

**Universidad de las Ciencias Informáticas**  
**Facultad 7**



Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

***Título: Diseño del módulo Vectores del sistema  
Control Sanitario Internacional.***

Autoras: Karelys Cairo Orozco

Jeney del Campo García

Tutora: Ing. Yunaysy Ortiz Batista

Asesor: Msc. Jorge Castillo Maceo

Ciudad de La Habana, Julio 2008

"Año 50 de la Revolución"

## ***DECLARACIÓN DE AUTORÍA***

Declaramos ser autoras de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 2 días del mes de Julio del año 2008.

Karelys Cairo Orozco

---

Firma de la Autora

Jeney del Campo García

---

Firma de la Autora

Yunaysy Ortiz Batista

---

Firma de la Tutora

## **AGRADECIMIENTOS**

### ***Queremos agradecer:***

*A nuestra tutora, por la ayuda y dedicación brindada.*

*A los que de una forma u otra han contribuido a nuestra educación, aunque sea con un gesto o una palabra.*

### ***De Karelys:***

*A mis padres por todo el esfuerzo que han hecho para contribuir a mi formación profesional. Los quiero, mamita y papito.*

*A mi hermano que siempre me ha apoyado y a quien quiero mucho.*

*A mi novio por su ayuda y paciencia. Gracias mi amor.*

*A todos mis profesores desde la primaria hasta la universidad.*

*A mis amigos y compañeros de grupo del preuniversitario y de la universidad.*

*A mi compañera de tesis que me ayudó en la realización del trabajo.*

*A todas las personas de mi pueblo que me han ayudado.*

### ***De Jeney:***

*A toda mi familia, incluyendo a Geidy y Armando, por su apoyo y dedicación en todos los momentos.*

*A Onay por tanto amor y comprensión.*

*A mis amistades, en especial a Liset, Yohana y Reynaldo, que me apoyaron y ayudaron siempre. Además, a mis compañeros de aula por darme la oportunidad de compartir estos cinco años tan especiales.*

*A Omarito, por ayudarme siempre y brindarme esta amistad tan bonita.*

*A mi compañera de tesis, por su dedicación, compañerismo, paciencia y comprensión.*

***A Todos: Muchas Gracias.***

## **DEDICATORIA**

### **De Jeney:**

*A mis padres y mi hermano, por ser las personas más especiales en mi vida. Por todo su sacrificio, porque su amor y cariño nunca me han faltado. Por lograr lo que soy, que no es más que el reflejo del fruto de sus propias vidas.*

*A mi familia, a mis primos que los adoro, a mis tías, mis abuelas y mis abuelos. Por añorar tanto este día, por poder contar con su presencia, por confiar en mí, y por haberme guiado durante toda mi vida.*

### **De Karelys:**

*A mis padres, los quiero mucho por dedicarse tanto a mí y ofrecerme siempre todo su amor.*

*A mi hermano, por todo el cariño que siempre me ha brindado.*

*A Douglas, por ser el sobrino más lindo del mundo.*

*A Wayner, por todo su amor y comprensión a lo largo de estos años.*

*A mi familia.*

## **RESUMEN**

La presente investigación se ha desarrollado con el objetivo principal de realizar el diseño de un sistema informático para lograr una adecuada gestión de la información referida al Programa de Vigilancia y Lucha Antivectorial en Cuba. Dirigida esencialmente al control de focos del mosquito *Aedes Aegypti*, agente transmisor de enfermedades como el dengue.

Para lograr el diseño de la aplicación se utilizó como metodología de desarrollo el Proceso Unificado de Rational (RUP) conjuntamente con el Lenguaje Unificado de Modelado (UML). Como herramienta de modelado se empleó el Visual Paradigm.

Se elaboraron los siguientes artefactos: Modelo de Negocio, Modelo de Sistema, Requerimientos funcionales y Modelo de Diseño.

El sistema tributa a la economía cubana al lograrse una reducción de los costos de inversión evitando la importación de software foráneo. Al poner en práctica la aplicación se logrará tener la información actualizada en todo momento, permitiendo a los especialistas de la campaña de vigilancia y lucha antivectorial, en todos los niveles, tener un adecuado control de los focos de mosquitos *Aedes Aegypti*, posibilitando un menor tiempo de respuesta en la toma de decisiones.

## ÍNDICE

<b>DEDICATORIA.....</b>	<b>II</b>
<b>RESUMEN.....</b>	<b>III</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....</b>	<b>6</b>
1.1 <i>Conceptos básicos relacionados con el dominio del problema.</i> .....	6
1.2 <i>Sistemas automatizados existentes vinculados al campo de acción.</i> .....	7
1.3 <i>Tendencias y tecnologías actuales.</i> .....	9
1.4 <i>Aplicación web.</i> .....	9
1.5 <i>Servidor web.</i> .....	10
1.6 <i>Navegador web.</i> .....	10
1.7 <i>Sistemas gestores de bases de datos.</i> .....	10
1.8 <i>Metodologías de software.</i> .....	11
1.9 <i>Arquitecturas de software y patrones arquitectónicos.</i> .....	17
1.10 <i>Herramientas Case.</i> .....	22
1.11 <i>Lenguajes de programación web.</i> .....	24
<b>CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>26</b>
2.1 <i>Objetivos de la organización.</i> .....	26
2.2 <i>Flujo actual de los procesos.</i> .....	27
2.3 <i>Objeto de automatización.</i> .....	28
2.4 <i>Propuesta del sistema.</i> .....	29
2.5 <i>Modelo de Negocio.</i> .....	30
2.6 <i>Especificación de los requisitos.</i> .....	37
2.7 <i>Comunicación con otros sistemas.</i> .....	39
2.8 <i>Definición de los casos de uso del sistema.</i> .....	39
<b>CAPÍTULO 3. DISEÑO DEL SISTEMA. ....</b>	<b>57</b>
3.1 <i>Modelo de diseño.</i> .....	57
<b>CONCLUSIONES.....</b>	<b>79</b>

<b>RECOMENDACIONES.....</b>	<b>80</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>81</b>
<b>BIBLIOGRAFÍA.....</b>	<b>83</b>
<b>ANEXOS .....</b>	<b>86</b>
<i>Anexo 1. Modelo de negocio.....</i>	<i>86</i>
<i>Anexo 2. Modelo de sistema.....</i>	<i>90</i>
<i>Anexo 3. Diseño del sistema. ....</i>	<i>97</i>
<b>GLOSARIO DE TÉRMINOS. ....</b>	<b>101</b>

### **INTRODUCCIÓN**

En la actualidad los cambios que se están produciendo con motivo del creciente desarrollo e implantación de las Tecnologías de la Información y las Comunicaciones (TIC) son enormes. Las TIC forman parte de la cultura tecnológica que rodea a la sociedad y con la que se debe convivir, amplían las capacidades físicas y mentales, las posibilidades de desarrollo social y permiten un fácil acceso a gran fuente de información. (1)

Brindan canales de comunicación inmediatos, facilitan la automatización de trabajos y la interactividad y digitalización de toda la información. Su desarrollo vertiginoso está introduciendo grandes avances en el campo de la informática, lo que hace necesario su aplicación en cualquier país para la automatización de los procesos complejos de realizar.

Por su gran importancia Cuba se está sumando a esta amplia gama de tecnologías, aplicándolas en todas las esferas de la sociedad, estrategia que tiene al ciudadano como centro de sus objetivos, buscando elevar la calidad de vida en su desempeño familiar, laboral, educacional, cultural, social y político.

Entre las áreas que se ha propuesto desarrollar el país, se encuentra el Sistema Nacional de Salud (SNS) que persigue, elevar la calidad del servicio de la salud, la interconectividad entre las diferentes unidades hospitalarias y crear de manera progresiva bases de conocimiento que faciliten la toma de decisiones administrativas y clínicas.

La informatización del SNS está dado por el conjunto de métodos, técnicas, procedimientos y actividades gerenciales dirigidas al manejo de la información en la salud, la cual comprende la información sobre el estado de salud de la población, la información sobre el conocimiento de las ciencias de la salud y la información en general para la toma de decisiones clínico-epidemiológicas, operativas y estratégicas. (2)

Desde los primeros años del Triunfo de la Revolución Cubana, fue una estrategia política e interés del gobierno revolucionario y el Ministerio de Salud Pública (MINSAP), el estudio y procesamiento de los hechos vitales y sanitarios, inicialmente de forma manual y después con equipos mecanizados IBM.

En años posteriores, se dispone de la Red Telemática de la Salud (INFOMED), que permite enlazar todo el sistema de salud para dar una respuesta más eficiente en el campo de la información científica. Esta red surge con una acertada visión de la influencia que las TIC ejercerán en la esfera de la

información, conocimiento y fruto de la voluntad política del país y la colaboración y solidaridad nacional e internacional.

Actualmente la informática aplicada en el área de la salud permite un mayor grado de acceso a la información unificada y confiable en tiempo real, aporta rapidez y fiabilidad necesaria para las modernas técnicas de administración, la toma de decisiones y el control de las estadísticas en los diferentes niveles, permite la realización de diagnósticos automatizados y la selección de los tratamientos adecuados.

Se espera que en todas las instituciones de salud del país se apliquen estos logros y se exploten al máximo, comenzando desde el Sistema de Atención Primaria hasta los eslabones más altos, o sea, los institutos a nivel nacional. Para la informatización del SNS se trabaja integradamente, estrechando alianzas extrasectoriales, en el progreso de un grupo de aplicaciones básicas de desarrollo tecnológico, y es precisamente esta integración la que permite hablar de informatización en el sector de la salud pública cubana.

El MINSAP ha definido a la informatización como una de sus prioridades y ha convocando para ello a un grupo de instituciones propias del sector, del Ministerio de Informática y las Comunicaciones y de otros organismos de la administración central del estado, para definir de conjunto la estrategia a desarrollar.

Se trabaja en proyectos basados en nuevas tecnologías de Internet, software libre y otras que garantizan una explotación integrada y compatible, serán introducidas utilizando como infraestructura a INFOMED. La Universidad de las Ciencias Informáticas (UCI) participa en dichos proyectos. Estos responden a diferentes necesidades de la población tanto de la salud propia del individuo como de su higiene y sanidad.

Los programas que se definan permitirán alcanzar por etapas la informatización de la salud pública cubana, al contar con la integración de los datos generados en los distintos niveles de salud donde puede ser atendido un paciente. Permitirá perfeccionar la calidad asistencial ofrecida a la sociedad, facilitar las funciones del personal de la salud y colaborar con la gestión administrativa, asistencial, docente y de investigación.

La investigación en curso está encaminada al programa Control Sanitario Internacional, que se integrará a los sistemas existentes y en desarrollo pertenecientes al Registro de Información para la Salud (RIS); sobre todo con aquellos donde se establecerá un vínculo permanente como el Registro de

Ubicación, Registro de Localidades, Registro de Unidades de Salud, y el Sistema de Autenticación, Autorización y Auditoría.

El propósito de Control Sanitario Internacional es responder a las estrategias trazadas por el MINSAP permitiendo prevenir, detectar la introducción y evitar la propagación en Cuba de enfermedades exóticas, emergentes y reemergentes y adoptar las medidas necesarias con la retroalimentación adecuada a los distintos niveles del SNS. Al mismo tiempo tiene como objetivo relatar las definiciones establecidas por los expertos para lograr un sistema que gestione de forma centralizada la información referente al control sanitario internacional.

Dentro del proyecto se ubica el subsistema Vectores con capacidades a nivel de área de salud, municipios, provincias y nación, que es el encargado de gestionar toda la información relacionada con el programa de Vigilancia y Lucha Antivectorial. Tiene la misión de contribuir a evitar la introducción y/o propagación de enfermedades transmitidas por vectores, como el dengue, para incrementar los niveles de salud y satisfacción de la población mediante la vigilancia y el control de los vectores y hospederos intermediarios. Estará centrado inicialmente en el control del mosquito *Aedes Aegypti*.

El control de los vectores que se encuentren en las diferentes zonas es realizado en las unidades de vigilancia y lucha antivectorial de las áreas de salud. El análisis general de los datos se efectúa en los diferentes niveles del SNS bajo la dirección del Viceministerio de Higiene y Epidemiología que radica en Ciudad de La Habana.

En esta sede, la información es recibida por diferentes vías: entrega personal, por correo electrónico, teléfono u otros medios. Debido a que la transferencia de los datos de un nivel (municipal, provincial o nacional) a otro es realizada en algunos casos de forma manual y en otras ocasiones por medios poco seguros y lentos, se retrasa su recogida. Como consecuencia puede ocurrir la introducción de errores humanos, que no se proporcione la ubicación rápida y exacta de los focos de mosquitos y que se retrasen los reportes e información estadística.

Debido a esta situación anteriormente mencionada se ha decidido, por parte de la dirección del país, del Viceministerio de Higiene y Epidemiología y de los expertos en el tema, que se le de una solución adecuada mediante la informatización de todos los procesos.

Dada esta situación problemática se plantea el siguiente *problema*: ¿Cómo facilitar la gestión de la información relacionada con el control de focos de mosquitos *Aedes Aegypti* en Cuba?

El *objeto de estudio* de la investigación es el proceso de gestión de la información referente al control sanitario internacional en Cuba.

El *campo de acción* es el proceso de gestión de la información relacionada con el control de focos de mosquitos *Aedes Aegypti* en Cuba.

El *objetivo general* es diseñar una aplicación web que permita viabilizar la gestión de la información relacionada con el control de focos de mosquitos *Aedes Aegypti* en Cuba.

Para dar cumplimiento al objetivo propuesto y para lograr el adecuado desarrollo del trabajo, se definieron las siguientes *tareas*:

1. Analizar los sistemas informáticos existentes.
2. Identificar las necesidades de funcionamiento del sistema informático.
3. Analizar la integración con otros componentes ya existentes en el Sistema de Información para la Salud (SISalud).
4. Elaborar el Modelo de Diseño.

En vista de obtener los datos necesarios para el diseño de la aplicación se realizará una entrevista con los clientes para su obtención, o sea, información referente a la lucha antivectorial, datos de entrada, datos de salida y características de los procesos.

