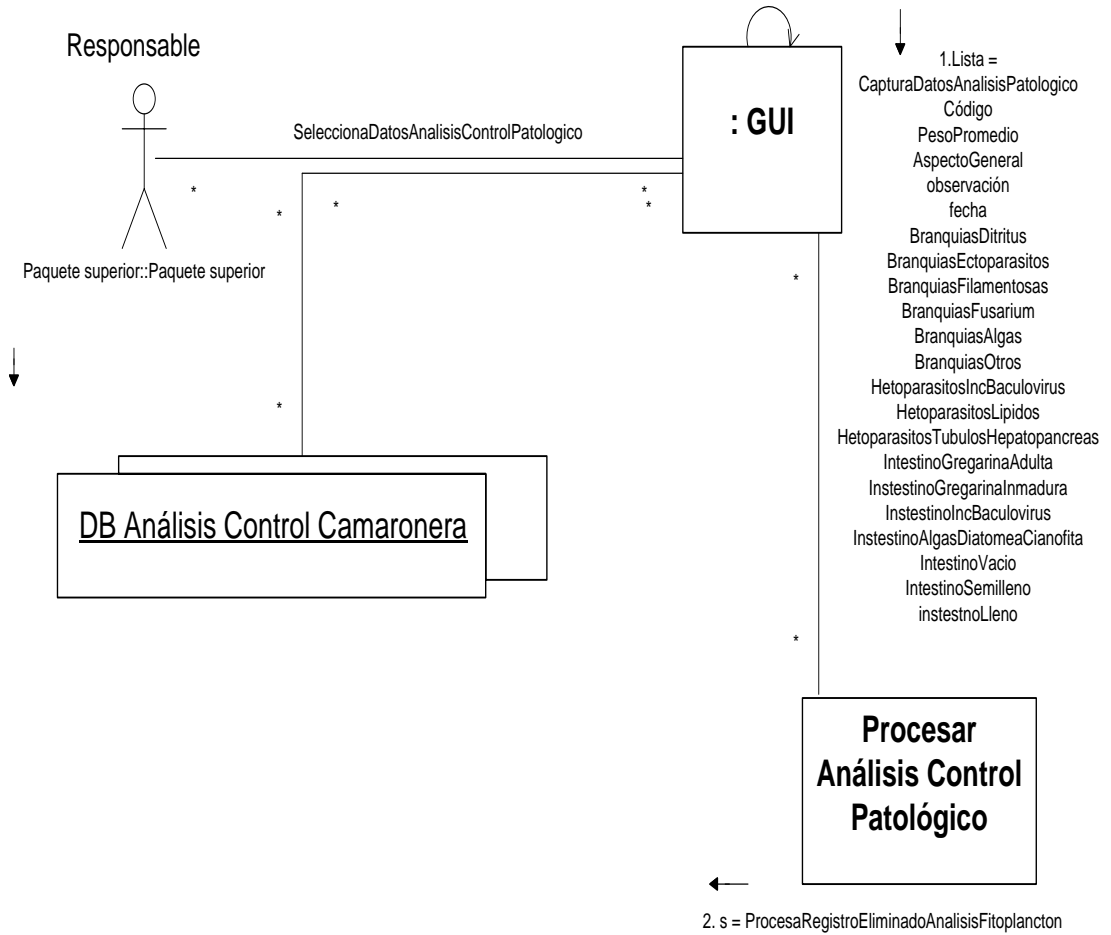


Figura. 4.16 Diagrama colaboración eliminación datos Control Patológico

4.8.8 Diagrama interacción ingreso datos Recepción de Larva

Diagrama de Interacción

Control de Análisis de Recepción de Larva

Diagrama Interacción – Caso de uso Ingreso Datos Recepción de Larva

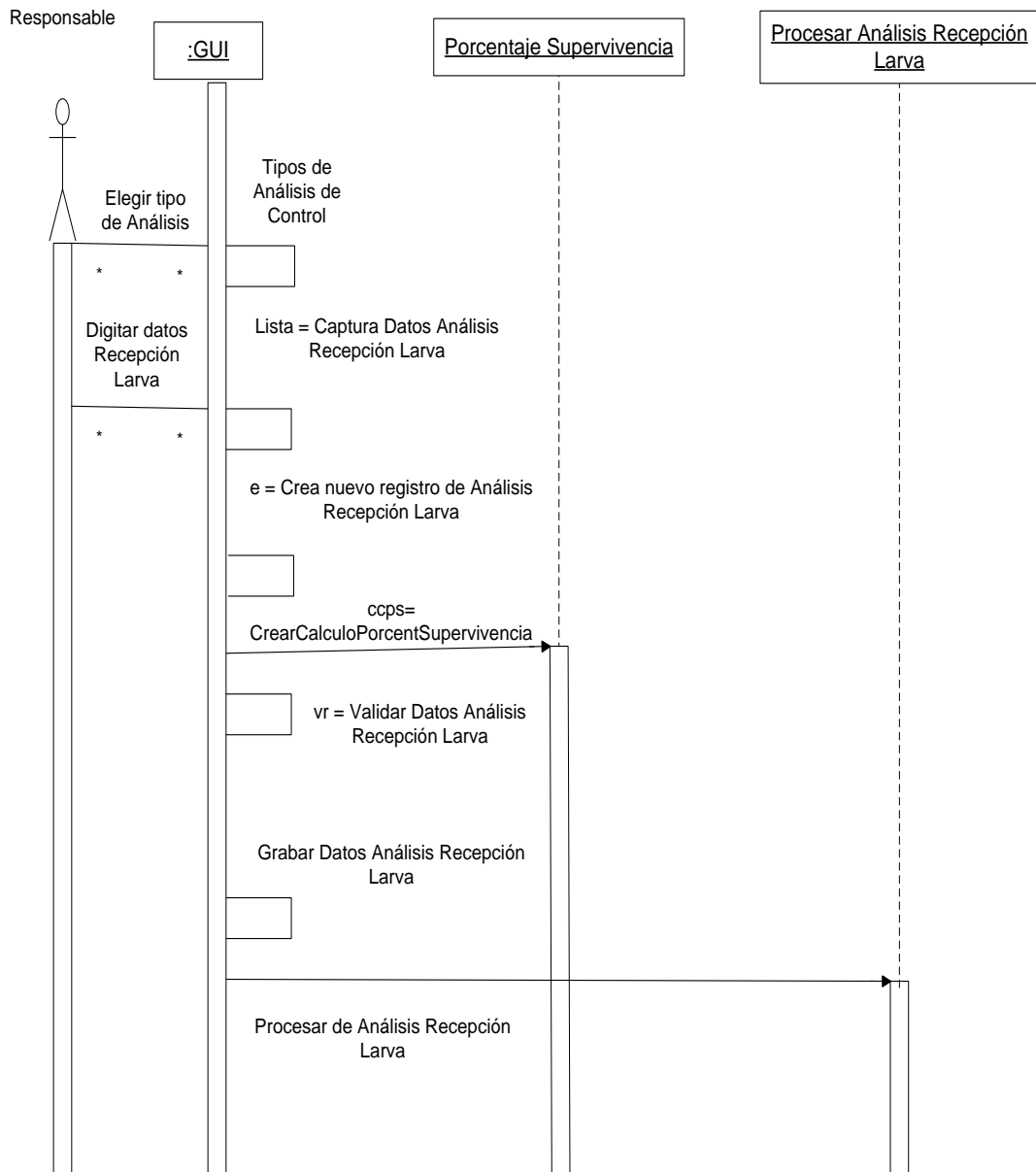


Figura. 4.17 Diagrama interacción ingreso datos Recepción de Larva

4.8.8.1 Diagrama colaboración ingreso datos Recepción de Larva

Diagrama de Colaboración: Caso de Uso Ingreso Datos Análisis de Recepción de Larva