Una vez puesta en práctica la aplicación web se logrará el intercambio entre los especialistas y el aumento de la calidad de la asistencia sanitaria a las diferentes zonas en el Programa de Vigilancia y Lucha Antivectorial para beneficio de toda la población cubana.

El presente trabajo de diploma está conformado por tres capítulos, de la siguiente forma:

El *primer capítulo* expone la fundamentación teórica, se analiza el objeto de estudio y el campo de acción. Se presentan los conceptos básicos relacionados con el dominio del problema y se muestra el estado del arte. Se realiza una valoración de las tendencias, tecnologías y herramientas actuales utilizadas en el desarrollo de sistemas informáticos, en específico del software a diseñar en la investigación.

El *segundo capítulo* plantea los objetivos de la organización y explica el flujo actual de los procesos realizados dentro de la campaña antivectorial, dando así cumplimiento al modelado del negocio. Se

plantea la propuesta del sistema teniendo en cuenta la especificación de los requisitos de software y se obtiene el modelo de sistema.

En el *tercer capítulo* se hace una explicación de los patrones de diseño que se utilizan. Se muestra el modelo de diseño y dentro de este los diagramas de clases. Se realizan los diagramas de interacción que representan cada escenario dentro de los casos de uso del sistema.

**CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.**

En este primer capítulo se plasman los principales conceptos relacionados con el campo de acción. Se investigan los sistemas semejantes al que se está diseñando tanto a nivel nacional como internacional y se realiza una valoración explicando por qué no se emplean para resolver los problemas existentes.

Se realiza un análisis de las tendencias y tecnologías actuales utilizadas en el desarrollo de sistemas informáticos, las metodologías de software, arquitecturas, patrones y herramientas, mostrando por qué se utilizan cada una de ellas.

**1.1 Conceptos básicos relacionados con el dominio del problema.**

Los vectores son un organismo vivo involucrado en la transmisión de agentes patógenos causantes de importantes enfermedades debido al daño que ocasionan. Son capaces de transmitir un agente infeccioso desde su reservorio a otro ser vivo, en forma activa, imprimiendo un determinado sentido a la transmisión de acuerdo a sus características.

Los insectos actúan como vectores, entre estos los mosquitos, pertenecientes a la familia Culícida. Son hospederos de una amplia variedad de agentes patógenos y parásitos, incluyendo virus, bacterias, protistas y nematodos que causan varias enfermedades a vertebrados no artrópodos incluyendo los humanos.

Entre los mosquitos más dañinos se encuentra el *Aedes Aegypti* de origen africano, inició hace siglos una dispersión acompañando los viajes del hombre a través del mundo. Eficaz vector de arbovirosis que origina una de las grandes problemáticas de salud pública mundial, con alta morbilidad capaz de bloquear las actividades de ciudades y países en picos epidémicos. Es un ejemplo de adaptación de una especie de mosquito al ámbito humano, con criaderos, hábitat, fuente de alimentación, desplazamientos activos y pasivos ligados al ámbito domiciliario. (3)

Cuba por su ubicación geográfica y características climatológicas es afectada por este insecto. Desde 1881 el sabio cubano Carlos J. Finlay declaró a esta especie de mosquito como el transmisor de la fiebre amarilla. En los primeros años del siglo XX se comprobó su papel en la transmisión del dengue, considerado como la arbovirosis más importante en el mundo. Esta es una enfermedad viral aguda que puede afectar a personas de cualquier edad, siendo más susceptibles los niños y las personas mayores.

Las acciones del Programa Nacional de Erradicación del *Aedes Aegypti* son múltiples y todas generan informaciones importantes para mantener índices de infestación que no constituyan riesgo de transmisión del dengue en Cuba, hasta lograr su erradicación.

La vigilancia y lucha antivectorial desarrollada es un programa dirigido a combatir el causante de esta enfermedad, es un desafío para el control y la vigilancia epidemiológica. Requiere cada día mayor participación de los diferentes sectores de la economía del país, la comunidad y la conducción científico-técnica especializada del sector de la salud.

La esencia está en efectuar una serie de acciones como son: la recogida de basura, la eliminación de criaderos, el tratamiento adulticida, la recuperación de las acciones en las viviendas cerradas, el control de la calidad y la constante divulgación del daño que pueda causar este culicido.

El programa exige una recogida de información que facilita el trabajo de los especialistas que deben monitorear todos los eventos relacionados con dengue y otras enfermedades, con el fin de prevenir epidemias a partir del dominio de condiciones de riesgos y del conocimiento de indicadores más susceptibles como parte de control del *Aedes Aegypti*.

### ***1.2 Sistemas automatizados existentes vinculados al campo de acción.***

La introducción de software bioestadísticos en la evaluación de los factores de riesgo asociados a todos los eventos de la vida cotidiana han revolucionado de forma ascendente desde el punto de vista metodológico y operacional, en la actualidad forman parte de casi todos los eventos relacionados con la vida humana.

Al culminar una búsqueda minuciosa correspondiente a los sistemas automatizados que se han desarrollado con respecto al tema de los vectores y la lucha antivectorial a nivel nacional e internacional, solo se encontraron algunos existentes en Cuba.

#### **SISVLA**

Este sistema fue creado por el licenciado Jorge Luis Duverger Goyanes, informático de la Unidad Provincial de Vigilancia y Lucha Antivectorial (UPVLA) de Guantánamo. Se implementó con lenguaje Delphi y en plataforma Windows.

Se creó para la actividad de vigilancia y el control del *Aedes Aegypti* a nivel de provincia y municipio, aunque por las condiciones técnicas no se pudo municipalizar. Experimentalmente hubo otras provincias que lo utilizaron, aunque no con sistematicidad.

La principal funcionalidad de esta aplicación consistía en un sistema de alerta ante índices elevados; así como el registro y cálculo automatizado de indicadores de la campaña.

### ***Proyecto de Automatización de Guantánamo***

Fue realizado por el mismo especialista y con similar uso que el anterior. Elaborado en Microsoft Excel y en la actualidad se encuentra en uso.

### ***VAEDA***

Estuvo enfocado a la vigilancia y algunas acciones de control del *Aedes Aegypti*. Creado por el técnico Leonardo Miguel Alfonso López, informático de la UPVLA de Matanzas.

### ***VIGENT***

Desarrollado por el técnico mencionado en el software anterior, estuvo encaminado a la vigilancia entomológica en general.

### ***APOLO***

Fue destinado al trabajo de control de vectores en polos e instalaciones turísticas, partiendo del hecho que en Matanzas radica el principal polo turístico del país, desarrollado también por Leonardo Miguel. Todos estos sistemas contaron con varias versiones siempre en plataforma Windows y utilizando Visual Basic como lenguaje.

### ***Control de Culícidos***

Como bien indica su nombre, es para el control de culícidos, elaborado por el entonces estudiante de programación Jucris Pujol Iglesias, hecho en plataforma Windows y lenguaje C. Este se trabajó a modo de pilotaje en la Unidad Municipal de Vigilancia y Lucha Antivectorial (UMVLA) de Marianao, con buenos resultados. Calculaba índices a nivel de manzanas, área de salud y municipio para un período de tiempo determinado.

La gran ventaja de todos estos sistemas fue el rápido procesamiento de información para la toma de decisiones. Entre las causas que obstaculizaron su uso se encuentran: la falta de equipo y la poca costumbre de uso de estas necesarias herramientas, lo que limitó su aceptación práctica. Estos en su mayoría datan de principios del 2000 y la ideología no era igual, en la actualidad la situación y la proyección son diferentes.

En sentido general, el gran problema de estas aplicaciones fue su generalización, al final quedaron usándose con sistematicidad exclusivamente en el mismo lugar donde fueron creadas.

Se decidió implementar un nuevo sistema informático porque los mencionados con anterioridad no ofrecen una solución adecuada a los problemas existentes vinculados al campo de acción. Estos productos no poseen una interfaz adecuada y no están desarrollados con las tecnologías que se requieren, carecen de soporte y están realizados con pequeñas bases de datos que no cubren la capacidad de almacenamiento de los centros del país.

### **1.3 Tendencias y tecnologías actuales.**

Se puede definir la tecnología como el conjunto de conocimientos y técnicas que, aplicados de forma lógica y ordenada, permiten al ser humano modificar su entorno material o virtual para satisfacer sus necesidades. Es un proceso combinado de pensamiento y acción con la finalidad de crear soluciones útiles. (4)

Se deben tener en cuenta la técnica, la ciencia y los aspectos económicos, sociales y culturales involucrados, que deben responder a necesidades o deseos de la sociedad y tener como propósito contribuir a mejorar la calidad de vida.

Para desarrollar un software con éxito se deben manipular ciertas tecnologías como metodologías a usar, lenguajes de programación, sistemas gestores de bases de datos y patrones arquitectónicos. Se presenta el análisis realizado durante la investigación sobre la propuesta tecnológica realizada por el área temática donde se desarrolla el software.

### **1.4 Aplicación web.**

Una **aplicación web** es un conjunto de páginas web estáticas y dinámicas. La primera es aquella que no cambia cuando un usuario la solicita porque el servidor web envía la página al navegador web solicitante sin modificarla. Por el contrario, el servidor modifica las páginas web dinámicas antes de enviarlas al navegador solicitante. La naturaleza cambiante de este tipo de página es la que le da el nombre de dinámica.

Los usuarios la utilizan accediendo a un servidor web a través de Internet o de una Intranet, contiene información específica de un tema en particular y es almacenado en algún sistema de cómputo. Las aplicaciones web permiten facilitar la comunicación entre gobiernos, instituciones educativas, empresas, asociaciones y personas, con el propósito de establecer una relación más estrecha entre ellos.

### **1.5 Servidor web.**

Los **servidores web** son máquinas que actúan como "almacenes" de información. Básicamente, un servidor web sirve como contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que interactúan el uno con el otro mediante una conexión. (5)

Los servidores se conectan a la red mediante una interfaz que puede ser una red verdadera o mediante conexión vía línea telefónica o digital.

Existen muchas clases de servidores, el servidor Apache es el complemento perfecto para las páginas dinámicas desarrolladas con PHP y MySQL. Comparte con estos muchas de sus características, como son su popularidad, sencillez de manejo y versatilidad. Es un software (libre) servidor HTTP de código abierto para plataformas Unix, Windows, Macintosh y otras.

### **1.6 Navegador web.**

Un **navegador web o browser** es una aplicación software que permite a los usuarios acceder a diferentes tipos de recursos en Internet. Debe permitir recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML (HyperText Markup Language) desde servidores web de todo el mundo a través de Internet. En sentido general, los navegadores aportan las ventajas de ser un estándar para la comunidad tecnológica que desea publicar documentos. Agrupan una serie de tareas complejas en un solo paquete.

Algunos extienden funcionalidades hacia otras áreas como lectura de correos electrónicos y apertura de diversos documentos en distintos formatos. Entre los más usados actualmente se encuentran Mozilla Firefox e Internet Explorer.

### **1.7 Sistemas gestores de bases de datos.**

Un **Sistema Gestor o Manejador de Bases de Datos (SGBD)** es un conjunto de programas que permite a los usuarios crear y mantener una Base de Datos (BD), por lo tanto, el SGBD es un software que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. (6)

Puede ser de propósito general o específico. Posibilita la independencia de los datos y los programas de aplicación, la integridad y la facilidad de manipulación de la información. Existen varios gestores de base de datos, entre ellos: Oracle, SQLServer, PostgreSQL y MySql.

Para la creación del software se utilizará MySQL que es, sin duda, la base de datos más popular y utilizada al desarrollar páginas web dinámicas y sitios de comercio electrónico. Se suele trabajar en combinación con PHP, y comparte con este algunas de las características que lo convierten en una elección segura, entre ellas están: popularidad, rapidez, versatilidad, sencillez de manejo, rendimiento, bajo costo, facilidad de uso y portabilidad.

### **1.8 Metodologías de software.**

La rama de la **metodología**, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software, centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega.

Las metodologías de software ocupan un lugar importante para producir software de calidad en cualquier contexto de desarrollo, pues determinan la planeación y seguimiento del proyecto que se desarrolla.

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica lo que hay que obtener a lo largo del desarrollo del proyecto pero no la forma de hacerlo. La metodología muestra como se obtienen los distintos productos parciales y finales.

A través de la historia se han desarrollado varios modelos de proceso de software cada uno con sus ventajas, desventajas y utilidad en algunos tipos de proyectos y problemas. Existen dos tipos de metodologías software:

Están las **metodologías ágiles** que son estrategias de desarrollo de software que promueven prácticas que son adaptativas en vez de predictivas, centradas en la gente o en los equipos, iterativas, orientadas hacia prestaciones y hacia la entrega, de comunicación intensiva, y que requieren que el negocio se involucre en forma directa. (7)

Los grupos de trabajo están diseñados con una mínima cantidad de personas, trabajando todos en el mismo sitio, generan pocos artefactos, poseen pocos roles y no hacen énfasis en la arquitectura del software. Se centran en aquel software con plazos reducidos y requisitos volátiles.

Existen numerosas metodologías ágiles entre las que se encuentran: Extreme Programming (XP), Dynamic System Development Method (DSDM), Adaptive Software Development (ASD), Scrum y Feature Driven Development (FDD).

### **eXtreme Programing (XP).**

La **Programación Extrema** es una de las metodologías de desarrollo de software más utilizada, se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Consiste en una programación rápida en la cual los individuos e interacciones son más importantes que los procesos y herramientas.

Al mismo tiempo es más primordial un software que funcione que una documentación exhaustiva y la respuesta ante el cambio y la colaboración con el cliente tienen mayor prioridad que el seguimiento de un plan y la negociación de contratos respectivamente. En la actualidad se utiliza para proyectos de corto plazo y pequeño equipo.

#### *XP tiene como características fundamentales:*

- Desarrollo iterativo e incremental, empieza en pequeño y añade funcionalidad.
- Pruebas unitarias, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas. Nadie programa en solitario, las tareas de desarrollo se llevan a cabo en dúo en un mismo ordenador. Se supone que la calidad del código escrito de esta manera es mayor (pues este es revisado y discutido mientras se escribe) lo que es más importante que la posible pérdida de productividad inmediata.
- Refactorización del código. Mantener el código en buena forma o sea, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento y sin la introducción de fallos.
- Frecuente interacción del equipo de programación con el cliente o usuario, un representante del cliente trabaja junto al equipo de desarrollo. El cliente o el usuario se convierten en miembro del equipo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Obliga a realizar entregas frecuentes.
- Diseño simple. El principio es utilizar el diseño más sencillo que consiga que todo funcione.
- Propiedad del código compartida, en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores sean detectados.