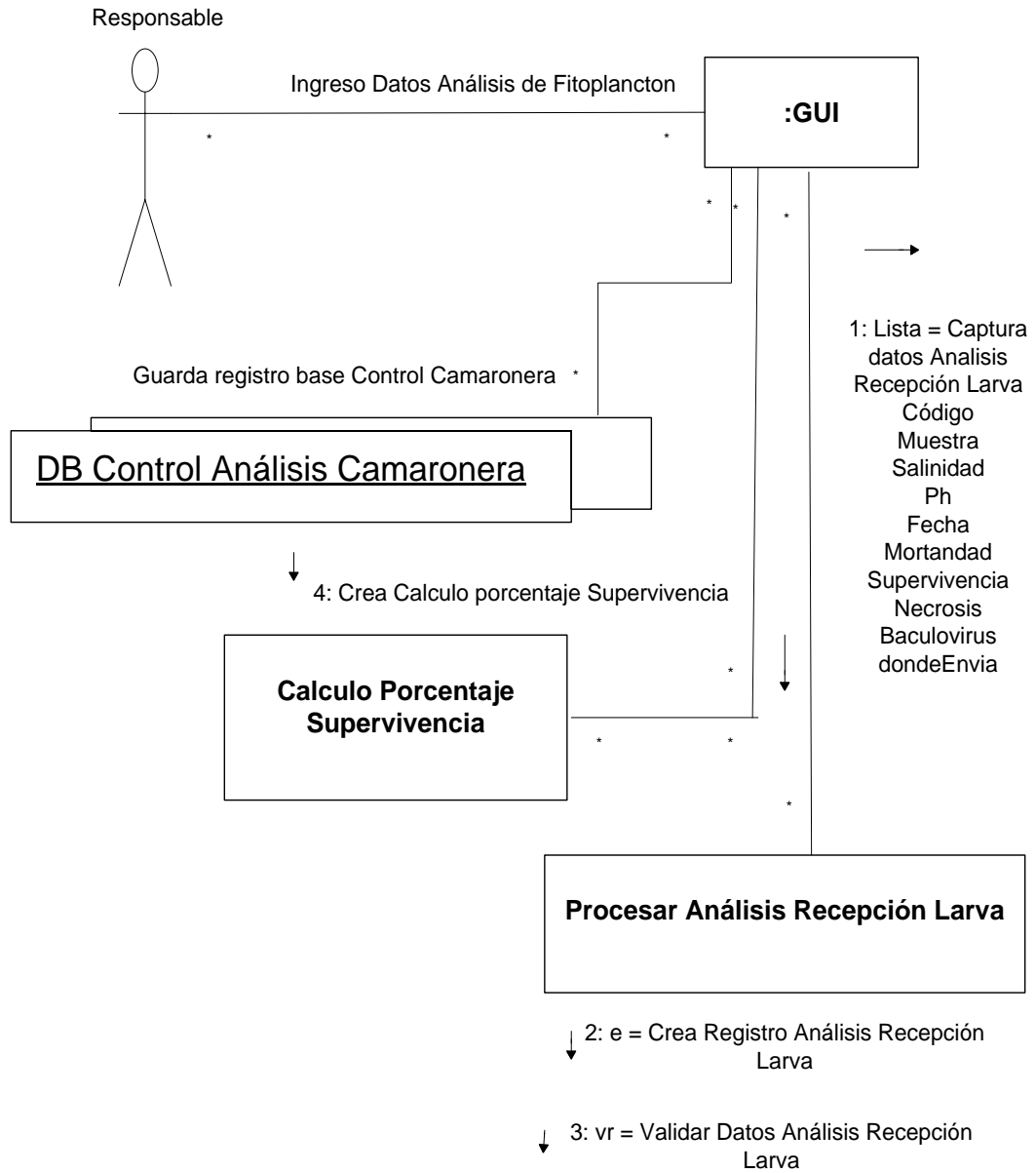


Figura. 4.18 Diagrama colaboración ingreso datos Recepción de Larva

4.8.9 Diagrama interacción actualización datos Recepción de Larva

Diagrama de Interacción

Control de Análisis de Recepción de Larva

Diagrama Interacción – Caso de uso Actualización Datos Recepción de Larva

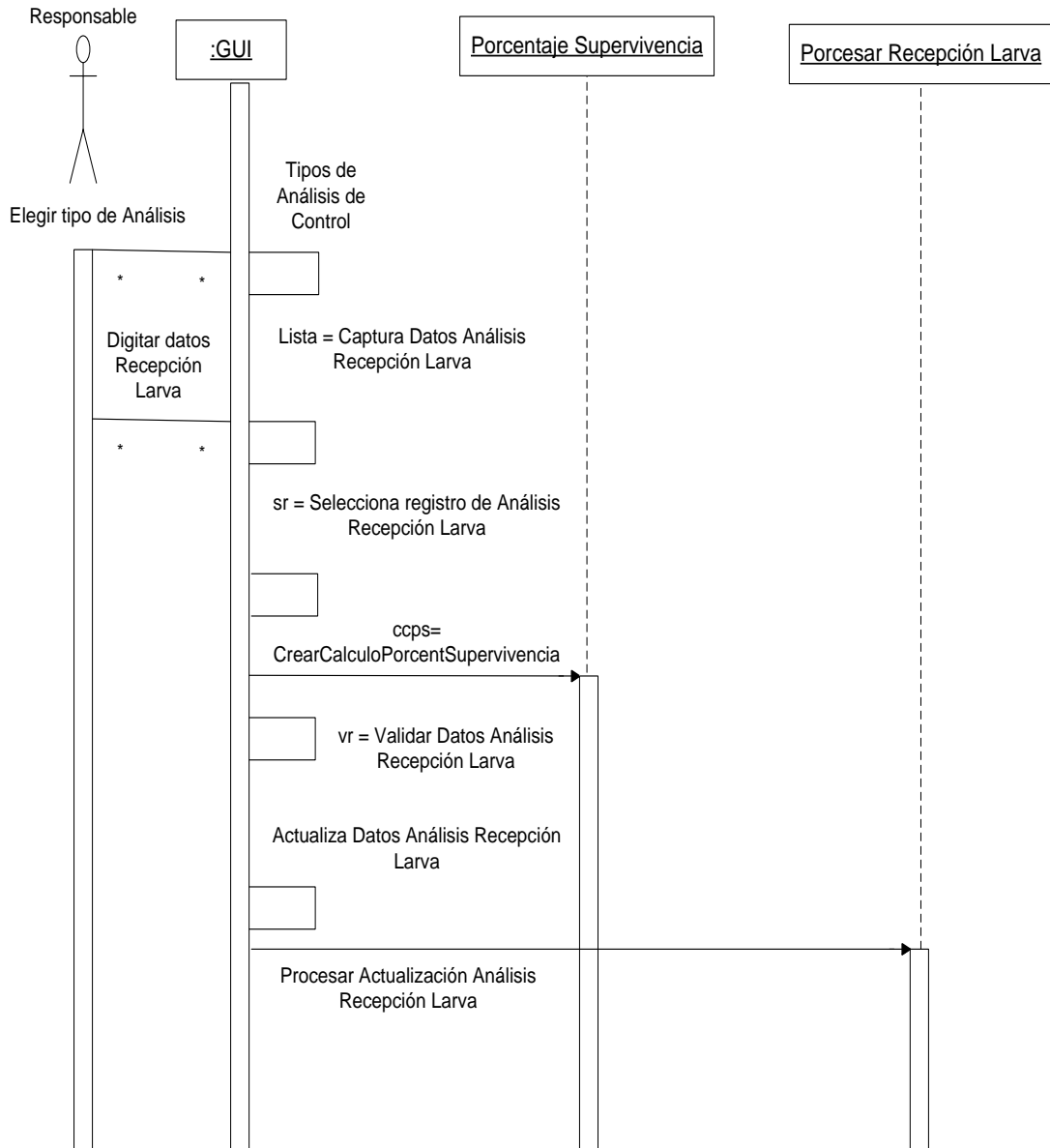


Figura. 4.19 Diagrama interacción actualización datos Recepción de Larva

4.8.9.1 Diagrama colaboración actualización datos Recepción de Larva

Diagrama de Colaboración: Caso de Uso Actualización Datos Análisis de Recepción de Larva

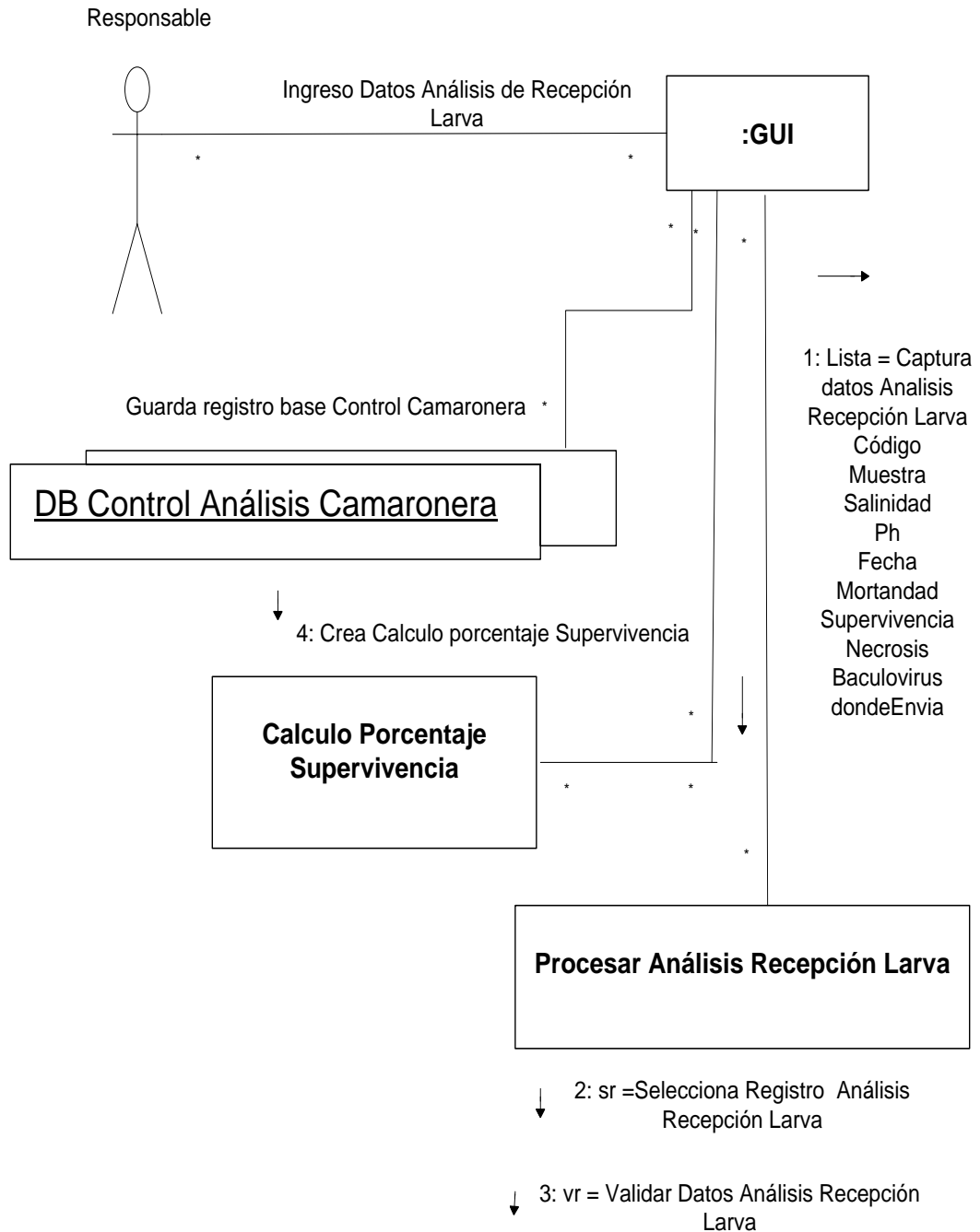


Figura. 4.20 Diagrama colaboración actualización datos Recepción de Larva

4.8.10 Diagrama interacción eliminación datos Recepción de Larva

Diagrama de Interacción

Control Análisis de Recepción Larva

Diagrama de Interacción – Caso de uso Eliminación datos Análisis de Recepción Larva