### **Dynamic Systems Development Method (DSDM).**

Define el marco para desarrollar un proceso de producción de software.

### Sus principales características son: (8)

- Es un proceso iterativo e incremental.
- El equipo de desarrollo y el usuario trabajan juntos.
- Propone cinco fases: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.

Para alcanzar estas metas promueve el uso del desarrollo rápido de aplicaciones. El foco radica en la frecuente entrega de productos, los cambios durante el desarrollo son reversibles y la línea de base de los requerimientos es de alto nivel. Esto permite que los requerimientos de detalles se cambien según se necesite y que los esenciales se capten tempranamente.

La prueba está integrada a través de todo el ciclo de vida y es esencial una estrategia colaborativa y cooperativa entre todos los participantes. Las responsabilidades son compartidas y la colaboración entre usuario y desarrolladores no debe tener fisuras.

DSDM reconoce que los proyectos son limitados por el tiempo y los recursos, y los planes acorde a las necesidades de la empresa.

### **Adaptive Software Development (ASD).**

Es el modelo de implementación de patrones ágiles para desarrollo de software que da forma a las fases básicas de la gestión ágil. Se adapta al cambio en lugar de luchar contra él. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios.

Se basa en la adaptación continua a circunstancias cambiantes. En esta no hay un ciclo de planificación-diseño-construcción del software, sino un ciclo especular-colaborar-aprender. (9) La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo. No proporciona el tipo de prácticas detalladas como lo hace XP.

ASD posee un trabajo orientado y guiado por la misión del proyecto, basado en la funcionalidad con un desarrollo iterativo y acotado temporalmente. Es guiado por los riesgos y tolerante al cambio.

### **Scrum.**

Es un proceso liviano que sirve para administrar y controlar el desarrollo del software. El desarrollo se realiza en forma iterativa e incremental. Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. (10)

Su ciclo de vida se compone de cuatro fases:

- Pre-Juego: Planeamiento.
- Pre-Juego: Montaje (Staging).
- Juego o Desarrollo.
- Pos-Juego: Liberación.

Scrum es la agrupación de los miembros del equipo. De esta manera el equipo trata de recorrer la distancia hacia la meta como una unidad. Es usado para trabajos complejos en los cuales es imposible predecir todo lo que ocurrirá.

Ofrece un framework y un conjunto de prácticas que mantienen todo visible lo que permite saber que está pasando y hacer ajustes en el momento para mantener al proyecto avanzando hacia las metas fijadas.

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Controla de forma empírica y adaptable la evolución del proyecto con las siguientes prácticas de la gestión ágil: revisión de las iteraciones, desarrollo incremental, desarrollo evolutivo, auto-organización y colaboración entre todos según su conocimiento y no según su rol.

La calidad en este caso, se logra y se mantiene de forma continua, pues se involucra al cliente durante todo el tiempo que tarda el desarrollo, permitiéndole hacer aportes que enriquezcan y generen nuevas funcionalidades y/o características al producto que se está desarrollando.

### **Feature Driven Development (FDD).**

Define un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente puede ver y monitorizar. Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. No requiere un modelo específico de proceso y se complementa con otras metodologías.

Está pensado para proyectos con tiempo de desarrollo relativamente cortos. Propone tener etapas de cierre cada dos semanas, lo cual implica que los desarrolladores tendrán nuevas actividades que

realizar en dicho período de tiempo. Esto hace que la motivación del equipo se mantenga durante todo el proyecto debido a que se ven los resultados periódicamente.

Con FDD, se obtienen porcentajes reales del proceso, lo cual ayuda a demostrar al cliente donde se encuentra el proyecto. Los clientes aprueban esta idea ya que pueden tener planes con entregas y resultados frecuentes, o sea, pretende dejar satisfechos a los desarrolladores, gerentes y clientes sin afectar el proyecto.

Enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso. La parte iterativa soporta desarrollo ágil con rápidas adaptaciones a cambios en requerimientos y necesidades del negocio.

Cada fase del proceso tiene un criterio de entrada, tareas, pruebas y un criterio de salida. FDD consta de cinco procesos secuenciales durante los cuales se diseña y construye el sistema:

- Desarrollar un modelo global.
- Construir la lista de funcionalidades.
- Plan de versiones en base a funcionalidades a implementar.
- Diseñar e implementar las funcionalidades.

El empleo de esta metodología se considera adecuado para proyectos grandes y de misión crítica.

Existen las **metodologías tradicionales**, basadas en normas provenientes de estándares, seguidos por el entorno de desarrollo, con cierta resistencia a los cambios. El proceso es mucho más controlado, con numerosas políticas o normas, existe un contrato prefijado y el cliente interactúa con el equipo de desarrollo mediante reuniones, los grupos de trabajos son grandes y posiblemente distribuidos. Generan gran cantidad de artefactos y roles. (11) Entre estas están: Microsoft Solution Framework (MSF) y Rational Unified Process (RUP), siendo esta última una de las más utilizadas.

### **Microsoft Solution Framework (MSF).**

Se caracteriza por ser flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. Se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. (12)

Posee las siguientes características:

- Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas e incluso proyectos que requieren hasta 50 personas.

- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología agnóstica: puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto. Según el equipo de trabajo que maneja, no parece ser tan rígido y sus roles a desempeñar tampoco lo son, ya que pueden ser escalados dependiendo de la magnitud del proyecto.

Aplica mucho e incentiva al trabajo en equipo y a la colaboración. Es útil para proyectos de pequeña y gran escala. Crea una disciplina de análisis de riesgos que ayuda y evoluciona con el proyecto. Cuenta con plantillas que ayudan para el proceso de documentación. Por ser un modelo prescriptivo, solicita demasiada documentación en sus fases. El análisis de riesgos es necesario, pero si se hace muy exhaustivo puede demorar o hasta frenar el avance del proyecto.

### **Rational Unified Process (RUP) con notación UML.**

Es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología más utilizada para el análisis, diseño y documentación de sistemas Orientados a Objetos (OO). Esta actúa como un modelo que puede adaptarse a cualquier tipo de proyecto y empresas (grandes o pequeñas). RUP es un marco de desarrollo de software que se caracteriza por:

- *Ser iterativo e incremental*: divide el proceso de desarrollo en ciclos e iteraciones, teniendo un producto final al terminar cada uno de ellos.
- *Está dirigido por casos de uso*: un caso de uso describe un fragmento de las funcionalidades del sistema que proporciona al usuario un resultado importante. Los casos de uso guían el diseño, construcción y prueba del sistema, por lo que guían el proceso de desarrollo.
- *Está centrado en la arquitectura*: le permite a los desarrolladores una mayor visibilidad del sistema, pues la arquitectura es una vista del diseño completo del software con las características más importantes resaltadas.

RUP es un proceso para el desarrollo de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Divide en cuatro fases el desarrollo del software: Inicio, Elaboración, Construcción y Transición y posee nueve flujos de trabajo, seis básicos y tres de apoyo. Mantiene al equipo enfocado en producir incrementalmente software operativo a tiempo, con las características y con la calidad requerida.

Permite la producción de software de calidad dentro de plazos y presupuestos predecibles. Ayuda a planificar, diseñar, ejecutar y evaluar pruebas que verifiquen las cualidades del software. Igualmente indica como controlar, rastrear y monitorear los cambios dentro del proceso iterativo de desarrollo.

Esta metodología utiliza como lenguaje de modelado **UML** que es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software OO. Hay que tener en cuenta un aspecto importante: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado.

Toda metodología ágil conlleva a no realizar procesos que son requeridos en las metodologías tradicionales, lo cual implica que empresas grandes; las cuales necesitan llevar un mayor control de procesos y personas, tener tareas asignadas a un estado o proceso específico, y en las cuales dicho incremento de procesos no afectan en gran medida al costo final del producto; elijan una metodología tradicional que resulta más rentable por el gran volumen de personal, de productos, y de costos que se manejan y para los cuales se tendrá un mayor control.

Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo y tipo de sistema). Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto.

Una vez realizado un análisis detallado se determinó que se empleará una de las metodologías tradicionales más usadas, en este caso RUP. Está implantada para adaptarse a cualquier tipo de proyecto, organización y proceso que esté implícito dentro de esta, permitiendo su modelación.

Posibilita que en cada ciclo o iteración se obtenga un producto final y una serie de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades, quedando bien documentado el software de forma perdurable y extensible.

### **1.9 Arquitecturas de software y patrones arquitectónicos.**

Una **Arquitectura de Software** es la representación de más alto nivel de la estructura de un sistema o aplicación que describe las partes que lo integran, las interacciones entre ellas, los patrones que supervisan su composición y las restricciones a la hora de aplicarlos.

Tiene la responsabilidad de:

- Definir los módulos principales.
- Definir las responsabilidades que tendrá cada uno de los módulos.
- Definir la interacción que existirá entre los mismos.

Es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán y los principios que orientan su diseño y evolución. Su objetivo principal es aportar elementos que ayuden a la toma de decisiones y al mismo tiempo proporcionar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participan.

Se seleccionó la arquitectura por parte del grupo de arquitectura MINSAP-MIC para los software en desarrollo que se integrarán al SISalud. A continuación se mencionan las que se utilizarán para el desarrollo de la aplicación.

### **Cliente-Servidor.**

El concepto de cliente-servidor proporciona una forma eficiente de utilizar todos los recursos de máquina de tal forma que la seguridad y fiabilidad que proporcionan los entornos se traspasan a la red de área local.

En esta, un programa, o sea, el cliente informático, produce una demanda de información a cualquier otro programa que la proporciona, conocido como servidor, este último responde a la demanda del cliente que la produjo.

Normalmente el servidor es una máquina bastante potente que actúa como depósito de datos y funciona como un SGBD. Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red.

Cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla según le convenga. Los distintos servidores pueden intercambiar información dentro de esta arquitectura. Los usuarios deben tener un permiso determinado para acceder o no a los datos, para lo cual se crean diferentes roles de acceso.

La interfaz de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él y de la ubicación de la información. Los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Mejora el rendimiento de la red pues elimina la necesidad de mover grandes bloques de información hacia los ordenadores personales.

Puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos que pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias. En esta arquitectura son más importantes las ventajas de tipo organizativo debido a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La aplicación que se diseña será centralizada y montada en un servidor de aplicaciones, en este caso los servidores de INFOMED. Brindará servicios a numerosos clientes que estarán distribuidos a lo largo del país en las diferentes unidades de vigilancia y lucha antivectorial.

### **En tres capas.**

La arquitectura en capas es la generalización de la arquitectura Cliente-Servidor, donde la carga se divide en varias partes con un reparto claro de funciones. Una capa de presentación, otra parte para las reglas del negocio, denominada capa de negocio, y la capa de datos para el almacenamiento de los mismos, pudiéndose crear nuevas capas intermedias de ser necesario.

#### La aplicación se divide en las siguientes tres capas:

1. La capa de presentación o interfaz de usuario. En este caso, está formada por los formularios y los controles que se encuentran en los formularios. Es la capa con la que interactúa el usuario.
2. La capa de negocio. Está formada por las entidades del negocio, que representan objetos que van a ser manejados o consumidos por toda la aplicación.
3. La capa de acceso a datos. Contiene clases que interactúan con la base de datos, estas clases altamente especializadas permiten, utilizando los procedimientos almacenados generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio.

Esta arquitectura posibilita la centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo; la no replicación de lógica de negocio en los clientes, lo que permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.

Está soportada sobre un nivel de abstracción creciente, lo cual permite a los desarrolladores la fragmentación de un problema complejo en una secuencia de pasos incrementados. Proporciona una amplia reutilización, pues los datos abstractos pueden ser utilizados por diferentes implementaciones o versiones de una misma capa, en la medida que soporten las mismas interfaces, permitiendo flexibilidad y facilidad a la hora de realizar posibles modificaciones.

Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC. El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular.

La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a las restantes.

### **Basada en Componentes.**

Su objetivo es construir aplicaciones complejas mediante ensamblado de módulos (componentes) que han sido previamente diseñados por otras personas a fin de ser reusados en múltiples aplicaciones. Su premisa es que los componentes cumplan con alta cohesión y bajo acoplamiento.

En esta la interfaz constituye el elemento básico de interoperabilidad. Cada componente debe describir de forma completa las interfaces que ofrece, así como las interfaces que requiere para su operación y correcto funcionamiento, con independencia de los mecanismos internos que utilice para soportar la funcionalidad de la interfaz.

### Los beneficios más importantes de esta arquitectura son los siguientes:

- *Reusabilidad de servicios:* Reducción considerable de tiempos y costos de desarrollo al utilizar servicios disponibles desarrollados con anterioridad, para resolver problemáticas comunes a otras aplicaciones. Aumentando por esta razón la robustez del nuevo sistema al utilizarse software que está probado.
- *Interoperabilidad de aplicaciones:* Disminución de la complejidad en el proceso de integración, pues se interactúa con elementos que se abstraen de la tecnología y ubicación de los servicios.

Se debe tener en cuenta que este tipo de arquitectura permite la integración de múltiples sistemas, lo que implica una dependencia en la información gestionada por cada uno de ellos. Los sistemas integrados deben convertir al software en una solución cualitativamente superior pero no debe ser una relación de dependencia absoluta. Se debe lograr que exista entre las aplicaciones integradas un débil acoplamiento de tal forma que de producirse alguna modificación tenga la mínima repercusión en los sistemas dependientes. (13)

Se emplea esta arquitectura porque el sistema estará integrado con componentes del SISalud de los que consumirán los siguientes servicios:

- Sistema de Autenticación, Autorización y Auditoría. (SAAA).
- Registro de Unidades de Salud (RUS).
- Registro de Ubicación (RU).
- Registro de Localidades (RL).

### **Modelo-Vista-Controlador (MVC).**

La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño ofuscado que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando se necesite cambiar la interfaz, se tiene que modificar trabajosamente los componentes de negocio, lo que conlleva a un mayor trabajo y riesgo de error.