Responsable

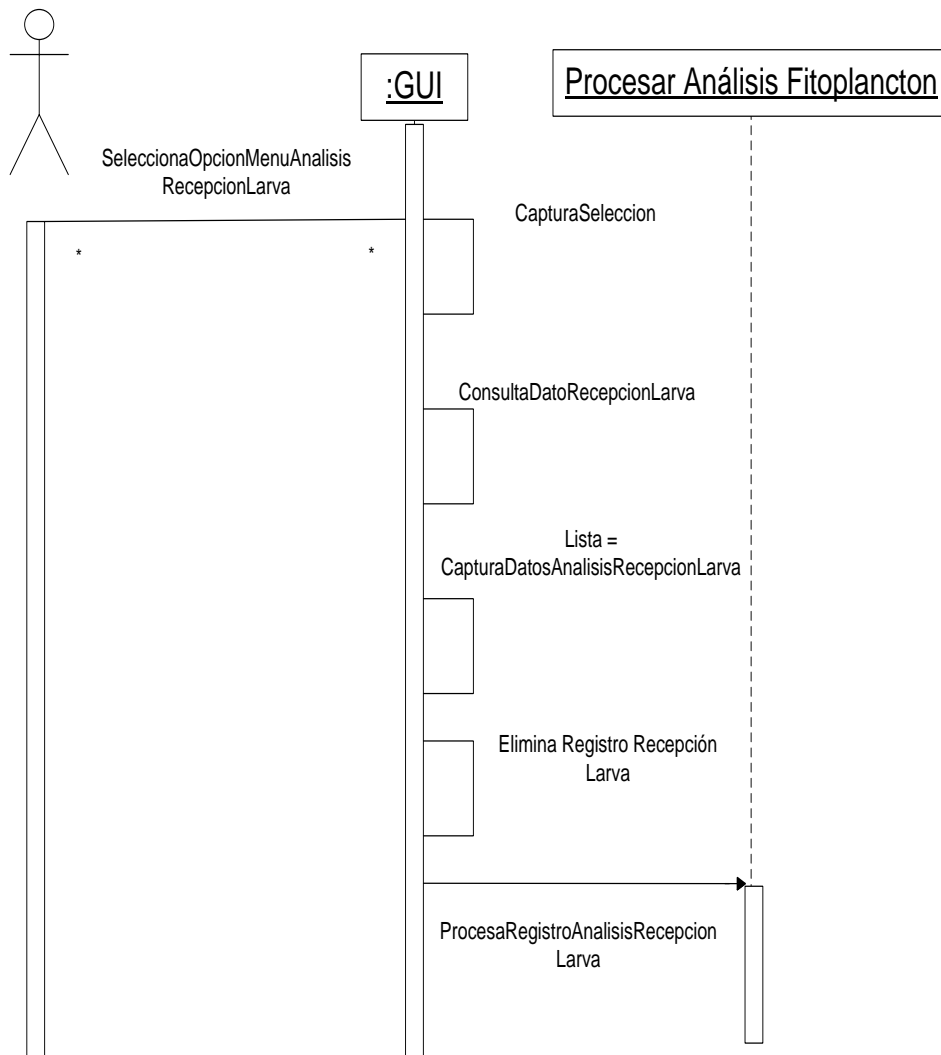


Figura. 4.21 Diagrama interacción eliminación datos Recepción de Larva

4.8.10.1 Diagrama colaboración eliminación datos Recepción de Larva

Diagrama de Colaboración

Casos de Usos Eliminación Datos Recepción de Larva

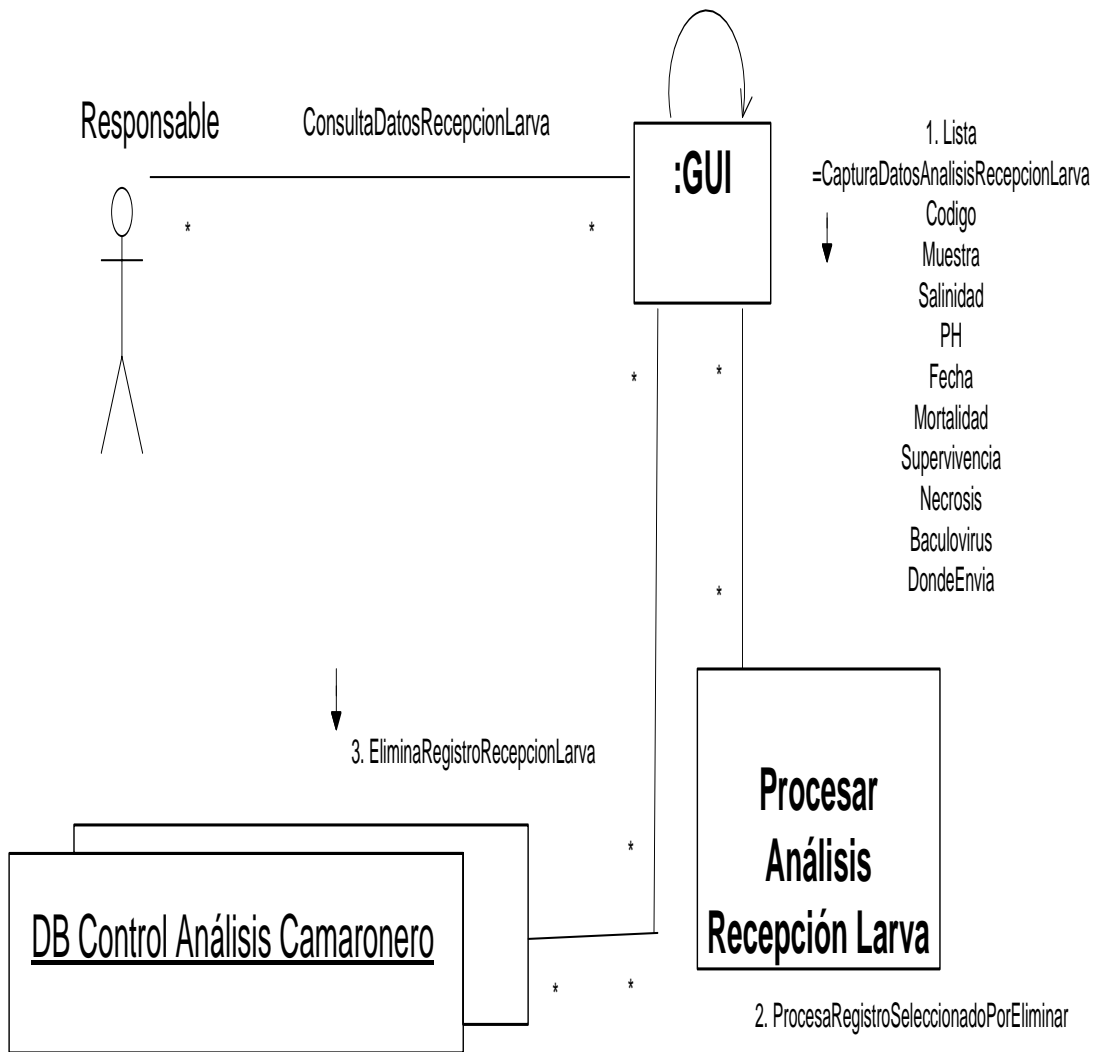


Figura. 4.22 Diagrama colaboración eliminación datos Recepción de Larva

4.9 Diagramas de Estado y Actividades

4.9.1 Introducción

Este diagrama muestra la secuencia de los estados de un objeto durante su ciclo de vida, en respuesta a un estímulo recibido. Los estados de los objetos están dados por el valor de sus atributos (estados) el cual cambia su comportamientos (métodos).

Los estados hacen referencia a una condición durante la vida de un objeto o a una interacción durante la cual se satisface alguna condición (ejecutar alguna acción, esperar algún evento, etc.), por ejemplo una validación de una captura.

Un objeto permanece en un estado por un tiempo finito, hasta que se cumpla la condición de cambio. Se construyen a partir del Diagrama de Estructura Estática, identificando cuáles objetos cambian de estado, cual es el estado inicial y el final, definiendo a qué eventos puede responder el objeto, y qué transacciones ejecutará.

4.9.1.1 Diagrama de Estado ingreso datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

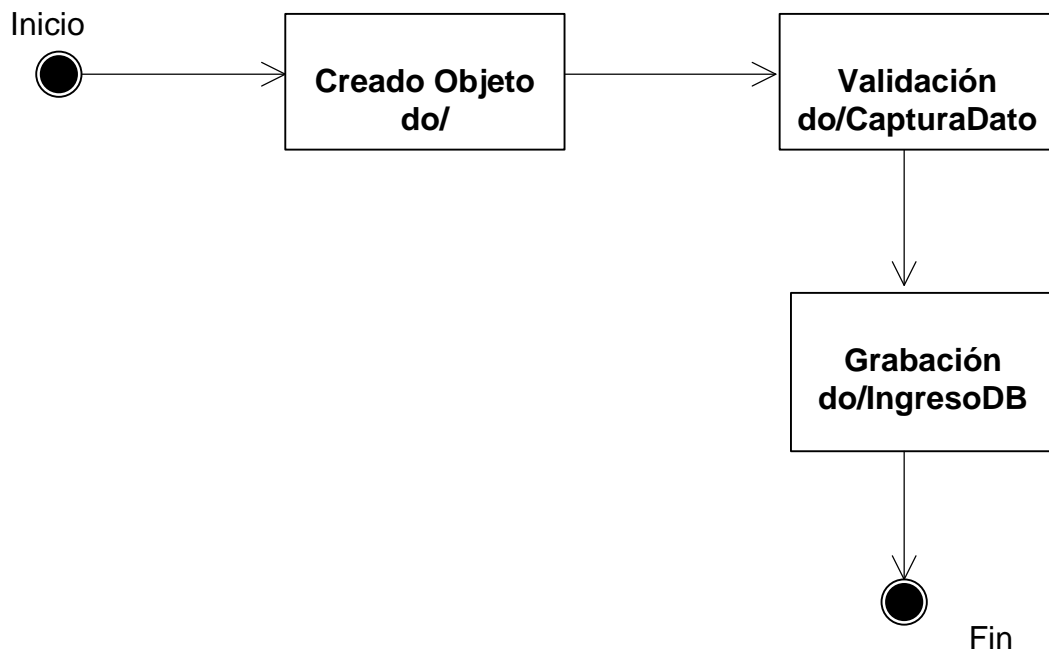


Figura. 4.23 Diagrama de Estado ingreso datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

4.9.1.2 Diagrama de Estado actualización datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

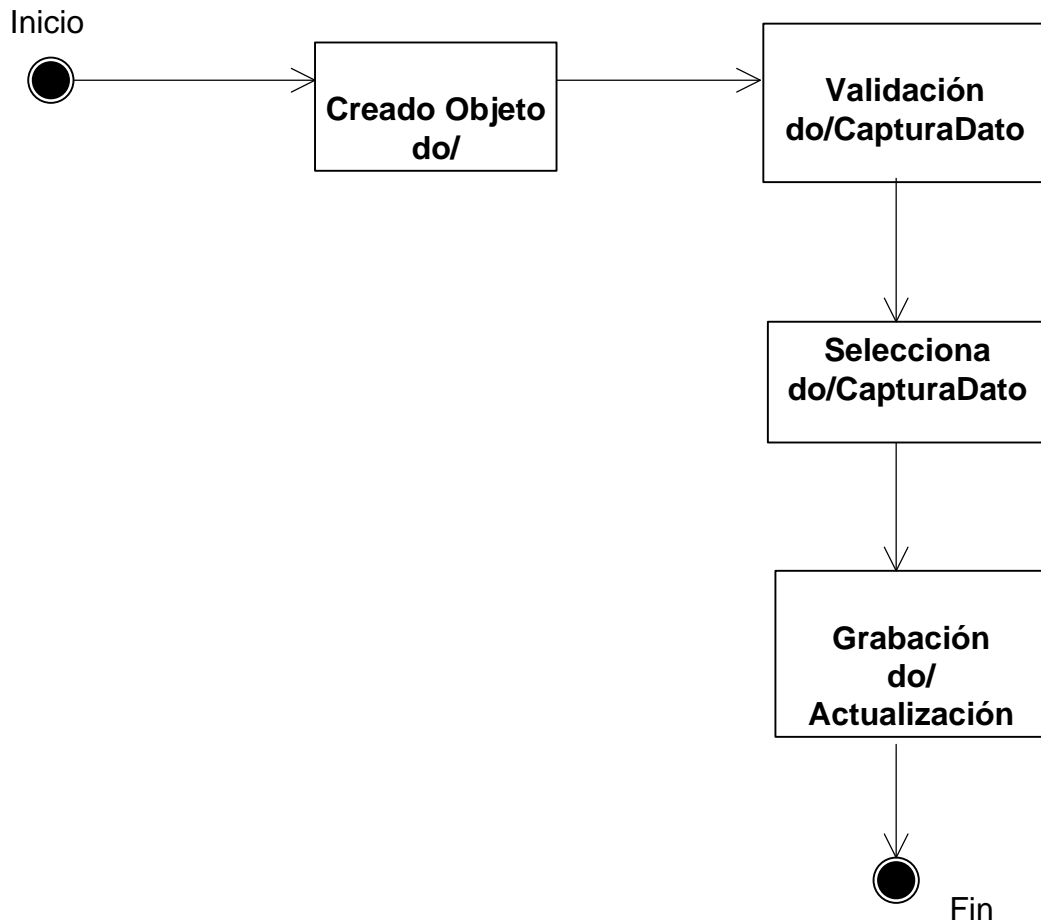


Figura. 4.24 Diagrama de Estado actualización datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

4.9.1.3 Diagrama de Estado eliminación datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

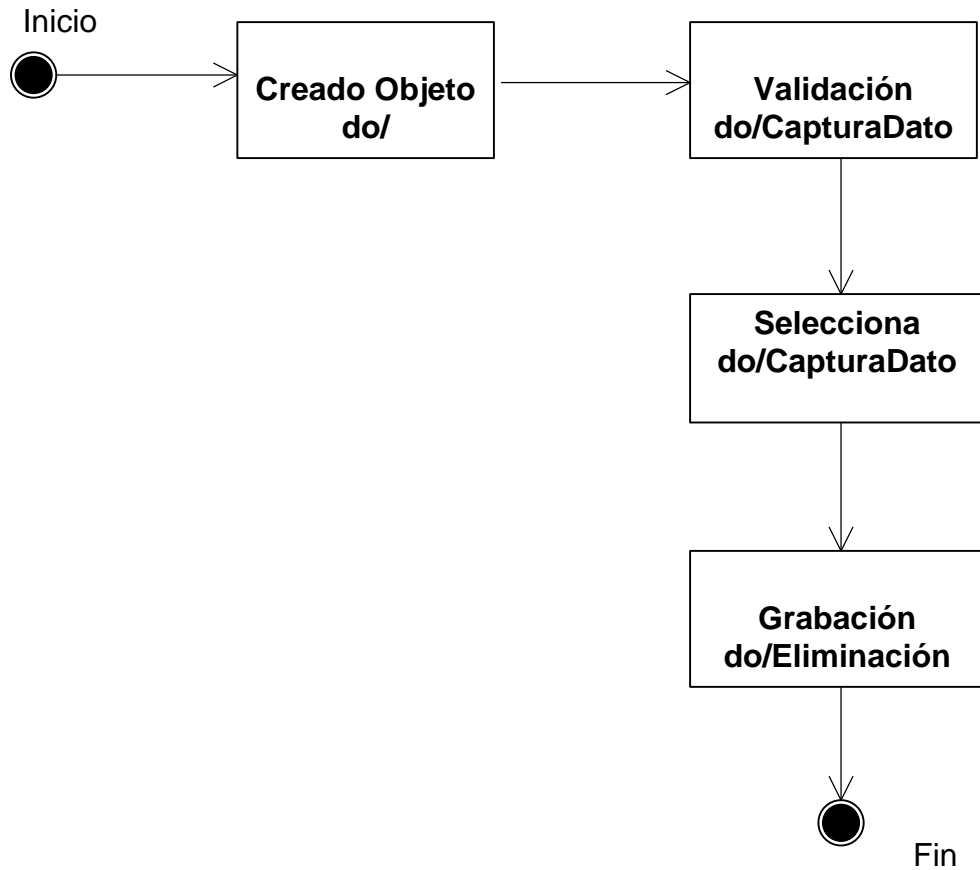


Figura. 5 Diagrama de Estado eliminación datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

4.9.2 Diagrama Actividades

Se utilizan para visualizar, especificar, construir y documentar la dinámica de un conjunto de objetos o simplemente para modelar el flujo de control de una operación (método de una clase). Fundamentalmente es un Diagrama de Flujo que muestra el flujo de control entre las actividades. Dentro del flujo se pueden encontrar pasos secuenciales y/o concurrentes y/o condiciones. Permiten en un momento dado construir sistemas ejecutables a través de ingeniería directa (del modelo al .exe) o inversa (de la implementación al modelado)

4.9.2.1 Diagrama de Actividades ingreso datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

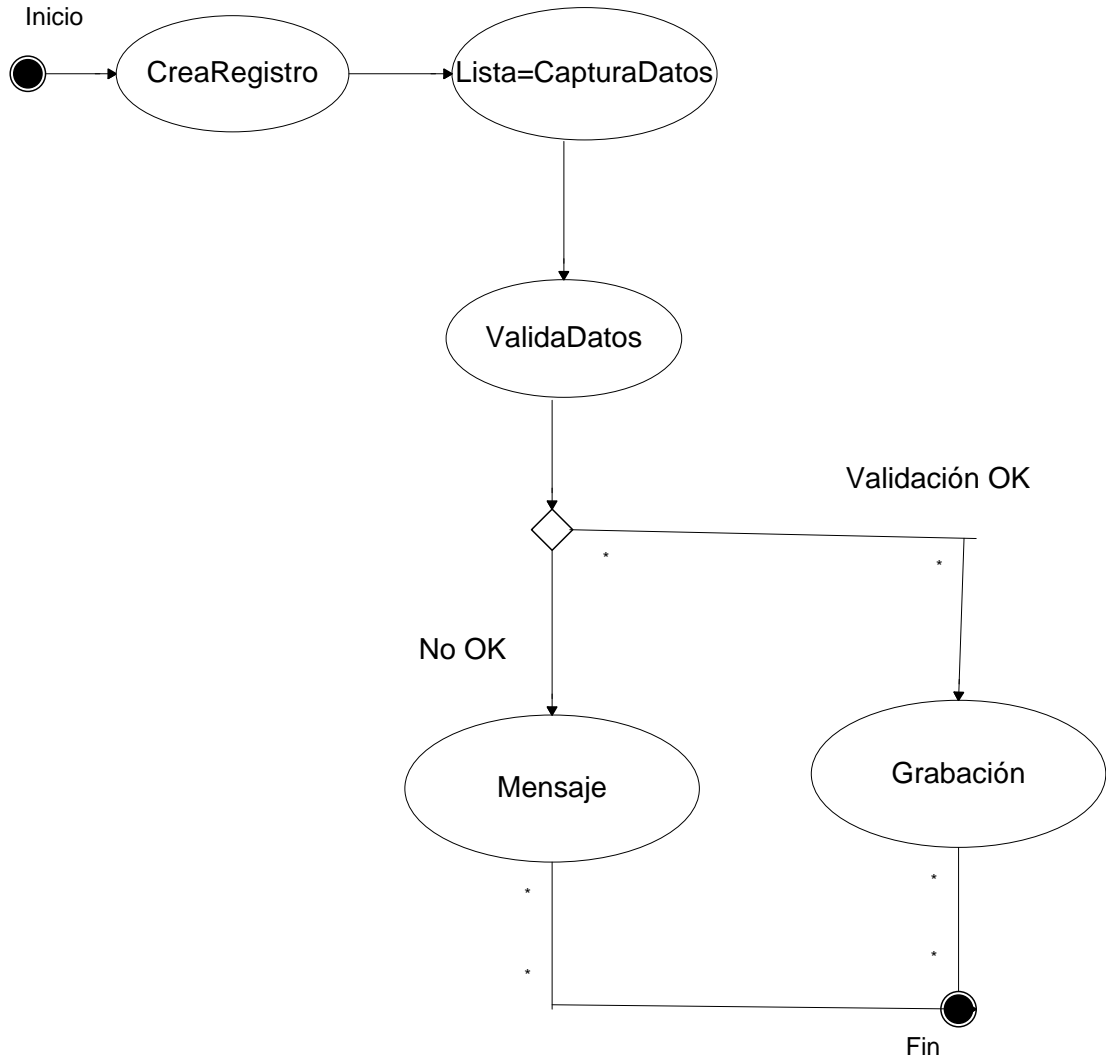


Figura. 4.26 Diagrama de Actividades ingreso datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

4.9.2.2 Diagrama de Actividades actualización datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

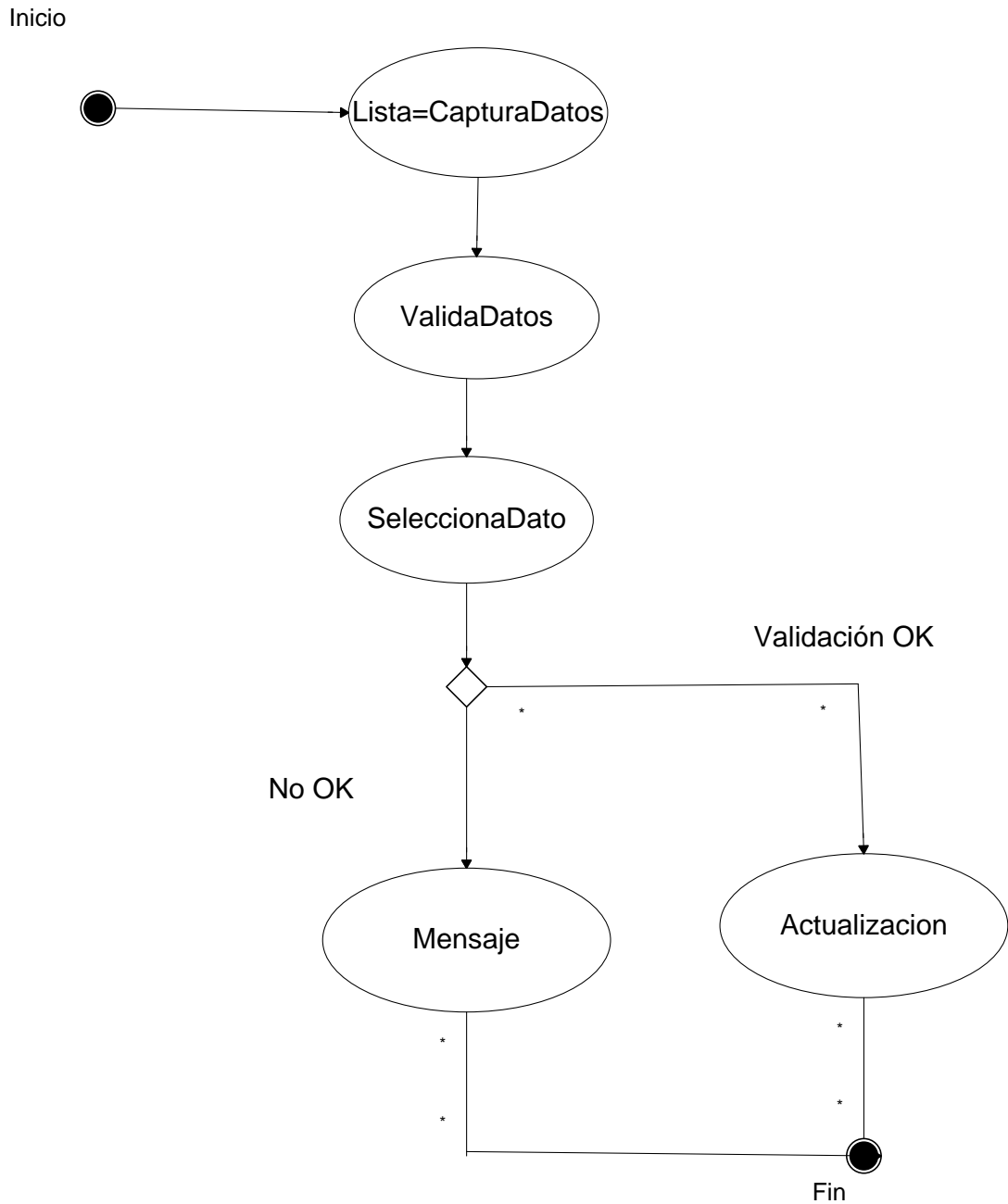


Figura. 4.27 Diagrama de Actividades actualización datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

4.9.2.3 Diagrama de Actividades eliminación datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva

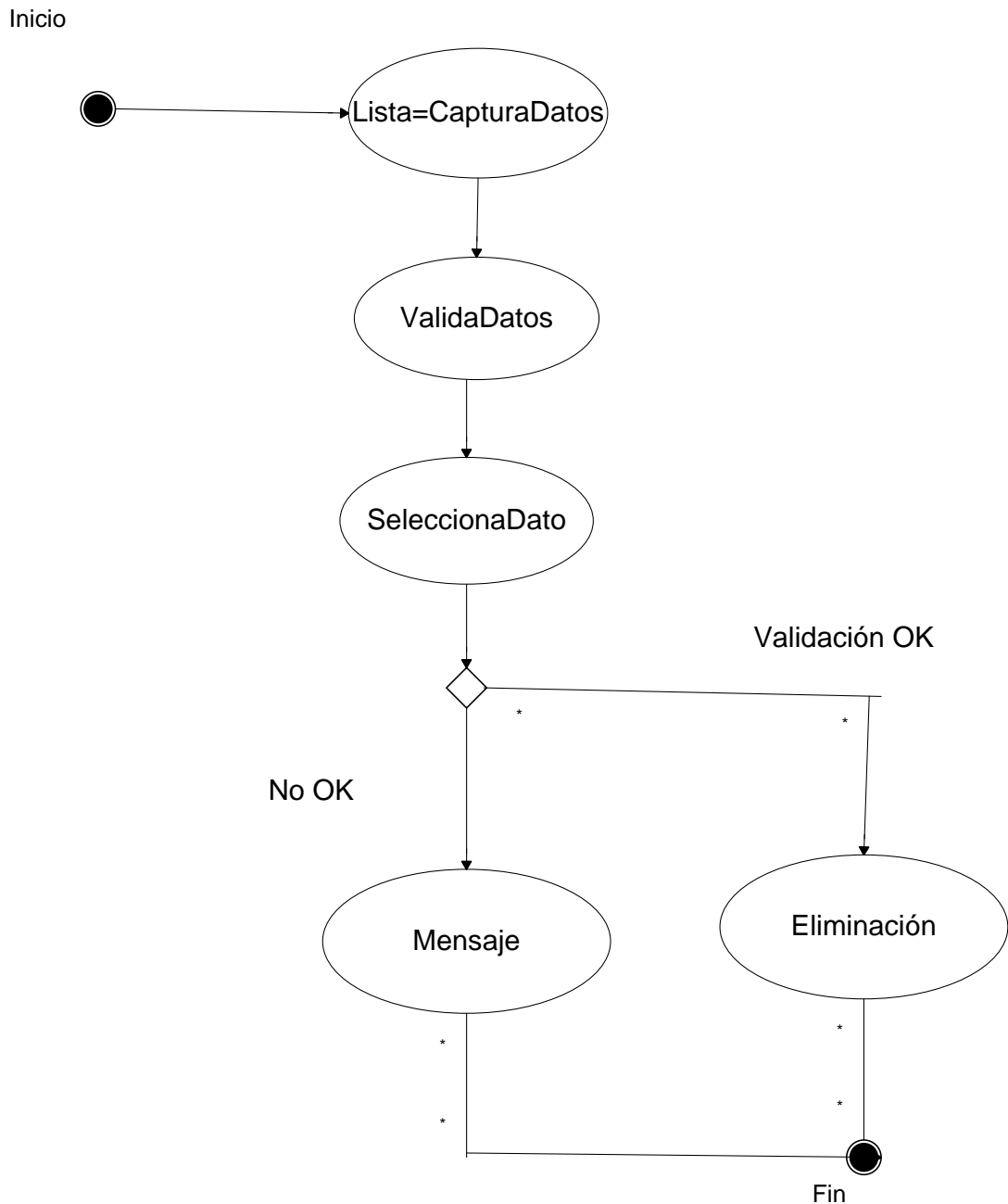
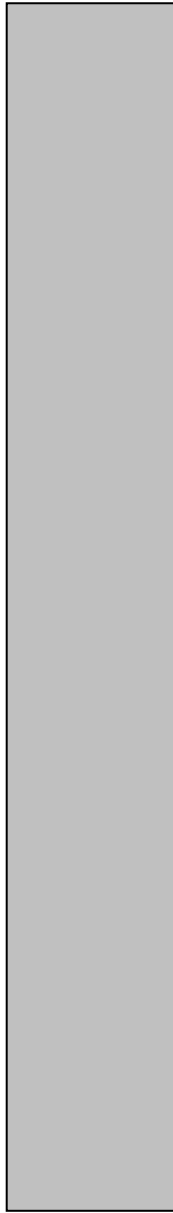