Este patrón se centra en la separación de los datos o modelo, y la vista, mientras que el controlador es el encargado de relacionarlos, o sea, su principal característica es aislar la vista del modelo con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Se divide de la siguiente forma: (14)

- *Modelo:* Administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado, usualmente formulados desde la vista, respondiendo a instrucciones de cambio para modificar el estado de los datos, habitualmente desde el controlador. Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos. Define las reglas de negocio (la funcionalidad del sistema).
- *Vista:* Se encarga de recibir datos del controlador y mostrarlos al usuario. Pueden existir múltiples vistas, donde cada una tiene asociado un componente controlador. Presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. El controlador elige la vista a usar, y hace disponible los datos que necesita.
- *Controlador:* Recibe las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del mouse y pulsaciones de teclas. Los eventos son traducidos a solicitudes de servicio para la vista.

En una aplicación web el controlador recibe la petición del usuario, interactúa con el modelo para procesar los datos y hace disponible esos datos a la vista. Los controladores son la única parte del MVC que deben ser definidos.

Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado, la principal ventaja de esta separación reside en la facilidad para realizar cambios en la aplicación puesto que cuando se realiza un cambio de bases de datos, programación o interfaz de usuario solo se toca uno de los componentes, o sea, se puede modificar uno de los componentes sin conocer como funcionan los restantes.

Es importante porque simplifica la comprensión y la organización de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores y ayuda a identificar que puede reutilizarse.

### **1.10 Herramientas Case.**

CASE es una sigla, que corresponde a las iniciales de: Computer Aided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computación. (15)

Las Herramientas Case representan una forma que permite modelar los Procesos de Negocios de las empresas y desarrollar los sistemas de información gerenciales. Proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información cuando los métodos y sistemas no satisfacen las necesidades de la organización.

Estas no son más que la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación.

El uso de las herramientas CASE puede mejorar la productividad en el desarrollo de una aplicación de bases de datos, por productividad se entiende tanto la eficiencia en tiempo como en dinero, de desarrollar la aplicación. Facilitan la mejora de la calidad y la normatividad.

### **Rational Rose.**

Es una herramienta para el modelado visual de sistemas software mediante UML, permite especificar, analizar y diseñar el sistema antes de codificarlo. Propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando un modelo completo que representa el dominio del problema y el sistema de software.

Utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Admite varias personas trabajando a la vez en el proceso iterativo

controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y que tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. Es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades.

Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

### **Visual Paradigm.**

Es una herramienta UML fácil de usar que soporta ingeniería inversa, generación de código, importación desde Rational Rose, exportación/importación XML, generación de informes y edición de figuras. (16)

Es muy profesional y soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y de menor costo. Proporciona a los desarrolladores una plataforma que les permita diseñar un producto con calidad de una forma rápida. Facilita la interoperabilidad con otras herramientas.

Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal. Genera código y realiza ingeniería inversa para varios lenguajes de programación entre ellos: Java, C++ y PHP. Se integra con el Visio para importar imágenes del mismo para realizar los diagramas de despliegue. Genera documentación para el proyecto en HTML, Microsoft Word y PDF.

Permite visualizar y diseñar los elementos del software, brinda facilidades para el diseño de los diagramas necesarios y su documentación. Es más fácil de utilizar que el Rational Rose desde el punto de vista informático.

Se determinó utilizar la herramienta Visual Paradigm pues es muy favorable para el diseño de aplicaciones multiplataformas. Ofrece la posibilidad de generar código en varios lenguajes, entre ellos PHP que es el lenguaje de programación utilizado para la implementación del sistema y fue la propuesta realizada por el grupo de arquitectura de la facultad 7, donde se desarrolla la aplicación.

### 1.11 Lenguajes de programación web.

Para el desarrollo de aplicaciones web existen muchos lenguajes. Se dividen en *lenguajes del lado del servidor*, son aquellos reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él; y *lenguajes del lado del cliente*, son los que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

Dentro de los lenguajes del lado de servidor más utilizados para el desarrollo de aplicaciones web se tienen: Personal Home Page (PHP), Active Server Pages (ASP) y Java Server Pages (JSP).

**PHP** es un lenguaje de propósito general, interpretado y ejecutado en el servidor para la creación de contenido dinámico para sitios web y está desarrollado en política de código abierto. Su código se incrusta dentro del HTML e interactúa con el mismo, lo que permite diseñar la página web en un editor común de HTML.

Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como: UNIX, Linux y Windows, y puede interactuar con los servidores web más populares como Apache. Permite la conexión a diferentes tipos de servidores de bases de datos tales como: MySQL, Postgres, Oracle, Microsoft SQL Server y SQLite; lo cual permite la creación de aplicaciones web muy robustas. Es seguro, con gran librería de funciones y mucha documentación.

Permite hasta cierto punto las técnicas de Programación Orientada a Objetos. Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, pero sin embargo para que sus páginas funcionen, el servidor donde están alojadas debe soportar PHP. Se utilizará como framework el Code Igniter 1.6.1.

Entre los lenguajes del lado del cliente se utilizará **JavaScript** este es un lenguaje interpretado y ejecutado por el cliente, sin embargo, posee una característica que lo hace especialmente idóneo, pues son los navegadores que se utilizan para viajar por ella los que interpretan (y por tanto ejecutan) los programas escritos en JavaScript. De esta forma, se puede enviar documentos a través de la web que llevan incorporados el código fuente de programas, convirtiéndose así en documentos dinámicos, y dejando de ser simples fuentes de información estáticas.

JavaScript no es un lenguaje OO propiamente dicho porque no dispone de herencia, es más bien un lenguaje basado en prototipos, pues las nuevas clases se generan clonando las clases bases (prototipos) y extendiendo su funcionalidad.

### **Conclusiones.**

Se ha explorado con el fin de encontrar los sistemas existentes semejantes o vinculados al campo de acción tanto a nivel internacional como nacional, encontrando solo algunos semejantes en Cuba. Se han mencionado sus deficiencias y se pudo concluir que la posibilidad de usarlos en las unidades de vigilancia y lucha antivectorial del país es muy escasa.

Se han expuesto las tendencias y tecnologías actuales, justificando las que se han propuesto en el desarrollo de la aplicación para dar una solución informática a la situación existente. Evitando realizar una aplicación de gran costo, se determinó utilizar las siguientes herramientas y tecnologías:

- Gestor de base datos: MySQL 5.0.
- Lenguaje de programación del lado del servidor: PHP 5.1.4.
- Lenguaje de programación del lado del cliente: JavaScript.
- Metodología de Desarrollo de Software: RUP con notación UML.
- Herramienta CASE: Visual Paradigm.
- Servidor web: Apache 2.0.
- Navegador: Internet Explorer o FireFox.

### **CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.**

En el capítulo actual se profundiza en la fase de inicio que tiene como prioridad definir la visión, los objetivos y el alcance del proyecto. Se describe el flujo actual de los procesos en las unidades de vigilancia y lucha antivectorial realizando un análisis de los mismos. Se definen las funcionalidades a automatizar realizando la propuesta del sistema, se genera el modelo de negocio y se explica la comunicación con sistemas externos. Se obtiene el modelo de casos de uso del sistema con sus respectivos diagramas.

#### **2.1 Objetivos de la organización.**

La emergencia y reemergencia de las enfermedades transmitidas por el mosquito *Aedes Aegypti* constituyen una de las consecuencias a las que se enfrenta la población actual, como resultado entre otros factores, de las inadecuadas condiciones socio-ambientales en que viven muchas comunidades del planeta.

El dengue actualmente amenaza a dos quintas partes de la población mundial en más de cien países, a pesar de la investigación intensiva y las acciones desarrolladas en los últimos años, continúa siendo la primera causa de mortalidad por virus en el mundo.

El programa *Control Sanitario Internacional* dirigido por el Viceministerio de Higiene y Epidemiología está dividido en los subprogramas: Salud Ambiental, Higiene y Epidemiología y Vectores. En este último se desarrolla el programa de Vigilancia y Lucha Antivectorial que está enfocado a eludir la introducción, propagación y transmisión de enfermedades provocadas por vectores, centrado inicialmente en el control de focos de mosquitos *Aedes Aegypti*. Para lo cual se han trazado los siguientes objetivos:

- Estructurar y desarrollar los programas y campañas de vigilancia y lucha antivectorial.
- Normar los métodos y técnicas de lucha y vigilancia con relación a vectores.
- Asesorar y controlar la aplicación de los programas, evaluando su cumplimiento, así como el de los métodos y técnicas de lucha antivectorial normados en todos los niveles del sistema de salud y otras entidades u organismos.

Las acciones del Programa de Erradicación son múltiples y todas generan informaciones importantes, dentro de ellas se pueden mencionar de forma simplificada las siguientes:

- Las inspecciones de las viviendas, centros de trabajos o terrenos baldíos por los Operarios A.
- La búsqueda de muestras de posibles focos.

- El tratamiento de los depósitos en los que se encuentran muestras.
- Las supervisiones que realizan los Jefes de Brigada.
- Las acciones de radiobatidas donde se realiza verificación, tratamiento focal y adulticida.
- Las revisiones de larvitrapas instaladas en las áreas de salud.
- La inspección a centros priorizados y zonas de riesgo.

Estas funciones son realizadas a nivel de área de salud a partir de las cuales se generan las estadísticas necesarias.

### **2.2 Flujo actual de los procesos.**

En el programa de vigilancia y lucha antivectorial para el control de focos del mosquito *Aedes Aegypti* se realizan una serie de procesos que transitan por los diferentes niveles desde el área de salud hasta el Viceministerio de Higiene y Epidemiología, los que se mencionan a continuación:

#### Inspecciones a locales

Al comenzar el día el *Controlador* del área de salud le indica a cada uno de los *Operarios A* el trabajo que debe realizar, estos visitan los locales (centros de trabajo, viviendas particulares o centros especializados) que están en su plan de trabajo.

Al llegar un operario a un local determinado registra en una planilla los datos particulares del mismo y procede a inspeccionarlo. En estas visitas se examinan los lugares y depósitos posiblemente vulnerables a presentar focos de mosquitos. Se le realiza un tratamiento a los depósitos que puede ser: destrucción, flameo o abatización.

Una vez culminada la jornada laboral cada operario le entrega la información recopilada a su Jefe de Brigada.

#### Recogida y análisis de muestras

Si en alguno de los depósitos examinados se encuentra un posible foco de mosquito, el Operario A recolecta una muestra y la entrega al Jefe de Brigada. Este envía las muestras al laboratorio para su análisis. En caso de resultar positiva, el Laboratorista informa el resultado al Responsable del Área de Salud (AS) donde fue recogida la muestra.

#### Recopilar información por niveles

Además del resultado de las muestras enviadas del laboratorio, el Responsable del AS recolecta toda la información de las brigadas, hace un resumen y lo entrega a la UMVLA. En esta unidad se efectúa un sumario de las áreas de salud asociadas al municipio y se transmite a la UPVLA y esta a su vez a la Unidad Nacional de Vigilancia y Lucha Antivectorial (UNVLA). En el caso de estar en la frontera (puertos o aeropuertos) se entrega directamente a la UPVLA.

Al concluir todos los procesos el Viceministerio de Higiene y Epidemiología obtiene reportes estadísticos generales a nivel nacional, provincial, municipal y por áreas de salud, con el objetivo de controlar todos los locales inspeccionados, incluyendo los positivos, el índice de infestación y las manzanas positivas reiterativas, posibilitándose así que se tomen las medidas pertinentes en las zonas más críticas.

Los jefes de brigada recolectan los modelos y los transmiten a través de entrega personal, por teléfono, correo electrónico u otros medios al Responsable del AS. Lo que trae consigo que no se gestione toda la información de forma rápida y eficiente pues existe la posibilidad de ocurrir pérdida de la misma en su trayectoria y en ocasiones no se cuentan con datos reales y precisos pues pueden ser cambiados accidentalmente durante su traslado.

Provoca, al ser lenta la gestión, que no se proporcione la ubicación rápida y exacta de los focos, pues el Responsable del AS se puede retrasar en su trabajo. Esta información lejos de ser inmediata es tardía.

### **2.3 Objeto de automatización.**

Los procesos descritos en la sección anterior implican la elaboración de una serie de informes que forman parte del trabajo en la campaña de Lucha Antivectorial. Los documentos que se explican en esta sección son los seleccionados para automatizarse, debido a la importancia que representan en la agilización y mejoramiento del trabajo desempeñado en el ámbito de la campaña.

El documento *Informe de trabajo diario en la Lucha Antivectorial* se utiliza con el objetivo de recoger la información específica de los locales visitados por cada Operario A. Los que registran en las planillas la dirección exacta de cada inspección, sea vivienda, centro de trabajo o terreno baldío. Se recoge si la vivienda fue inspeccionada (VI), si estaba cerrada (VC), si fue recuperada (VR) en una posterior visita, o vivienda perdida (VP) en el caso de que no se haya examinado.

Se recoge el ciclo en el que se encuentra, que es el período de tiempo o la cantidad de días que se demora un operario en volver a inspeccionar el local. Los ciclos pueden ser semanales, mensuales o bimestrales.

Al mismo tiempo en cada local se recopilan los datos de los depósitos pudiendo ser: tanque, goma, lata, cascarón o cualquier recipiente que actúe como criadero. Se recoge el tipo de estos depósitos, que pueden ser naturales o artificiales y el tratamiento que se les realiza.

El documento *Focos Aedes Aegypti* se utiliza para recoger los datos principales y específicos de los locales positivos, como son: fecha, ciclo, municipio, dirección, manzana, larvas, pupas, especies asociadas y circunscripción. El mismo es confeccionado por cada uno de los responsables de las áreas de salud una vez que se les informe desde el laboratorio la existencia de un foco positivo en un lugar determinado.

El Responsable del AS recibe todos los informes de los Jefes de Brigada y confecciona la planilla *Resumen Diario* en la cual hace un resumen con los datos de todas las brigadas de su área de salud.