Figura. 4.28 Diagrama de Actividades eliminación datos Análisis: Fitoplancton, Control Patológico y Recepción de Larva.



CAPÍTULO 5
USO DEL PRODUCTO

5. Uso del Producto

El aplicativo Análisis de Control de Laboratorio Camaronero tiene como principal propósito ser una herramienta de apoyo de trabajo para el Biólogo, Zootecnista, Técnico Acuicultor que como responsable del manejo de los procesos y análisis que son realizados para una correcta producción y manejo del camarón desde su etapa larvaria cuando llega como semilla para ser introducido en las piscinas, para luego pasar a las piscinas en donde permanecerá hasta su extracción para luego ir a la empacadora.

En todas estas etapas se necesitan hacer análisis para ver como el camarón se va desarrollando en aspectos como tamaño, peso, parásitos que afectan su fisiología y por ende a la producción, que si no se corrigen a tiempo darían al traste con una inversión cuantiosa en lo económico y de tiempo.

Nuestra aplicación dará a toda esa información provenientes de los análisis fundamentales tales como Análisis de Fitoplancton que busca conocer básicamente el numero de algas y microorganismos (fitoplancton) que son de los que se alimenta el camarón en el manejo semi-intensivo (alimentación por balanceados y microorganismos que se encuentran en el agua estuarina que es bombeada a la piscina) el análisis de control patológico cuyo objetivo es hacer un seguimiento del estado de sanitario del camarón viendo aspectos como tamaño, peso, revisión de órganos internos en busca de parásitos endémicos que podrían estar afectando su crecimiento y por último el análisis recepción de larva que busca prevenir que la semilla (larva) que proviene de los diversos laboratorios venga en buen estado para iniciar la producción todos procesos manejados a través de sistema informático que busca dar a conocer la información a todo aquel que desee conocerla y cuente con acceso a internet muy en especial a departamentos tales como producción, y de gerencia que toman decisiones basados en información técnica proveniente de dichos análisis.

5.1 Pantalla Seguridad del Sistema

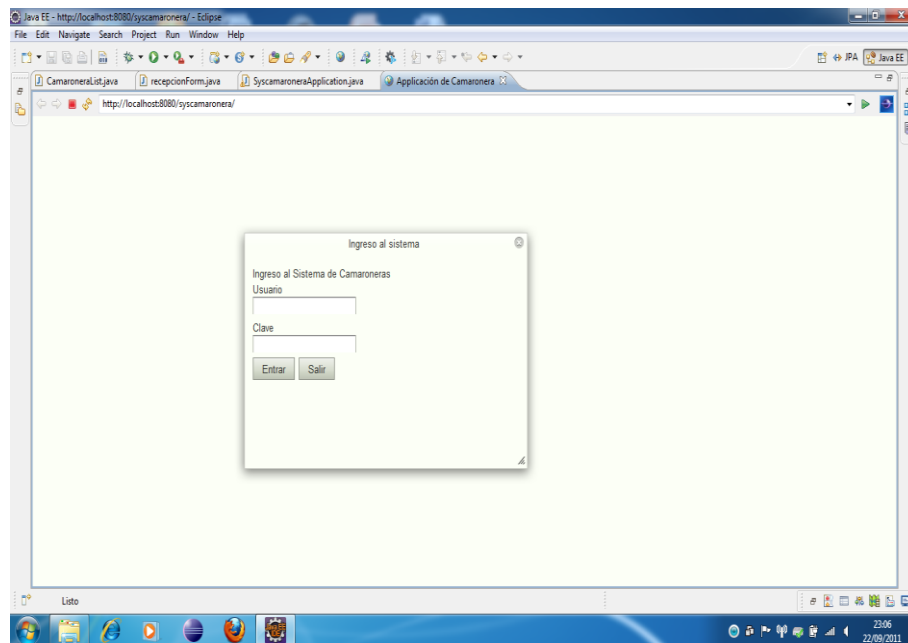


Figura. 5.1 Seguridad

Permite la entrada a la aplicación ingresando UserName: admin y PassWord: 123

5.2 Pantalla Principal del Sistema

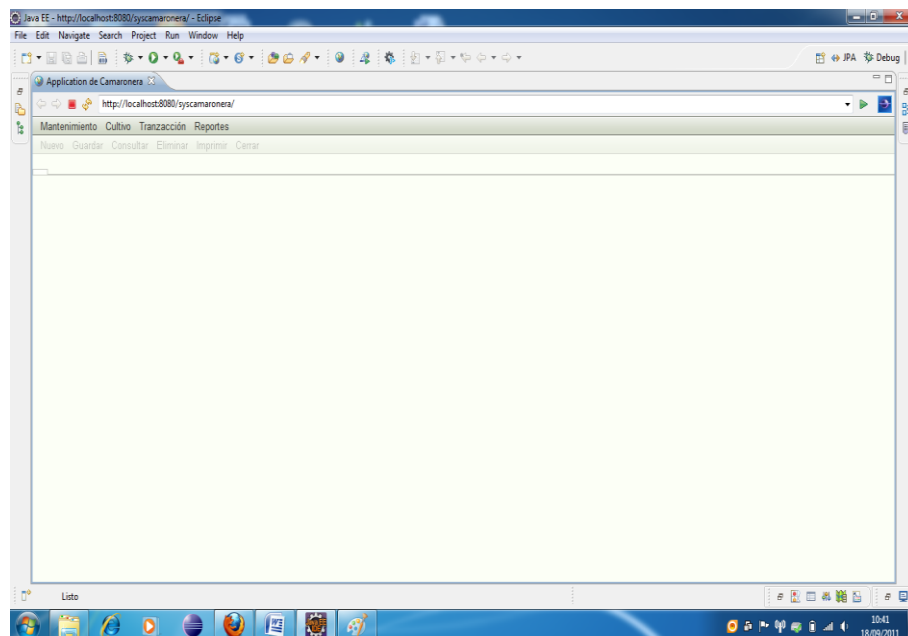


Figura. 5.2 Menú Principal

Cuenta con un menú principal con las opciones Mantenimiento, Cultivo, Transacción y Reportes

5.3 Pantalla Principal opción Mantenimiento

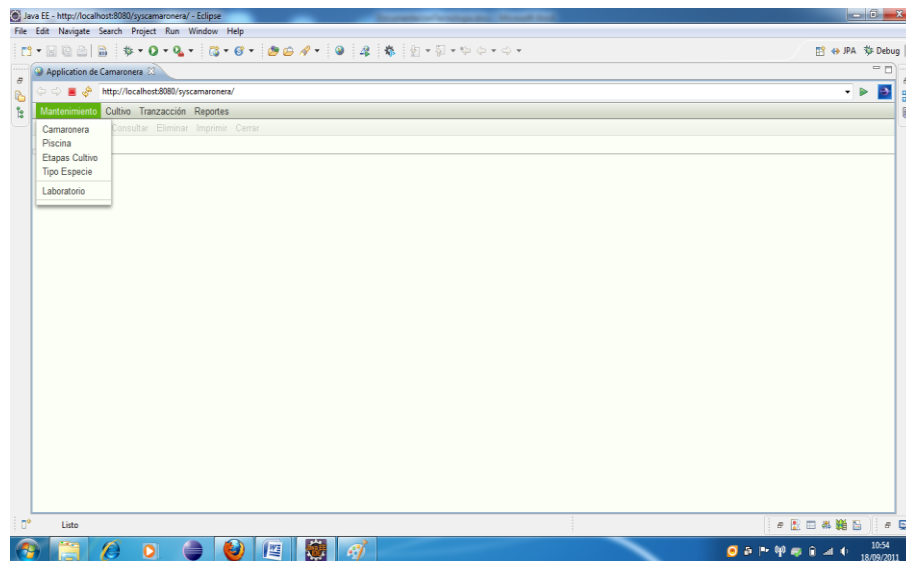


Figura. 5.3 Opción Mantenimiento

Cuenta con las subopciones Camaronera, Piscina, Etapa Cultivo, Tipo Especie.

5.4 Pantalla Camaronera: Subopción Mantenimiento del menú Principal del sistema

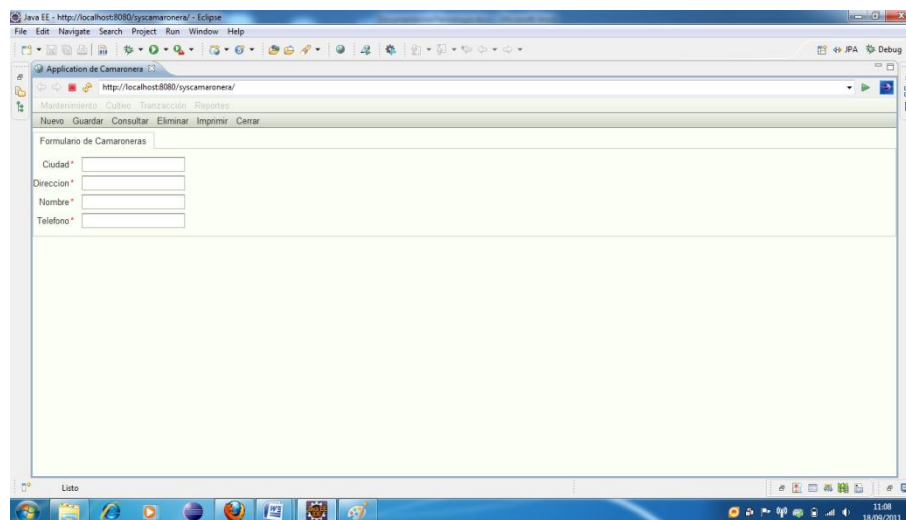


Figura. 5.4 Formulario Camaronera

Se registrará la información general de los datos de la camaronera.

5.5 Pantalla Piscina: Subopción Mantenimiento del menú Principal del Sistema

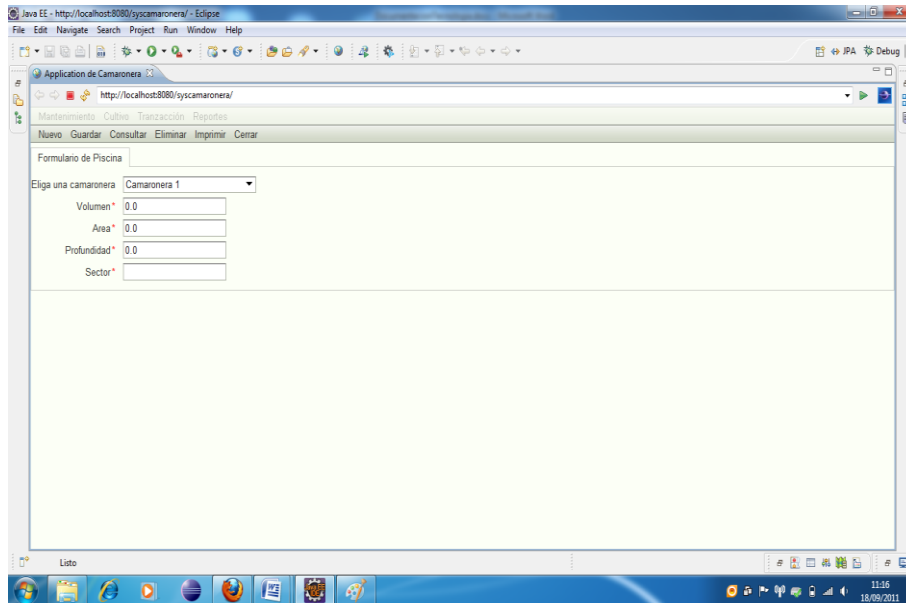


Figura. 5.5 Formulario Piscina

Registrará la información de las dimensiones y capacidad de las Piscinas

5.6 Pantalla Etapas Cultivo: Subopción Mantenimiento del menú Principal

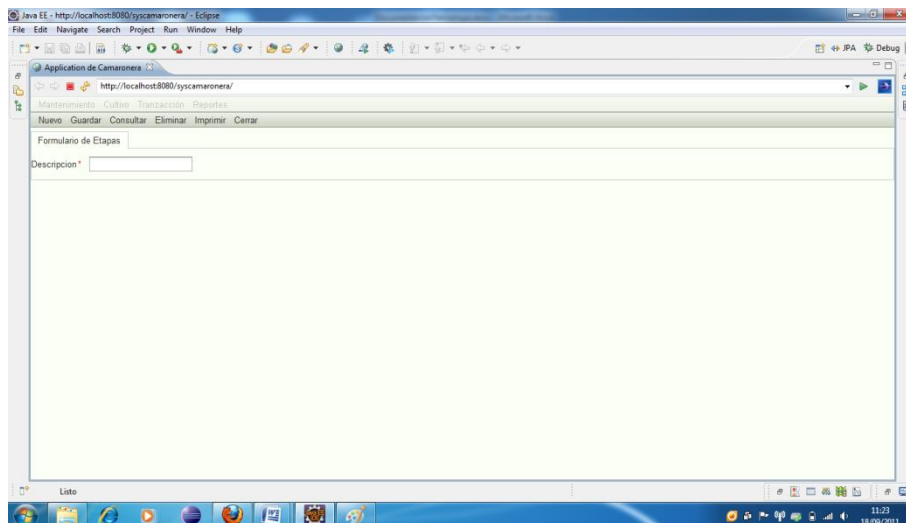


Figura. 5.6 Formulario de Etapas

Realizará una descripción de las etapas de desarrollo en que se encuentra la explotación.

5.7 Pantalla Tipo Especie: Subopción Mantenimiento del menú Principal

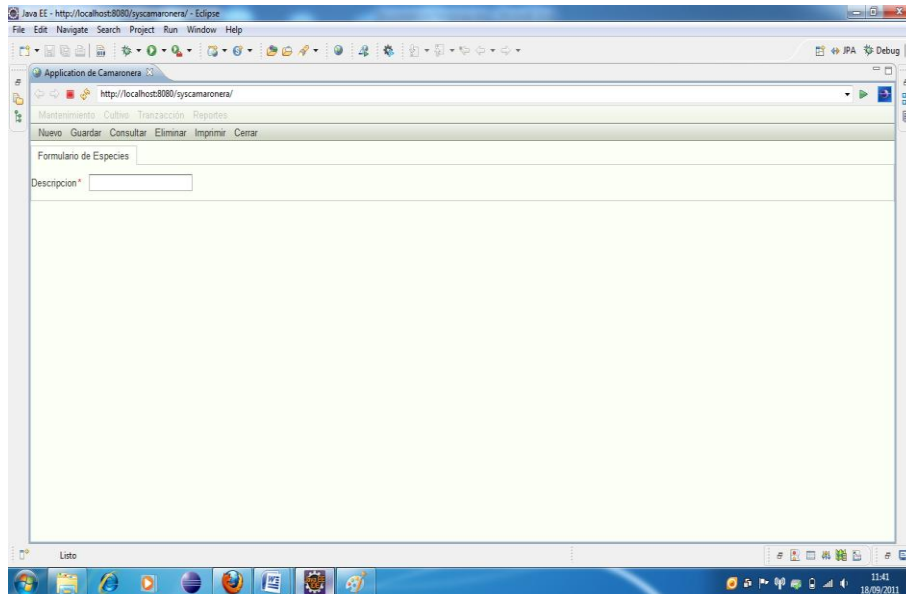


Figura. 5.7 Formulario de Especies

Por medio de esta, se Realizará una descripción del tipo de especie que se haya decidido explotar ejemplo especie Vannamei del camarón.

5.8 Pantalla Laboratorio: Subopción Mantenimiento del menú Principal del Sistema

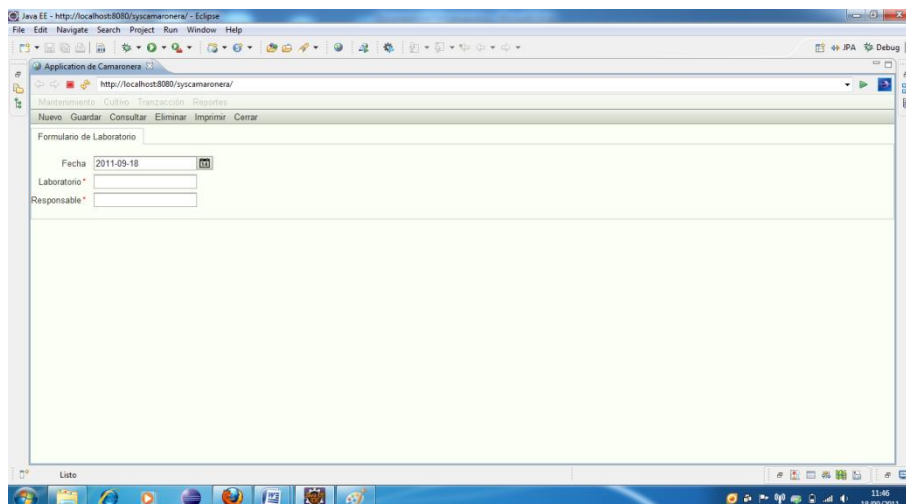


Figura. 5.8 Formulario Laboratorio

Se Registrará la información de los distintos laboratorios de donde procede la larva de cultivo.

5.9 Pantalla Principal del Sistema: Opción Cultivo del Menú Principal

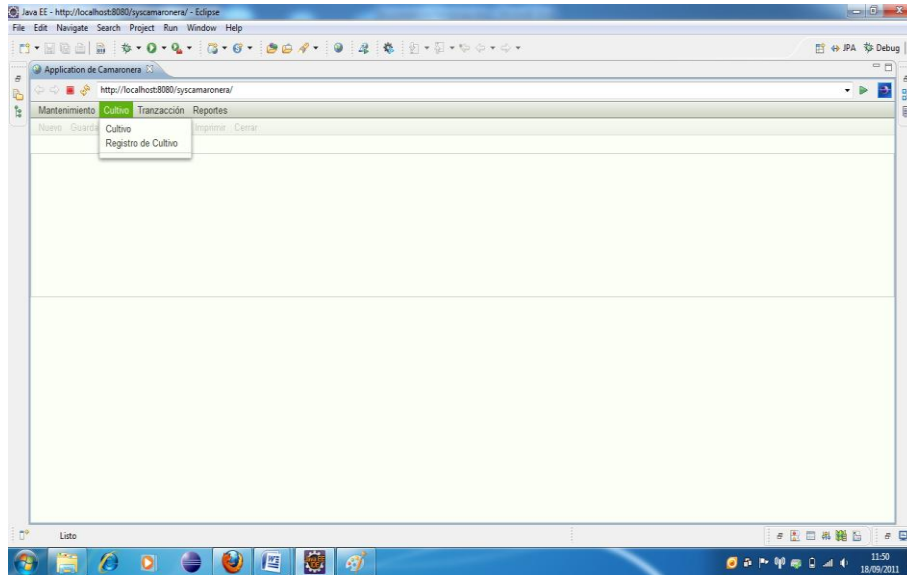


Figura. 5.9 Opción principal Cultivo

Opción del menú principal contiene las subopciones Cultivo y Registro de Cultivo

5.10 Pantalla Cultivo: Subopción Cultivo del menú principal del sistema

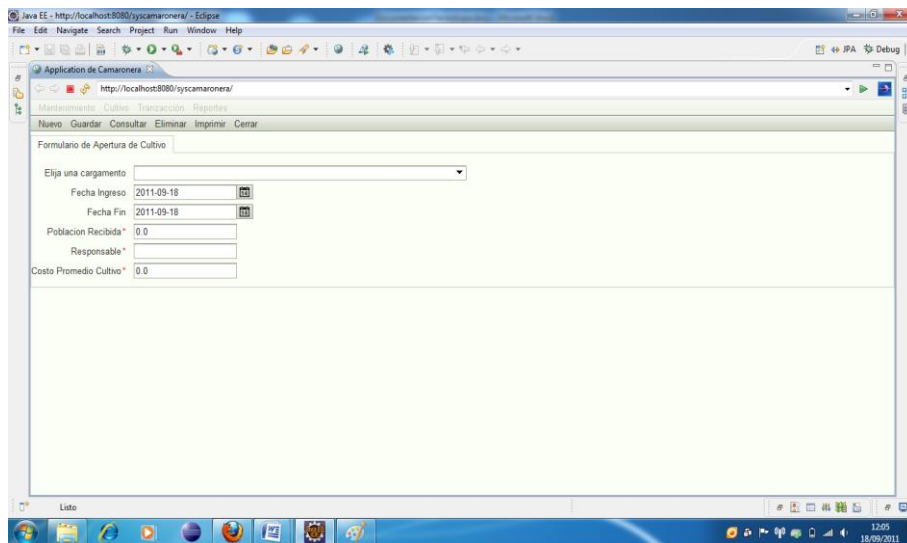


Figura. 5.10 Formulario apertura de Cultivo

Por su intermedio se registrará la información específica del cultivo a explotar

5.11 Pantalla Registro de Cultivo: Subopción Cultivo del menú Principal del sistema

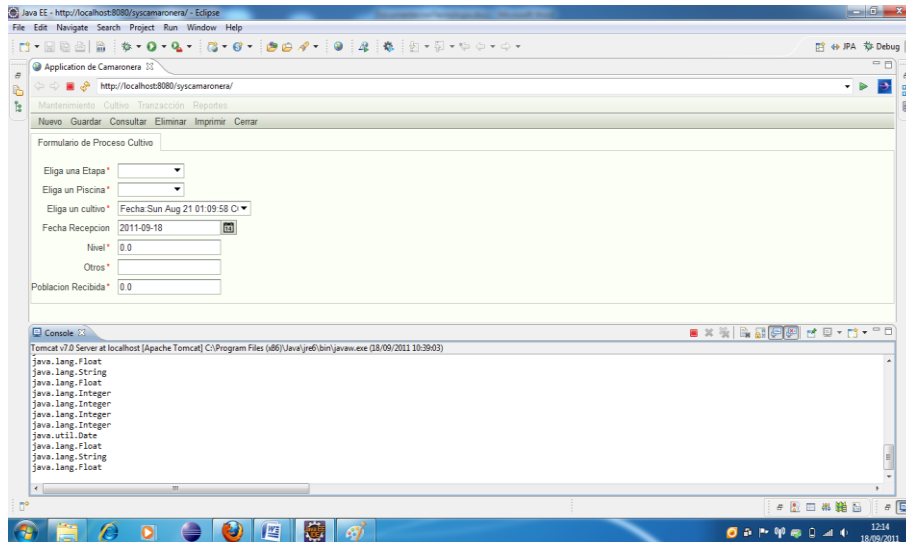


Figura. 5.11 Formulario Proceso de Cultivo

Registrará la información de la etapa y datos específicos en que se encuentra el cultivo a explotar