La hoja *Parte Focal* es una plantilla creada por el Responsable de la UMVLA que es confeccionada a partir de las plantillas de resumen que le entregan cada uno de los responsables de los departamentos de Vigilancia y Lucha Antivectorial en las áreas de salud. En esta se recoge el diario y el acumulado de las viviendas inspeccionadas, existentes, cerradas, recuperadas y perdidas. Igualmente se recogen los datos de los depósitos, o sea, si son inspeccionados, destruidos, tratados, o flameados. Se recoge la fecha y el ciclo en que se encuentran cada una de las áreas de salud.

Actualmente el cliente no posee ningún sistema automatizado que gestione la información recogida en las planillas explicadas, toda la información es procesada en papel, de forma manual, y en el mejor de los casos con herramientas del paquete de Office principalmente con Microsoft Excel y Word.

### **2.4 Propuesta del sistema.**

El sistema debe garantizar la gestión (inserción, actualización y eliminación) de la información que demandará la sostenibilidad del programa de vigilancia y lucha antivectorial la cual debe estar accesible y actualizada en todos los centros de salud médica para garantizar el mantenimiento de índices de infestación ínfimos, que impidan la propagación del dengue.

Como objetivo principal se tiene lograr un adecuado diseño de una aplicación web sencilla de manejar, fácil de comprender y con una interfaz amigable que inspire confianza y control en lo que se esté

haciendo. Al mismo tiempo que procese toda la información relacionada con el control de los focos de mosquitos que representan peligro para la sociedad cubana, o sea, Vectores posibilitará reforzar las acciones de vigilancia y control de estos transmisores de enfermedades.

Para la optimización de todos los procesos se definen las siguientes características:

- Permitir la autenticación de cada usuario en el sistema proporcionándole acceso.
- Definir los tipos de usuarios y los diferentes roles y privilegios que posee dentro de la aplicación.
- Mostrar a los usuarios las opciones a las que tiene acceso.
- Registrar, modificar y eliminar la información de locales inspeccionados por brigadas, locales positivos y larvitrapas.
- Visualizar manzanas positivas reiterativas y reportes estadísticos generales de positividad al vector en el área de salud, municipio, provincia y nación.
- Visualizar información de locales inspeccionados por brigadas, locales positivos y larvitrapas en las áreas de salud, municipios, provincias y nación.
- Visualizar índice de infestación de las áreas de salud (por manzanas).

Las soluciones existentes hasta el momento no dan respuesta a ninguna de las funcionalidades anteriormente expuestas. Se propone la realización de un nuevo sistema que cumpla con cada uno de los requisitos necesarios, o sea, el nuevo software permitirá viabilizar la gestión de la información relacionada con el control de focos de mosquitos *Aedes Aegypti* a nivel de área de salud, municipios, provincias y nación, contándose en todo momento con los datos más actualizados.

### **2.5 Modelo de Negocio.**

Al comenzar el desarrollo de un software según la metodología RUP el primer flujo de trabajo que se desarrolla es el Modelamiento del Negocio en el cual se analizan los procesos fundamentales que se realizan en la organización, se comprende la estructura, dinámica y problemas actuales de la misma y se identifican las mejoras potenciales. Se llega a un consenso entre los consumidores, usuarios finales y desarrolladores para definir los requerimientos del sistema.

Los problemas existentes en los procesos de negocio explicados anteriormente se solucionarán mediante la automatización de los mismos, dándole respuesta de la siguiente forma:

- Permitirá almacenar gran cantidad de información estadística y un acceso rápido a la misma.
- Disminuirá la introducción de errores humanos pues el sistema se encargará de realizar los cálculos pertinentes.

- Permitirá obtener información actualizada en el momento deseado, desde todas los centros de vigilancia y lucha antivectorial.
- Facilitará el trabajo de los especialistas que deben monitorear todos los datos relacionados con el control del mosquito Aedes Aegypti en Cuba.

### **2.5.1 Reglas del Negocio.**

Las reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. El proceso de especificación implica que hay que identificarlas dentro del negocio, evaluar si son relevantes dentro del campo de acción que se está modelando e implementarlas en la propuesta de solución. Son múltiples las clasificaciones que se dan a las reglas de negocio.

#### Vectores debe cumplir las siguientes reglas:

1. Los ciclos de trabajo de cada área de salud se definen a nivel de Viceministerio.
2. Puede existir un área de salud con más de un ciclo, en dependencia de la situación entomológica que presente.
3. Un área de salud nunca va a tener más de 3 ciclos.
4. Los ciclos de trabajo se realizan por semanas estadísticas.
5. Los ciclos siempre van a ser: de dos semanas (aquí pueden ser de dos o tres semanas, en dependencia de la cantidad de semanas estadísticas que traiga el mes), mensual o bimestral.
6. Los ciclos comienzan los lunes y terminan los sábados (alternos).
7. En un ciclo, el número máximo de visitas a un local realizadas por un operario, cuando estos locales se encuentran cerrados, es de 3 ó 4 veces.
8. Si un local es visitado esa cantidad de veces (definida con anterioridad) y siempre se encuentra cerrado se informa como perdido y se queda sin inspeccionar en ese ciclo.
9. El jefe de brigada envía las muestras al entomólogo, solo se registra en caso que sean positivas.
10. Una manzana puede ser visitada por más de un operario.
11. Una brigada puede visitar más de una manzana.

### **2.5.2 Actores del negocio.**

Un actor del negocio es cualquier individuo, grupo, entidad, organización o máquina que interactúa con el negocio. El actor siempre permanece fuera de la frontera del negocio que se está investigando.

<b>Actor</b>	<b>Descripción</b>
Controlador.	Es el encargado de asignar responsabilidades a los operarios al comenzar el día, iniciando así el proceso de inspección a locales.

**Tabla 2.1 Actores del negocio.**

### **2.5.3 Trabajadores del negocio.**

Un trabajador del negocio representa a personas o sistemas (software) dentro del negocio que son los que realizan las actividades que están comprendidas dentro de los procesos representados en un caso de uso. (17)

<b>Trabajador</b>	<b>Descripción</b>
Operario A.	Es el encargado de inspeccionar cualquier local en el programa de vigilancia y lucha antivectorial y de registrar la información de los vectores detectados (mosquitos).
Jefe de Brigada.	Su misión es recoger la información de todos sus operarios, hacer un resumen y entregarlo al Responsable del AS.
Responsable del AS.	Se encarga de recoger toda la información de las brigadas, realizar un resumen de su área de salud y transmitirlo al municipio.
Laboratorista.	Se encarga de analizar las muestras recogidas e informar las positivas.
Responsable UMVLA.	Recibe las estadísticas de sus áreas de salud, hace un resumen de la información del municipio y lo envía al Responsable de la UPVLA.
Responsable UPVLA.	Recibe las estadísticas de sus municipios, hace un resumen de la información de la provincia y lo envía al Responsable de la UNVLA.

Responsable UNVLA.	Recibe las estadísticas de las provincias y realiza un resumen general.
--------------------	---

Tabla 2.2 Trabajadores del negocio.

**2.5.4 Diagrama de Casos de uso del Negocio.**

Un caso de uso del negocio representa un proceso dentro del negocio que se estudia, por lo que se corresponde con una secuencia de acciones con un orden lógico y que producen un resultado observable para ciertos actores del negocio.



Figura 2.1 Diagrama de Casos de Uso del Negocio.

*Recopilar información de inspecciones a locales* detalla el proceso desarrollado por cada uno de los operarios en las inspecciones diarias.

*Recopilar información en los diferentes niveles* se refiere a los procesos de recopilación de los datos desde el AS hasta la UNVLA.

*Enviar muestras* consiste en el envío hacia el laboratorio de cada una de las muestras recopiladas por los Jefes de Brigadas para analizarlas y ver si son positivas o no.

**2.5.5 Descripción textual de los casos de usos del negocio (CUN).**

<b>Caso de Uso</b>	Recopilar información de inspecciones a locales.
<b>Actores</b>	Controlador.
<b>Resumen</b>	El CUN se inicia cuando el Controlador orienta a los Operarios A su trabajo diario, los que posteriormente comienzan a visitar cada uno de los centros que tiene en el plan de trabajo. El CUN termina cuando el operario termina de visitar todos los locales, entrega la información de los mismos al jefe de brigada y este hace un resumen de la información de sus operarios y la entrega al Responsable del Dpto. de VLA en el AS.

<b>Precondiciones</b>	---
<b>Referencias</b>	<ul style="list-style-type: none"> <li>• CUN Enviar muestra (extendido)</li> <li>• CUN Recopilar información en los diferentes niveles (AS, UMVLA, UPVLA, UNVLA) (incluido)</li> </ul>
<b>Prioridad</b>	Crítica.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. El caso de uso inicia cuando el Controlador da las orientaciones a los operarios al comenzar el día para realizar las inspecciones a los locales.	2. El Operario A visita el local, lo inspecciona y registra los datos del mismo.
	3. Si el Operario A encuentra muestras de mosquitos recolecta las mismas.
	4. El Operario A verifica si terminó la jornada laboral.
	5. Si terminó la jornada laboral entrega la información al Jefe de Brigada, de lo contrario, ir a 2.
	6. El Jefe de Brigada recibe información de sus Operarios A, hace un resumen de la información entregada por los mismos, la entrega al Responsable del AS y termina el caso de uso.
	7. Si el Jefe de Brigada recibió muestras, se realiza el CUN Enviar muestras y verifica si terminó de recopilar toda la información de sus operarios, de lo contrario ir a 6.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
	2.1. Si el local estaba cerrado se registra como local cerrado. Ir a 5.
<b>Poscondiciones</b>	Se hace un resumen de la brigada y se entrega al Responsable del AS.

**Tabla 2.3 Descripción textual CUN Recopilar información de inspecciones a locales.**

<b>Caso de Uso</b>	Recopilar información en los diferentes niveles (AS, UMVLA, UPVLA,
--------------------	--

	UNVLA).
<b>Actores</b>	---
<b>Resumen</b>	El CUN se inicia cuando el Responsable del AS recibe la información de las brigadas de su AS. La información es procesada y enviada a través de los diferentes niveles. El CUN termina cuando en la UNVLA se procesa la información de todo el país.
<b>Precondiciones</b>	Cada nivel debe contar con toda la información necesaria de los niveles precedentes para poder realizar los resúmenes.
<b>Referencias</b>	---
<b>Prioridad</b>	Crítica.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. El Jefe de Brigada entrega resumen al Responsable del AS.	2. El Responsable del AS recibe la información de los locales inspeccionados de las brigadas y de los locales positivos en caso de que se hayan detectado.
	3. Si terminó de recopilar toda la información de las brigadas, la procesa y hace un resumen de la misma, si no, ir a 2.
	4. El Responsable del AS entrega la información al Responsable de la UMLA.
	5. El Responsable de la UMLA recibe el resumen de sus AS.
	6. Si el Responsable de la UMLA terminó de recopilar toda la información de sus AS, la procesa, hace un resumen de la misma y la envía a la UPVLA, si no, ir a 5.
	7. El Responsable de la UPVLA recibe el resumen de sus AS.
	8. Si el Responsable de la UPVLA terminó de recopilar toda la información de sus AS, la procesa, hace un resumen de la misma y la envía a la UNVLA, si no, ir a 7.
	9. El Responsable de la UNVLA recibe el resumen de sus provincias.
	10. Si el Responsable de la UNVLA terminó de recopilar

	toda la información de sus provincias, la procesa y hace un resumen de todo el país, terminando así el CUN, si no, ir a 9.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
<b>Poscondiciones</b>	Se hace un resumen de la información de todo el país.

**Tabla 2.4 Descripción textual CUN Recopilar información en los diferentes niveles.**

<b>Caso de Uso</b>	Enviar muestras
<b>Actores</b>	---
<b>Resumen</b>	El CUN se inicia cuando el Jefe de Brigada entrega las muestras de su brigada al Laboratorista. El caso de uso termina cuando el Laboratorista analiza la muestra y entrega los resultados de dicho análisis.
<b>Precondiciones</b>	---
<b>Referencias</b>	---
<b>Prioridad</b>	Crítica.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. El Jefe de Brigada entrega las muestras al Laboratorista.	2. El Laboratorista las analiza.
	3. Si son positivas el Laboratorista lo informa al Responsable del AS, si no, termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
<b>Poscondiciones</b>	

**Tabla 2.5 Descripción textual CUN Enviar muestras.**

Para profundizar en el Modelo de Negocio ver [Anexo 1](#).

### **2.6 Especificación de los requisitos.**

El modelado del negocio brinda una visión general de los procesos que existen en una determinada organización, permitiendo comprender a que se dedica, así como establecer una comunicación entre el cliente y el equipo de desarrollo. Una vez en este punto, se comienza a definir lo que debe hacer el sistema, siendo necesaria la captura de los requerimientos.

El IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) define un requerimiento como una “condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo”. Además lo define como “condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.” (18)

Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Definen las funciones que el sistema será capaz de realizar y describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

#### **2.6.1 Requerimientos funcionales. (19)**

*Funcionalidades que el sistema debe permitir:*

R1.Gestionar información de locales inspeccionados por brigadas.

R1.1 Registrar información de locales inspeccionados por brigadas.

R1.2 Visualizar información de locales inspeccionados por brigadas.

R1.3 Modificar información de locales inspeccionados por brigadas.

R1.4 Eliminar información de locales inspeccionados por brigadas.

R2.Gestionar información de locales positivos.

R2.1 Registrar información de locales positivos.

R2.2 Visualizar información de locales positivos.

R2.3 Modificar información de locales positivos.

R2.4 Eliminar información de locales positivos.

R3.Gestionar información de larvitrapas.

R3.1 Registrar información de larvitrapas.

R3.2 Modificar información de larvitrapas.

R3.3 Visualizar información de larvitrapas.

R3.4 Eliminar información de larvitrapas.

R4. Visualizar Reporte estadístico general de positividad al vector.

R5. Visualizar manzanas positivas reiterativas.

R6. Visualizar reporte estadístico general por área de salud.

R7. Visualizar información de locales inspeccionados por área de salud.

R8. Visualizar información de locales positivos por áreas de salud.

R9. Visualizar índice de infestación de las áreas de salud (por manzanas).

R10. Visualizar reporte estadístico general por municipios.

R11. Visualizar información de locales inspeccionados por municipios.

R12. Visualizar información de locales positivos por municipios.

R13. Visualizar índice de infestación de áreas de salud en la provincia (por manzanas).

R14. Visualizar reporte estadístico general por provincias.

R15. Visualizar información de locales inspeccionados por provincias.

R16. Visualizar información de locales positivos por provincias.

R17. Visualizar índice de infestación de áreas de salud en la provincia.

R18. Gestionar Nomenclador (Brigada, Especie, Etapa Colecta, Nombre Depósito, Organismo, Rin, Tipo Ciclo, Tipo Colecta, Vector).

R18.1 Insertar información de Nomenclador.

R18.2 Modificar información de Nomenclador.

R18.3 Visualizar información de Nomenclador.

R18.4 Eliminar información de Nomenclador.

R19. Gestionar Ciclo.

R19.1 Insertar información de Ciclo.

R19.2 Visualizar información de Ciclo.

R19.3 Eliminar información de Ciclo.

R20. Gestionar Área de Salud-Ciclo.

R20.1 Insertar información de un ciclo en un área de salud.

R20.2 Modificar información de un ciclo en un área de salud.

R20.3 Visualizar información de un ciclo en un área de salud.

R20.4 Eliminar información de un ciclo en un área de salud.

R21. Gestionar manzanas.

R21.1 Insertar información de una manzana.

R21.2 Modificar información de una manzana.

R21.3 Visualizar información de una manzana.

R21.4 Eliminar información de una manzana.

Para conocer cuáles son los datos que se manejan en cada una de los requerimientos funcionales antes mencionados, se puede ver el documento *Especificación de Requisitos* en el Expediente del Proyecto.

### **2.7 Comunicación con otros sistemas.**

Para lograr un completo funcionamiento del sistema se necesitarán consumir varios servicios de otros software desarrollados con anterioridad en el SISalud. Los sistemas externos y los aspectos que se solicitarán de ellos son:

- El SAAA se utilizará con el objetivo de conocer el usuario que está autenticado en el sistema, el nivel al que pertenece, el de usuario es y a que módulos tiene acceso.
- El RUS para conocer cuál es el nombre de la Unidad de Salud (US) en la que se encuentra el usuario.
- El RU para listar todas las provincias, municipios y las localidades de cada municipio, para listar las manzanas por localidad, las calles por manzana y las calles por localidad.
- El RL se necesitará para listar los consejos populares por localidad y las circunscripciones por consejo popular.

### **2.8 Definición de los casos de uso del sistema.**

Los casos de uso se han convertido en la técnica más utilizada a nivel mundial para el levantamiento y la comunicación clara y eficiente de los requerimientos para el desarrollo de sistemas. (20)

Son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto un caso de uso consiste en una secuencia

de actividades que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Generalmente cada caso de uso debe comunicarse con al menos un actor.

Es de vital importancia realizar una buena selección de los casos de uso debido a que el proceso de desarrollo está guiado por ellos, lo que quiere decir que una serie de flujos de trabajo se inician a partir de los mismos.

Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios u otros sistemas. O lo que es igual, es un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

### **2.8.1 Definición de los actores del sistema.**

Un actor del sistema es una persona, grupo de persona o sistema que interactúa con la aplicación. Cada trabajador del negocio (inclusive si fuera un sistema existente) que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema.

<b>Actor</b>	<b>Descripción</b>
<b>Especialista AS</b>	Es el encargado de gestionar (Insertar, Modificar y Eliminar) la información de locales inspeccionados, locales positivos y larvitrapas del AS.
<b>Visualizador AS</b>	Es una generalización del rol de Especialista del AS para visualizar la información de los locales inspeccionados, locales positivos, larvitrapas, reportes estadísticos generales y manzanas positivas reiterativas. A este rol tendrán acceso los especialistas de las UMVLA.
<b>Especialista UMVLA</b>	Es el encargado de visualizar los reportes estadísticos generales por AS, índice de infestación de AS, locales inspeccionados y positivos por AS.

<b>Especialista UPVLA</b>	Se encarga de visualizar los reportes estadísticos generales por municipio, locales inspeccionados y positivos por municipio e índices de infestación por AS en la provincia.
<b>Especialista UNVLA</b>	Es el encargado de visualizar reportes estadísticos por provincia, locales inspeccionados y positivos por provincia e índice de infestación de la provincia.
<b>Administrador nacional</b>	Es el encargado de gestionar todas las configuraciones generales del sistema. Configurar los ciclos que tendrá cada AS, además de gestionar la información referente a todos los nomencladores.
<b>Administrador AS</b>	Es el encargado de configurar la aplicación a nivel de AS, como por ejemplo gestionar las brigadas de cada AS.

**Tabla 2.6 Actores del sistema.**

**2.8.2 Diagrama de casos de uso del sistema.**

El siguiente diagrama está formado por los casos de uso arquitectónicamente significativos, están representados solo a nivel de área de salud. El resto de los casos de uso, que son secundarios, son visualizaciones de esta información pero en los diferentes niveles del SNS (municipal, provincial y nacional), los cuales se representan en el diagrama de casos de usos del sistema representado en los anexos.

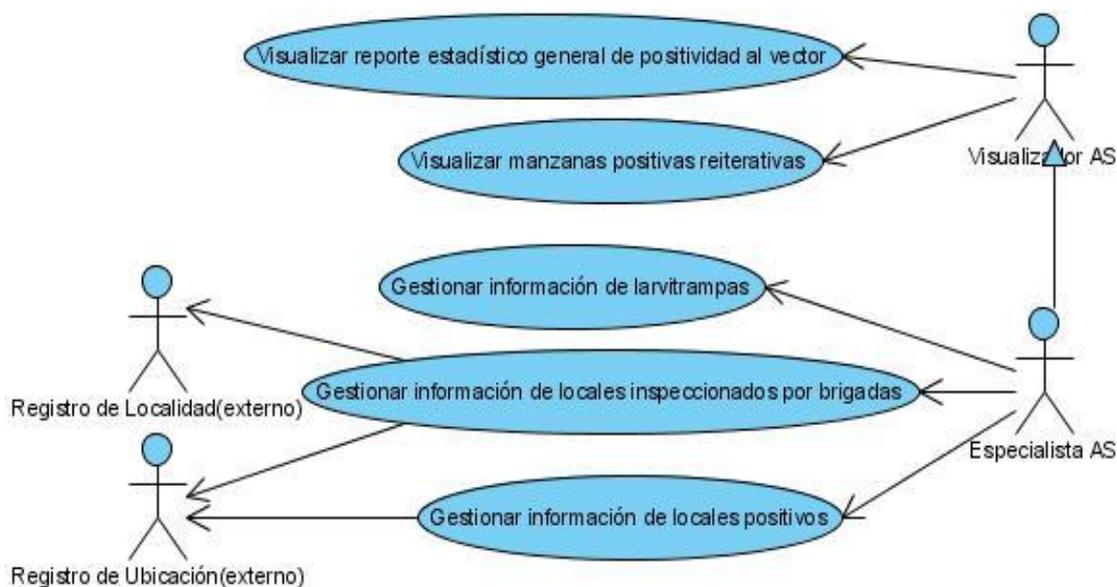


Figura 2.2 Diagrama de casos de uso del sistema (casos de usos arquitectónicamente significativos).

2.8.3 Descripción textual de los Casos de Usos del Sistema (CU).

<b>Caso de Uso</b>	Gestionar información de locales inspeccionados por brigadas.
<b>Actores</b>	Especialista AS.
<b>Resumen</b>	El CU se inicia cuando el actor Especialista AS accede a la opción de insertar, modificar, eliminar o visualizar los datos de los locales inspeccionados. El CU termina cuando este culmina cualquiera de esas acciones.
<b>Precondiciones</b>	El Especialista AS debe tener permiso para acceder a la aplicación.
<b>Referencias</b>	R1
<b>Prioridad</b>	Crítica.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Insertar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Especialista AS accede a la opción “Registrar Locales Inspeccionados”.	2- El Sistema muestra una pantalla para que el actor introduzca los datos.
3- El Especialista AS introduce los datos de los	4- El sistema valida los datos.

locales inspeccionados.		
		5- El sistema registra los datos y termina el CU.
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
		4.1- El sistema muestra un mensaje de error y termina el CU.
<b>Poscondiciones</b>	El sistema queda actualizado después de haber insertado los datos.	
<b>Flujo Normal de Eventos</b>		
<b>Sección "Modificar"</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1- Si el Especialista AS desea modificar algún dato selecciona la fila y pincha el botón "Modificar".		2- El Sistema muestra una nueva pantalla con los datos brindando la posibilidad de que el Especialista AS introduzca los nuevos datos.
3- El Especialista AS introduce los datos que desea modificar y pincha el botón "Aceptar"		4- El sistema valida los datos.
		5- El sistema modifica los datos y termina el CU.
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
		4.1- Si los datos están incorrectos el sistema lanza un mensaje de error y termina el CU.
<b>Poscondiciones</b>	El sistema queda actualizado después de haber modificado los datos.	
<b>Flujo Normal de Eventos</b>		
<b>Sección "Eliminar"</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1- El Especialista AS accede a la opción "Eliminar Locales Inspeccionados".		2- El Sistema muestra una pantalla para que el Especialista AS introduzca la fecha y la brigada a la cual desea eliminarle alguna inspección.

3- El Especialista AS introduce los datos.	4- El sistema busca la inspección con dichos criterios de búsqueda y la muestra al Especialista AS.
5- El Especialista AS pincha el botón “Eliminar”	6- El sistema debe mostrar un mensaje preguntando si realmente desea borrar la información.
	7- En caso positivo elimina la inspección y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	El sistema queda actualizado después de haber eliminado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Visualizar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Especialista AS accede a la opción “Mostrar Locales Inspeccionados”.	2- El Sistema muestra una pantalla para que el Especialista AS introduzca los datos de la inspección que desea ver.
3- El Especialista AS introduce los datos.	4- El sistema busca la inspección con dichos criterios de búsqueda y la muestra al Especialista AS, terminando así el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	

**Tabla 2.7 Descripción textual CUS Gestionar información de locales inspeccionados por brigadas.**

<b>Caso de Uso</b>	Gestionar información de locales positivos.
<b>Actores</b>	Especialista AS.
<b>Resumen</b>	El CU se inicia cuando el Especialista AS accede a la opción de insertar,

	modificar, eliminar o visualizar los datos de los locales positivos. El CU termina cuando este accede a otra opción.
<b>Precondiciones</b>	El Especialista AS ya debe tener permiso para acceder a la aplicación.
<b>Referencias</b>	R2
<b>Prioridad</b>	Critica.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Insertar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Especialista AS accede a la opción “Registrar Locales Positivos”.	2- El Sistema muestra una pantalla para que el Especialista AS introduzca los datos.
3- El Especialista AS introduce los datos de los locales positivos y pincha el botón “Registrar”.	4- El sistema valida los datos.
	5- El sistema registra los datos y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1- Si los datos están incorrectos el sistema lanza un mensaje de error y termina el caso de uso.
<b>Poscondiciones</b>	El sistema queda actualizado después de haber insertado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- Si el Especialista AS desea modificar algún dato selecciona la fila y pincha “Modificar Local Positivo”.	2- El Sistema muestra una nueva pantalla con los datos que fueron seleccionados anteriormente con posibilidad de que el Especialista AS modifique los datos.
3- El Especialista AS introduce los datos que desea modificar y pincha el botón “Aceptar”.	4- El sistema valida los datos.
	5- El sistema modifica los datos y

	termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1- Si los datos están incorrectos el sistema lanza un mensaje de error y termina el caso de uso.
<b>Poscondiciones</b>	El sistema queda actualizado después de haber modificado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Especialista AS selecciona la fila que desea eliminar y pincha “Eliminar”.	2- El sistema debe mostrar un mensaje preguntando si realmente desea borrar la información.
	3- En caso positivo elimina el local positivo y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	El sistema queda actualizado después de haber eliminado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Visualizar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Especialista AS accede a la opción “Mostrar Locales Positivos”.	2- El Sistema muestra una pantalla para que el Especialista AS introduzca los datos de los locales que desea ver.
3- El Especialista del AS introduce los datos.	4- El sistema busca los locales positivos con dichos criterios de búsqueda y los muestra al Especialista del AS, terminando así el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	

**Tabla 2.8 Descripción textual CUS Gestionar información de locales positivos.**

<b>Caso de Uso</b>	Gestionar información de larvitrapas.	
<b>Actores</b>	Especialista AS.	
<b>Resumen</b>	El CU se inicia cuando el Especialista AS accede a la opción de insertar, modificar, eliminar o visualizar los datos de las larvitrapas. El CU termina cuando este culmina cualquiera de estas acciones.	
<b>Precondiciones</b>	El Especialista AS debe tener permiso para acceder a estas funcionalidades.	
<b>Referencias</b>	R3	
<b>Prioridad</b>	Medio.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “Insertar”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El Especialista AS accede a la opción “Registrar Larvitrapas”.	2- El Sistema muestra una pantalla para que el actor introduzca los datos.	
3- El Especialista AS introduce los datos de la larvitrapa.	4- El sistema valida los datos.	
	5- El sistema registra los datos y termina el CU.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	4.1- El sistema muestra un mensaje de error y termina el CU.	
<b>Poscondiciones</b>	El sistema queda actualizado después de haber insertado los datos.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “Modificar”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- Si el Especialista AS desea modificar algún dato selecciona la fila y pincha el botón “Modificar”.	2- El Sistema muestra una nueva pantalla con los datos anteriores brindando la posibilidad de que el Especialista AS	

	introduzca los nuevos datos.
3- El Especialista AS introduce los datos que desea modificar y pincha el botón "Aceptar".	4- El sistema valida los datos.
	5- El sistema modifica los datos y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1- Si los datos están incorrectos el sistema lanza un mensaje de error y termina el caso de uso.
<b>Poscondiciones</b>	El sistema queda actualizado después de haber modificado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección "Eliminar"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Especialista AS selecciona la fila que desea eliminar y pincha "Eliminar".	2- El sistema debe mostrar un mensaje preguntando si realmente desea borrar la información. En caso positivo elimina la larvitrapa y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	El sistema queda actualizado después de haber eliminado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección "Visualizar"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Especialista AS accede a la opción "Mostrar Larvitrapas".	2- El Sistema muestra una pantalla para que el Especialista AS introduzca los datos de las larvitrapas que desea ver.
3- El Especialista del introduce los datos.	4- El sistema busca la larvitrapa con dichos criterios de búsqueda y la muestra al Especialista AS,

	finalizando así el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	

**Tabla 2.9 Descripción textual CUS Gestionar información de larvitrampas.**

<b>Caso de Uso</b>	Gestionar Ciclos.
<b>Actores</b>	Administrador nacional.
<b>Resumen</b>	El CU lo inicia el Administrador nacional cuando accede a la opción Gestionar (insertar, modificar, eliminar o visualizar) el período de tiempo que demora cada uno de los ciclos. El CU termina cuando el administrador accede a otra opción.
<b>Precondiciones</b>	El Administrador nacional debe tener permiso para acceder a la aplicación.
<b>Referencias</b>	R19
<b>Prioridad</b>	Crítica.
<b>Flujo Normal de Eventos</b>	
<b>Sección "Insertar"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Administrador nacional accede a la opción "Registrar Ciclos".	2- El Sistema muestra una interfaz para que el actor introduzca los datos.
3- El Administrador nacional introduce la fecha de inicio y fin del ciclo.	4- El sistema valida los datos.
	5- El sistema registra los datos y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1- Si las fechas están incorrectas, el sistema muestra un mensaje de error y termina el CU.
<b>Poscondiciones</b>	El sistema queda actualizado después de haber insertado los datos.