5.12 Pantalla Principal del Sistema: Opción Transacción del menú principal

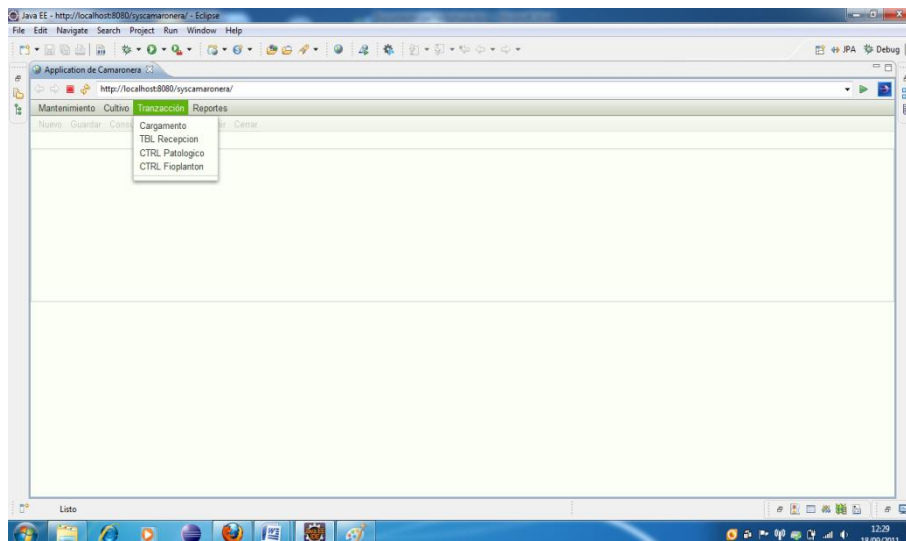


Figura. 5.12 Opción principal Transacción

Opción Transacción del menú principal contiene las subopciones cargamento, Análisis de Fitoplancton, Análisis Control Patológico, Análisis Recepción de Larva.

5.13 Pantalla Cargamento: Subopción Transacción del menú principal sistema

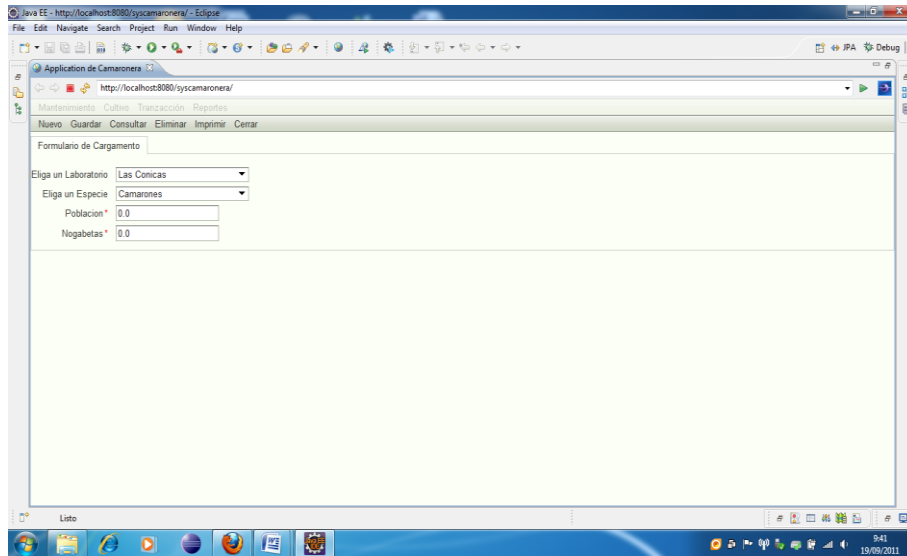


Figura. 5 13 Formulario Cargamento

Mediante esta se Registrará de la información de los diferentes cargamentos de donde proviene la larva para cultivo en piscina

5.14 Pantalla Análisis Recepción de Larva: Subopción Transacción del menú principal del sistema

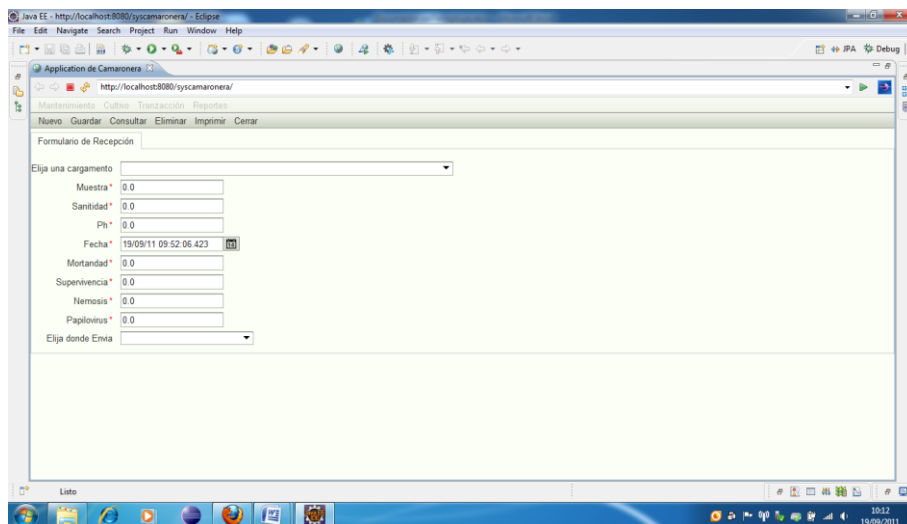


Figura. 5.14 Formulario de Recepción

Realizará el proceso y registro de información de datos del Análisis de Recepción de larva.

5.15 Pantalla Análisis Control Patológico: Subopción Transacción del menú principal del sistema.

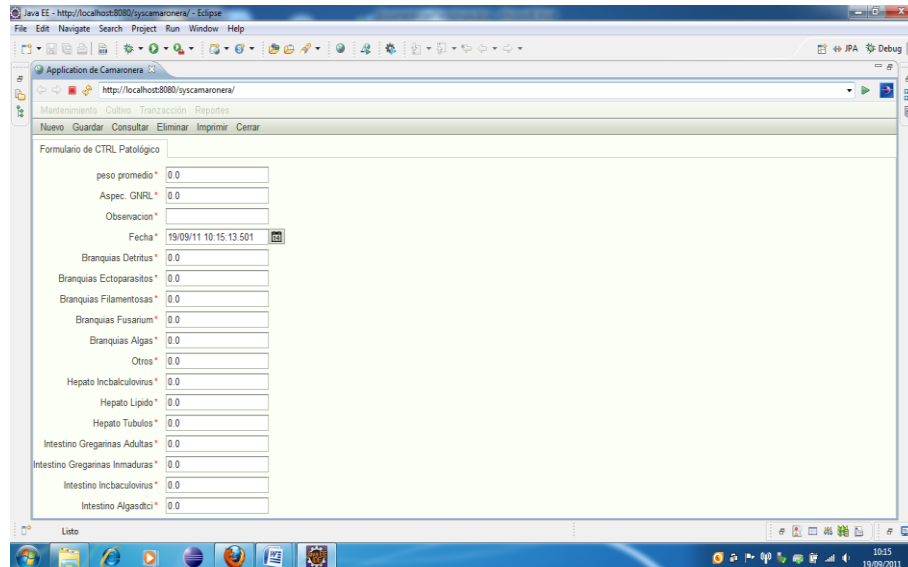


Figura. 5.15 Formulario de control Patológico

Por medio de esta se realizará el registro de información de Análisis de control Patológico.

5.16 Pantalla Análisis de Fitoplancton: Subopción Transacción del menú principal del sistema

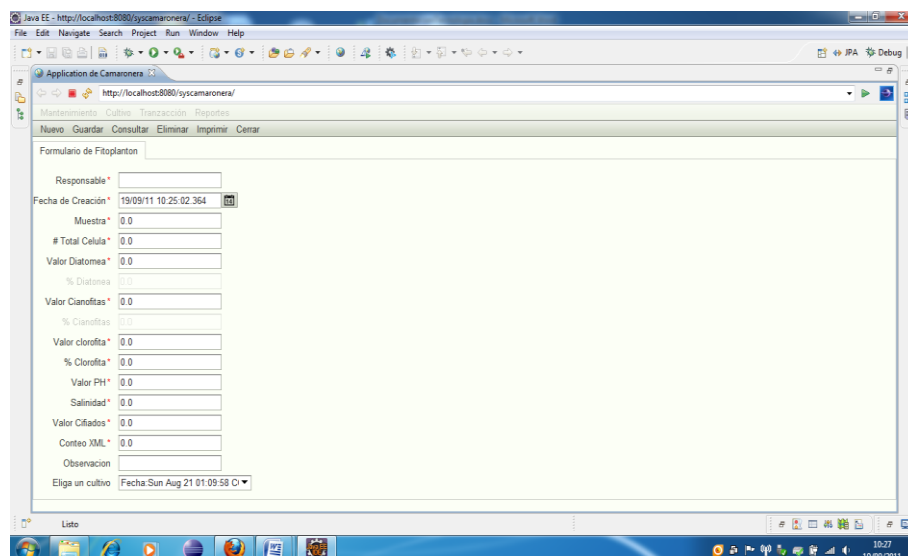


Figura. 5.16 Formulario Análisis de Fitoplancton

Procesará y registrará la información del Análisis de Fitoplancton

5.17 Pantalla Consultar: Análisis de Fitoplancton, Control Patológico y Recepción de Larva

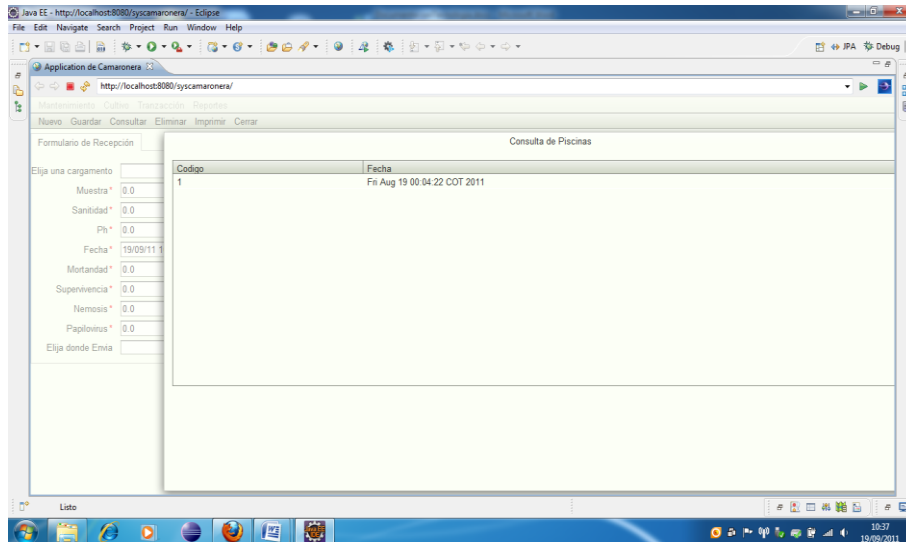


Figura. 5.17 Pantalla Consultar

Esta pantalla es la misma para los tres tipos Análisis de la Aplicación como son: Análisis de Fitoplancton, Análisis de Control Patológico y Análisis de Recepción de Larva.