<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Administrador nacional selecciona la fila que desea eliminar y pincha “Eliminar”.	2- El sistema debe mostrar un mensaje preguntando si realmente desea borrar la información. En caso positivo elimina el ciclo y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	El sistema queda actualizado después de haber eliminado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Visualizar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Administrador nacional accede a la opción “Mostrar ciclos”.	2- El Sistema muestra una pantalla para que el Administrador nacional escoja el tipo de ciclo que desea visualizar.
3- El Administrador nacional escoge el ciclo.	4- El sistema busca los ciclos del tipo seleccionado y los muestras, finalizando así el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	

**Tabla 2.10 Descripción textual CUS Gestionar Ciclos.**

<b>Caso de Uso</b>	Gestionar Área de Salud-Ciclos.
<b>Actores</b>	Administrador nacional.
<b>Resumen</b>	El CU es inicializado por el Administrador nacional cuando accede a la opción de Gestionar (insertar, modificar, eliminar o visualizar) los ciclos que se asignan a las AS. El CU termina cuando este accede a otra opción.

<b>Precondiciones</b>	El Administrador nacional debe tener permiso para acceder a la aplicación.	
<b>Referencias</b>	R20	
<b>Prioridad</b>	Crítica.	
<b>Flujo Normal de Eventos</b>		
<b>Sección "Insertar"</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El Administrador nacional accede a la opción "Registrar Área de Salud-Ciclos."	2- El Sistema muestra una interfaz para que el actor introduzca los datos.	
1- El Administrador nacional introduce los datos.	2- El sistema valida los datos.	
	3- El sistema registra los datos y termina el CU.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	4.1- Si el usuario no selecciona ningún ciclo para algún AS, el sistema muestra un mensaje de error y termina el CU.	
<b>Poscondiciones</b>	El sistema queda actualizado después de haber insertado los datos.	
<b>Flujo Normal de Eventos</b>		
<b>Sección "Visualizar"</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El Administrador nacional accede a la opción "Mostrar área de salud - ciclos".	2- El Sistema muestra una pantalla para que el Administrador nacional introduzca el área de salud que desea ver.	
3- El Administrador nacional introduce los datos.	4- El sistema busca el área de salud con los ciclos que tiene asignados y lo muestra al Administrador nacional, finalizando así el CU.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<b>Poscondiciones</b>		

<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- Si el Administrador nacional desea modificar algún dato selecciona la fila y pincha el botón “Modificar”.	2- El Sistema muestra una nueva pantalla con los datos brindando la posibilidad de que el Administrador nacional cambie uno de los ciclos en los que se encuentra el área de salud.
3- El Administrador nacional introduce el ciclo que desea modificar y pincha el botón “Aceptar”.	4- El sistema valida los datos.
	5- El sistema modifica los datos y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1- Si los datos están incorrectos el sistema lanza un mensaje de error y termina el caso de uso.
<b>Poscondiciones</b>	El sistema queda actualizado después de haber modificado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Administrador nacional selecciona la fila que desea eliminar y pincha “Eliminar”.	2- El sistema debe mostrar un mensaje preguntando si realmente desea borrar la información. En caso positivo elimina el ciclo asignado al área de salud y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	El sistema queda actualizado después de haber eliminado los datos.

**Tabla 2.11 Descripción textual CUS Gestionar Área de Salud-Ciclos.**

<b>Caso de Uso</b>	Gestionar manzanas.	
<b>Actores</b>	Administrador AS.	
<b>Resumen</b>	El CU se inicia cuando el Administrador AS accede a la opción de insertar, modificar, eliminar o visualizar las manzanas de cada AS. El CU termina cuando este realiza cualquiera de estas opciones.	
<b>Precondiciones</b>	El Administrador AS debe tener permiso para acceder a esta funcionalidad.	
<b>Referencias</b>	R21	
<b>Prioridad</b>	Crítica.	
<b>Flujo Normal de Eventos</b>		
<b>Sección "Insertar"</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El Administrador AS accede a la opción "Configurar manzanas".	2- El Sistema muestra una interfaz para que el actor introduzca los datos.	
3- El Administrador AS introduce los datos de la manzana.	4- El sistema realiza la funcionalidad escogida por el usuario y valida los datos.	
	5- El sistema registra los datos y termina el CU.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	4.1- Si el usuario no introduce todos los datos o estos están incorrectos, el sistema muestra un mensaje de error y termina el CU.	
<b>Poscondiciones</b>	El sistema queda actualizado después de haber insertado los datos.	
<b>Flujo Normal de Eventos</b>		
<b>Sección "Modificar"</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- Si el Administrador AS desea modificar algún dato selecciona la fila y pincha el botón "Modificar".	2- El Sistema muestra una nueva pantalla con los datos brindando la posibilidad de que el Administrador AS	

	introduzca los nuevos datos.
3- El Administrador AS introduce los datos que desea modificar y pincha el botón "Actualizar".	4- El sistema valida los datos.
	5- El sistema modifica los datos y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1- Si el usuario no introduce todos los datos o estos están incorrectos, el sistema muestra un mensaje de error y termina el CU.
<b>Poscondiciones</b>	El sistema queda actualizado después de haber modificado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección "Eliminar"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Administrador AS selecciona la fila que desea eliminar y pincha "Eliminar".	2- El sistema debe mostrar un mensaje preguntando si realmente desea borrar la información. En caso positivo elimina la manzana y termina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	El sistema queda actualizado después de haber eliminado los datos.
<b>Flujo Normal de Eventos</b>	
<b>Sección "Visualizar"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Administrador AS accede a la opción "Mostrar manzanas".	6- El Sistema muestra una pantalla para que el Administrador AS introduzca los datos de las manzanas que desea ver.
3- El Administrador AS introduce los datos.	7- El sistema busca la manzana con dichos criterios de búsqueda y la muestra al Administrador AS,

	finalizando así el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Poscondiciones</b>	

**Tabla 2.12 Descripción textual CUS Gestionar manzanas.**

Los restantes casos de usos son secundarios, son similares a los explicados anteriormente, solo que se visualiza la información que se inserta en el AS, a distintos niveles. Se generan reportes y estadísticas en dependencia del nivel donde se encuentre el actor.

Para profundizar en el Modelo de Sistema ver [Anexo 2](#).

### **Conclusiones.**

En el capítulo se realizó un correcto modelado de los procesos que se desarrollan en la campaña de vigilancia y lucha antivectorial, permitiendo una mejor comprensión del negocio actual y logrando extraer los requerimientos con la finalidad de conseguir una futura aplicación que posea las funcionalidades necesarias. Se abordaron todos los aspectos relacionados con el software como los diagramas de casos de uso del sistema, y la descripción textual de cada uno de ellos para brindar un mejor entendimiento.

### **CAPÍTULO 3. DISEÑO DEL SISTEMA.**

En el actual capítulo se aborda un flujo de trabajo de gran importancia en el proceso de desarrollo de un software, que permitirá realizar una correcta implementación de la aplicación: análisis y diseño. La investigación en curso se centra principalmente en el diseño.

El diseño es un refinamiento del análisis mostrando como cumple el sistema sus objetivos, este constituye el centro de atención al final de la fase de elaboración. El diseño debe ser suficientemente claro para que el sistema pueda ser implementado sin ambigüedades.

Se presentará el diseño del software incluyendo las clases del diseño y los diagramas de interacción para la explicación de cada uno de los escenarios de los casos de uso.

#### **3.1 Modelo de diseño.**

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Contribuye a una arquitectura estable y sólida, y a crear un plano del modelo de implementación. Durante la fase de construcción, cuando la arquitectura es estable y los requisitos están bien entendidos, el centro de atención se desplaza a la implementación. (21)

El diseño de un software consiste en el proceso de aplicar distintas técnicas y principios con el propósito de definir un producto con los suficientes detalles como para permitir su realización física. (22) Dentro de este flujo de trabajo se obtiene el modelo de diseño que es un modelo de objetos que describe la realización física de los casos de uso.

##### **3.1.1 Patrones de diseño.**

Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias). (23) Es una pareja de problema/solución con un nombre, que codifica (estandariza) buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades.

Los patrones de diseño han contribuido a dar flexibilidad y extensibilidad a los diseños, ellos no sólo nombran, abstraen e identifican aspectos claves de estructuras comunes de diseño, sino que generalmente son descritos en una forma específica documental, haciendo su comprensión y aplicación fácil para el conjunto de desarrolladores.

Contribuyen a reutilizar diseño, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones. Mejoran la flexibilidad, modularidad y extensibilidad, factores internos e íntimamente relacionados con la calidad percibida por el usuario.

Los básicos se refieren a cuestiones y aspectos fundamentales del diseño. Con el objetivo de realizar un correcto diseño se aplicaron una serie de patrones para asignar las responsabilidades a las clases y objetos y diseñar la colaboración entre ellos.

Los **patrones GRASP** describen los principios fundamentales de la asignación de responsabilidades a objetos. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). (24)

El patrón **Experto**, como su nombre lo indica, asigna una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplirla. Por tanto al aplicarlo a cada clase le fueron asignadas las tareas que podían realizar según la información que poseían. Permite que se conserve el encapsulamiento, los objetos se valen de su propia información para hacer lo que se les pide.

Soporta un bajo acoplamiento, lo que favorece el hecho de tener sistemas más robustos y de fácil mantenimiento. El comportamiento se distribuye entre las clases que cuentan con la información requerida, definiendo clases "sencillas" y más cohesivas que son más fáciles de comprender y de mantener.

El patrón **Creador** posibilita asignarle la responsabilidad de crear una instancia de un objeto a la clase que tenga la información necesaria para inicializarlo. La creación de objetos es una de las actividades más frecuentes, en consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella. El diseño bien asignado puede soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y la reutilizabilidad.

Se hace necesario mantener un **Bajo Acoplamiento**, el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. Esto se aplica para evitar varios problemas, por ejemplo son más difíciles de reutilizar porque se requiere la presencia de otras clases de las que dependen.

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. La aplicación del patrón **Alta Cohesión** permite caracterizar a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme, o sea, evita asignar demasiadas responsabilidades a las clases. No convienen las clases con baja cohesión pues presentan los siguientes problemas: son difíciles de comprender, de reutilizar, de conservar y las afectan constantemente los cambios.

Otro patrón aplicado fue el **Controlador** que asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación.

### **3.1.2 Diagramas de clases del diseño.**

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones. Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados para mostrar lo que el sistema puede hacer (análisis), y como puede ser construido (diseño).

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Gráficamente, un diagrama de clases es una colección de nodos y arcos. Se utilizan para modelar la vista de diseño estática de un sistema.

Se obtuvo un diseño que basado en los patrones *GRASP* explicados en la sección anterior. Al aplicar el patrón MVC se definieron tres tipos de clases: las vistas, los modelos y las controladoras. El modelo se encarga de gestionar los datos y reglas de negocio. La vista muestra la información del modelo al usuario mediante los formularios. Estas clases interactúan con el modelo mediante la controladora, nunca de forma directa. La controladora gestiona las entradas del usuario. Se definieron las clases entidades que se utilizan para modelar información que posee una vida larga y que es a menudo persistente.

Al incorporar MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución. Si uno de los componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas.

Se incluyó la clase `MyController`, es una librería que encapsula funcionalidades las cuales pueden ser invocadas en cualquier momento por la clase controladora que se encuentre activa, pues todas las

clases de este tipo heredan de MyController, la que a su vez es hija de Controller, como establece el framework CodeIgniter que deben hacer todas las clases controladoras.

A continuación se muestran los diagramas de clases del diseño desarrollados en el módulo Vectores.