5.18 Pantalla Reporte Control Patológico: opción Imprimir del Formulario Control Patológico

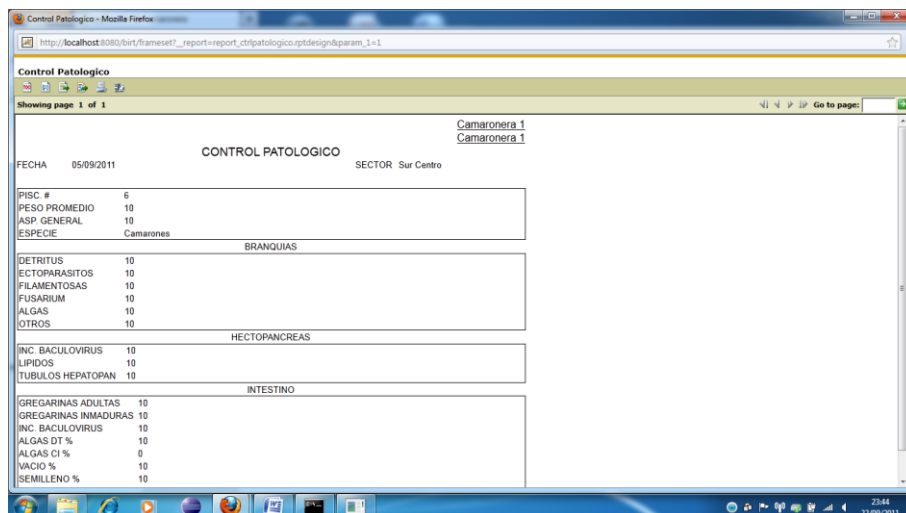


Figura. 5.18 Reporte de Control Patológico

Presentará la información del Análisis de Control Patológico dado por su formulario

5.19 Pantalla Reporte Análisis de Fitoplancton: opción Imprimir del Formulario Análisis de Fitoplancton

BIRT Report Viewer

Showing page 1 of 1

Camaronera 1
Camaronera 1

4 de septiembre de 2011

SECTOR Sur Centro
N° 1

N° Celula	10	Salinidad	10
Diatonea	10	Cifados	10
% Diatonea	10	Conteo XML	10
Cianofita	10		
% Cianofita	10		
Clorofita	10		
% Clorofita	10		
PH	10		
OBSERVACION	10		

22/09/2011 23:51

Figura. 5.19 Reporte Análisis de Fitoplancton

Visualizará la información del Análisis de Fitoplancton dado por su formulario de origen.

5.20 Pantalla Reporte Análisis de Recepción de Larva: opción Imprimir del Formulario Análisis Recepción de Larva

BIRT Report Viewer

Showing page 1 of 1

CONTROL DE RECEPCION

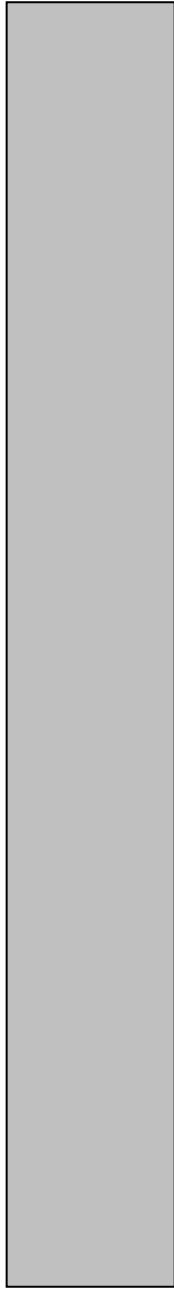
Laboratorio Las Conicas
Fecha 19/08/2011

Codigo	1	Nemosis	10
Muestra	10	PapiloVirus	10
Salinidad	10	Polibación	10
PH	10	Supervivencia	10
Mortandad	10	% Supervivencia	0
% Mortandad	0	Especie	Camarones

03/10/2011 13:10

Figura. 5.20 Reporte Análisis de Recepción de Larva

Se mostrará la información del Análisis de Recepción Larva dado por su formulario de origen



CAPÍTULO 6
ESPECIFICACIONES
TÉCNICAS

6. Especificaciones Técnicas

Como instalar el sistema.

Requisitos:

1. Descargar e Instalar Eclipse Indigo
<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/indigo/R/eclipse-jee-indigo-win32.zip>
2. Descargar e Instalar Apache Tomcat 7.0 <http://tomcat.apache.org/download-70.cgi>
3. Descargar e Instalar BIRT-RPC
<http://www.eclipse.org/downloads/packages/node/609>
4. Descargar e Instalar MYSQL <http://www.mysql.com/downloads/installer/>
5. Descargar e Instalar Java SDK y Java JRT
<http://java.com/es/download/manual.jsp>

Instalación

Abrir el eclipse configurar los accesos como el

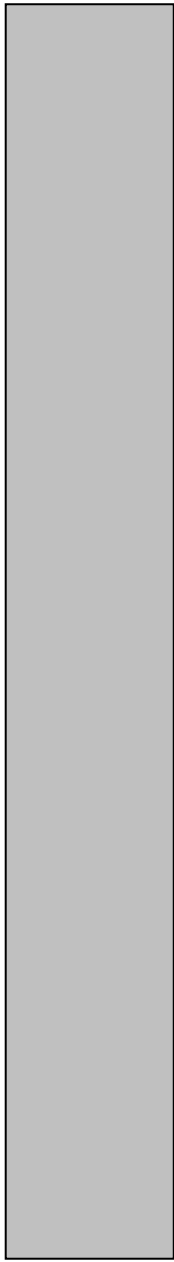
Configurar Base de datos, Subir el backup,
<http://dev.mysql.com/doc/administrator/en/mysql-administrator-restore-restore-content.html>

Copiar la Carpeta del sistema (desarrollo) en el directorio raíz

Configure el archivo syscamaronera\syscamaronera\src\META-INF\ persistence.xml
acceso a base de datos

Levante el eclipse y busque la carpeta syscamaronera este es su workspace

Configure Tomcat en eclipse <http://www.banym.de/eclipse/install-tomcat-with-eclipse>

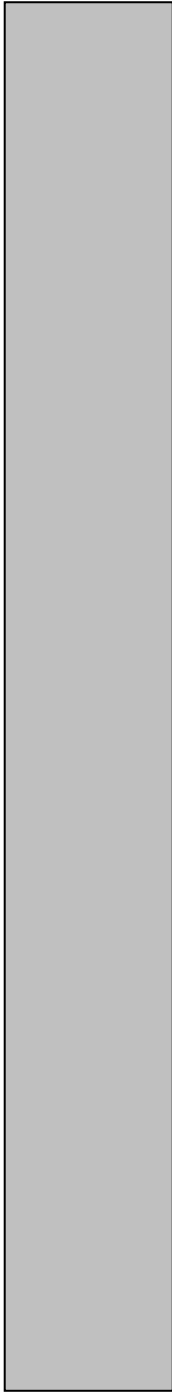


CAPÍTULO 7
SUGERENCIAS Y
RECOMENDACIONES

7. Sugerencias y Recomendaciones

El programa Análisis de control de Laboratorio Camaronero busca primordialmente dar un manejo sistemático de la información en este caso de los análisis que son tomados en el laboratorio de una camaronera, por el momento son solo tres, lo mas importantes pero existe otros que se podrán ir agregando con el paso de tiempo y la consolidación en una primera etapa de estos tres primeros análisis.

Este tipo de herramientas open source son la mejor forma de poner en marcha prototipos de esta clase en sectores de la producción (Pecuarios) donde es incipiente todavía su uso, así que hay un campo extenso por desarrollar aun.



CAPÍTULO 8

ANEXOS

8. Anexos:

8.1 Configuración del JPA

La entidad, Embeddable y MappedSuperclass

- Estas clases deben decorarse con las anotaciones necesarias, y/o
- Se defina en el orm.xml (o algún otro archivo de la cartografía) usando el descriptors de XML.

Las propiedades - use estos persistence.xml o propiedades de unidad de persistencia a extenso configure su aplicación de JPA.

- "javax.persistence.transactionType" - Standard JPA PersistenceUnitTransactionType property, "JTA" or "RESOURCE_LOCAL".
- "javax.persistence.jtaDataSource" - Standard JPA JTA DataSource name.
- "javax.persistence.nonJtaDataSource" - Standard JPA non-JTA DataSource name.
- "javax.persistence.jdbc.driver" - Standard JPA 2.0 JDBC driver class name for JSE deployments (was "eclipselink.jdbc.driver" in EclipseLink 1.1).
- "javax.persistence.jdbc.url" - Standard JPA 2.0 JDBC URL for JSE deployments (was "eclipselink.jdbc.url" in EclipseLink 1.1).
- "javax.persistence.jdbc.user" - Standard JPA 2.0 database user for JSE deployments (was "eclipselink.jdbc.user" in EclipseLink 1.1).
- "javax.persistence.jdbc.password" - Standard JPA 2.0 database password for JSE deployments (was "eclipselink.jdbc.password" in EclipseLink 1.1).

8.2 Instalando y Configurando EclipseLink

Este documento incluye la instalación e instrucciones de la configuración para:

- EclipseLink (including EclipseLink Workbench)
- EclipseLink with OSGi

8.3 EclipseLink instalando (JavaSE & EE)

Use los procedimientos siguientes para instalar EclipseLink para Java SE y desarrollo EE. Se proporcionan instrucciones de la instalación para el desarrollo de OSGi en otra sección. Antes de que usted proceda con el instale, se recomienda que usted deba apoyar los datos de todo el proyecto existente.

8.4 Las Variables de Ambiente de Sistema

Usted debe poner las variables de ambiente de sistema siguientes antes de instalar EclipseLink:

- 1- JAVA_HOME - JAVA_HOME Fijo a dónde usted instaló su Java el SDK casa directorio. Por ejemplo:
El ejemplo de Windows: JAVA_HOME = C:\JDK
El ejemplo de UNIX: JAVA_HOME =...: / el usr/java/jdk
- 2- El CAMINO (PATH) - el CAMINO (PATH) Fijo para incluir el directorio de JDK/bin. Por ejemplo:
El ejemplo de Windows: PATH = C:\JDK\bin
El ejemplo de UNIX: PATH =...: / el usr/java/jdk/bin

La nota: Asegure que una Java la Máquina Virtual (JVM) compatible con JDK 1.5 (o superior) también se instala en su máquina.

8.5 La instalación

- 1- Transmite los EclipseLink instalan el archivo zip de archivo del EclipseLink transmite la esta página <http://www.eclipse.org/eclipseLink/downloads>. El archivo zip también incluye Banco de trabajo de EclipseLink, una interfaz gráfica que le permite configurar el descriptors y trazar sus proyectos. Es una herramienta poderosa si usted está emigrando de Oracle TopLink con clásico o el uso de ORM/OXM/EIS nativo.
- 2- Desempaquete el archivo zip transmitido en el directorio de la instalación deseado. Cuando usted abre la cremallera el archivo, usted encontrará un subdirectorio del eclipselink, con el subdirectorios múltiple. Este directorio es su nuevo directorio de ECLIPSELINK_HOME. Por ejemplo:
Por ejemplo de Windows:
ECLIPSELINK_HOME=<INSTALL_DIR>/eclipselink
Por ejemplo de UNIX:
ECLIPSELINK_HOME =...: / el usr/el/<INSTALL_DIR>/eclipselink
- 3- Cuando abrió la cremallera, se exigen los pasos adicionales ejecutar el Banco de trabajo de EclipseLink.

Bibliografía

Tecnologías Java2EE de Juan José Morroño Sánchez Plan de Formación PAS 2009
Universidad de Murcia

Java Persistence API JPA Manuel Vega Ulloa (Exposición)

Tutorial básico de Java EE de Abraham Otero Abraham@javahispano.org

Manual Hibernate de Héctor Suarez González

Manual Avanzado JPA por Carmelo Navarro Serna

Book of Vaadin: 4th Edition Vaadin Ltd
Marko Grönroos Vaadin Framework 6.7.0