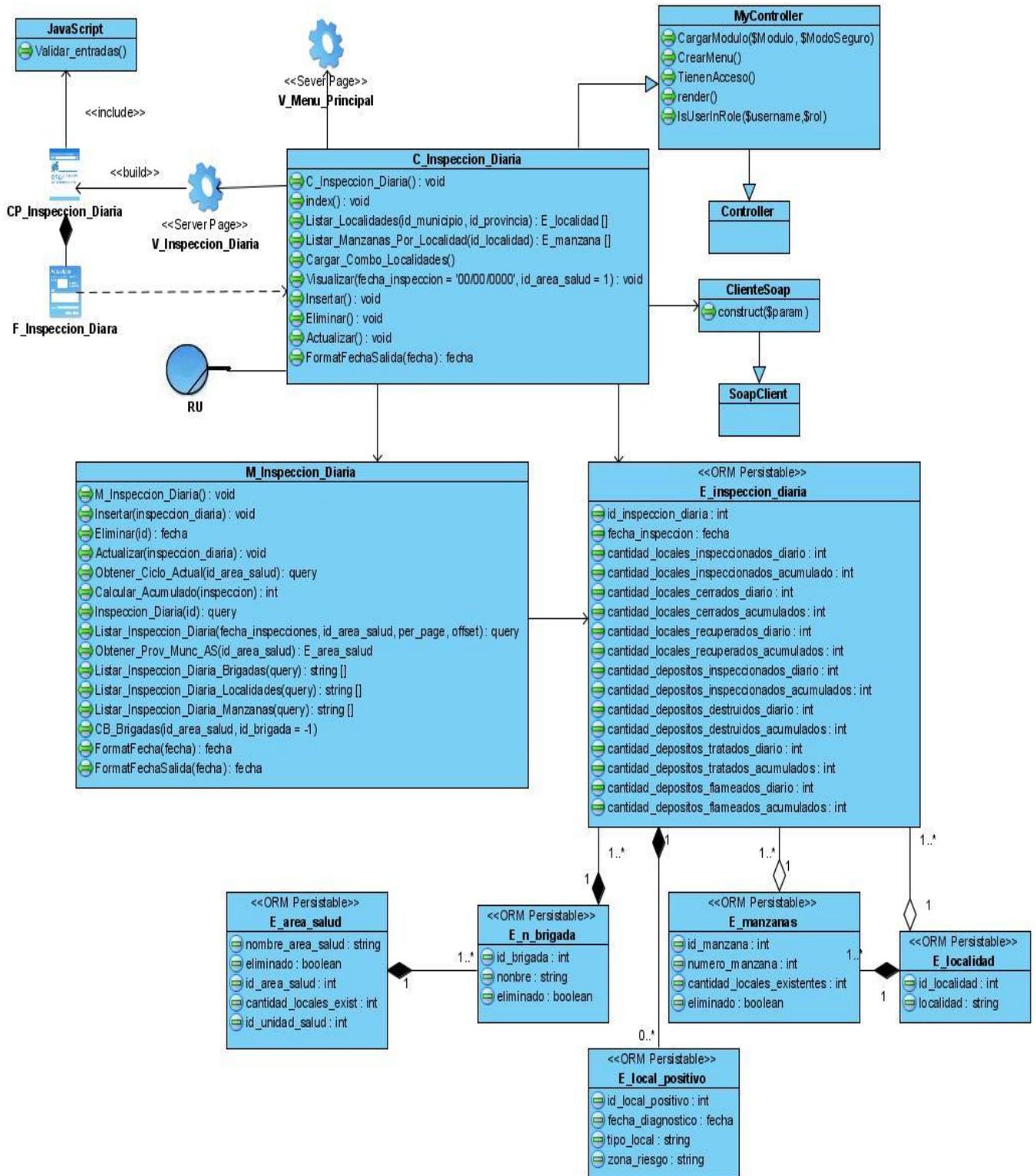


Figura 3.1 Diagrama de clases. Gestionar información de locales inspeccionados por brigadas.

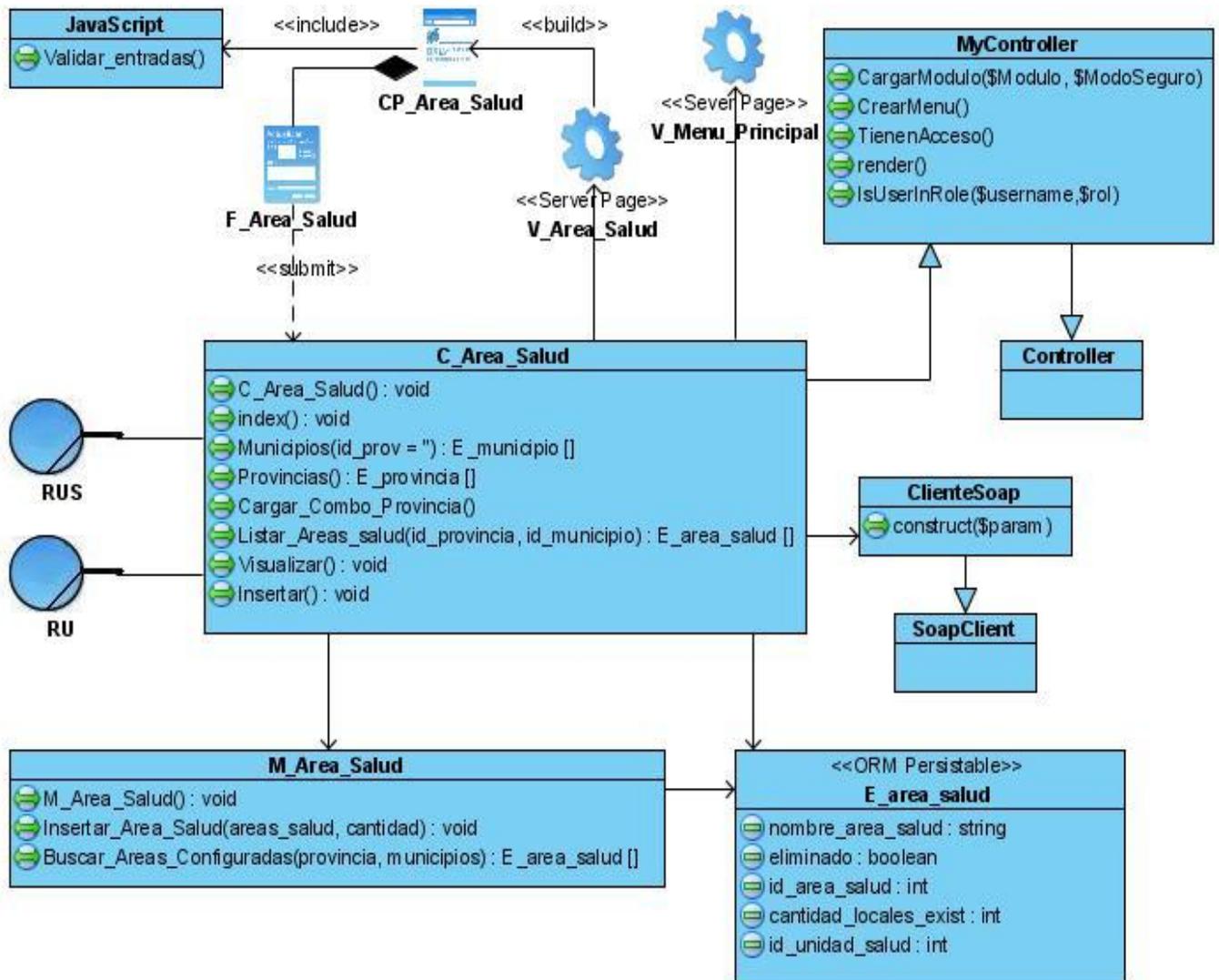


Figura 3.2 Diagrama de clases. Gestionar información de larvitrapas.

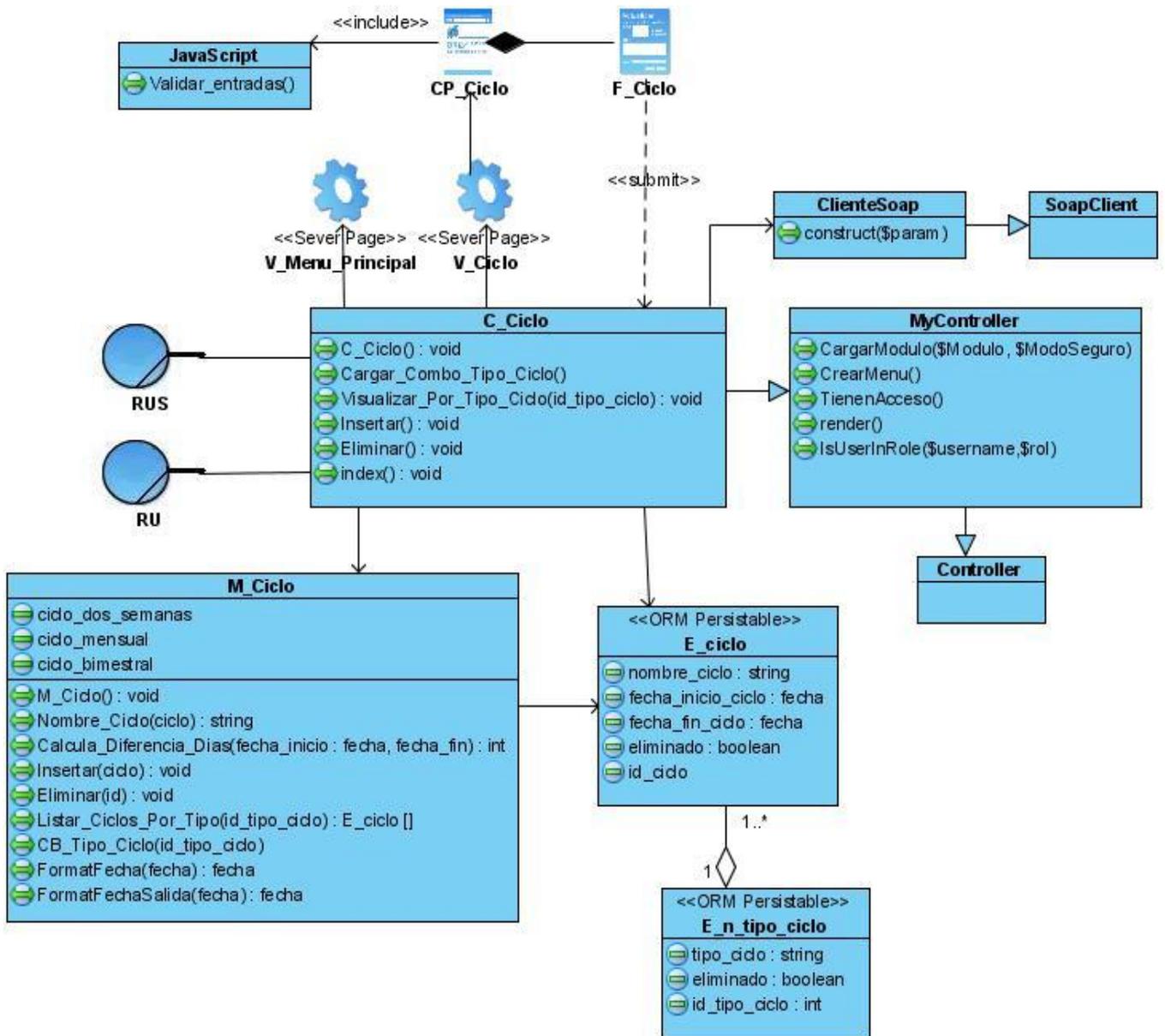


Figura 3.3 Diagrama de clases. Gestionar Ciclos.

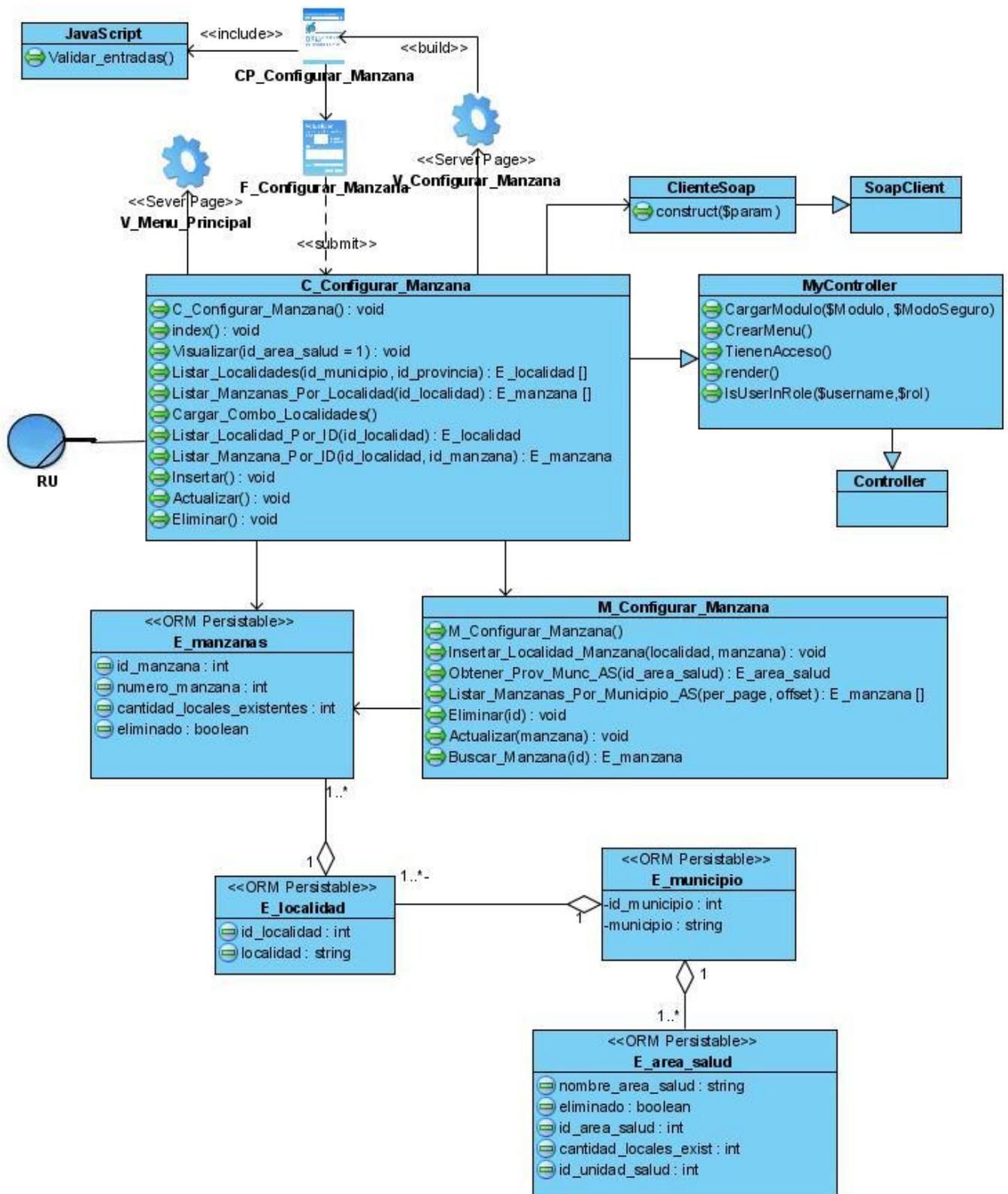


Figura 3.4 Diagrama de clases. Gestionar manzanas.

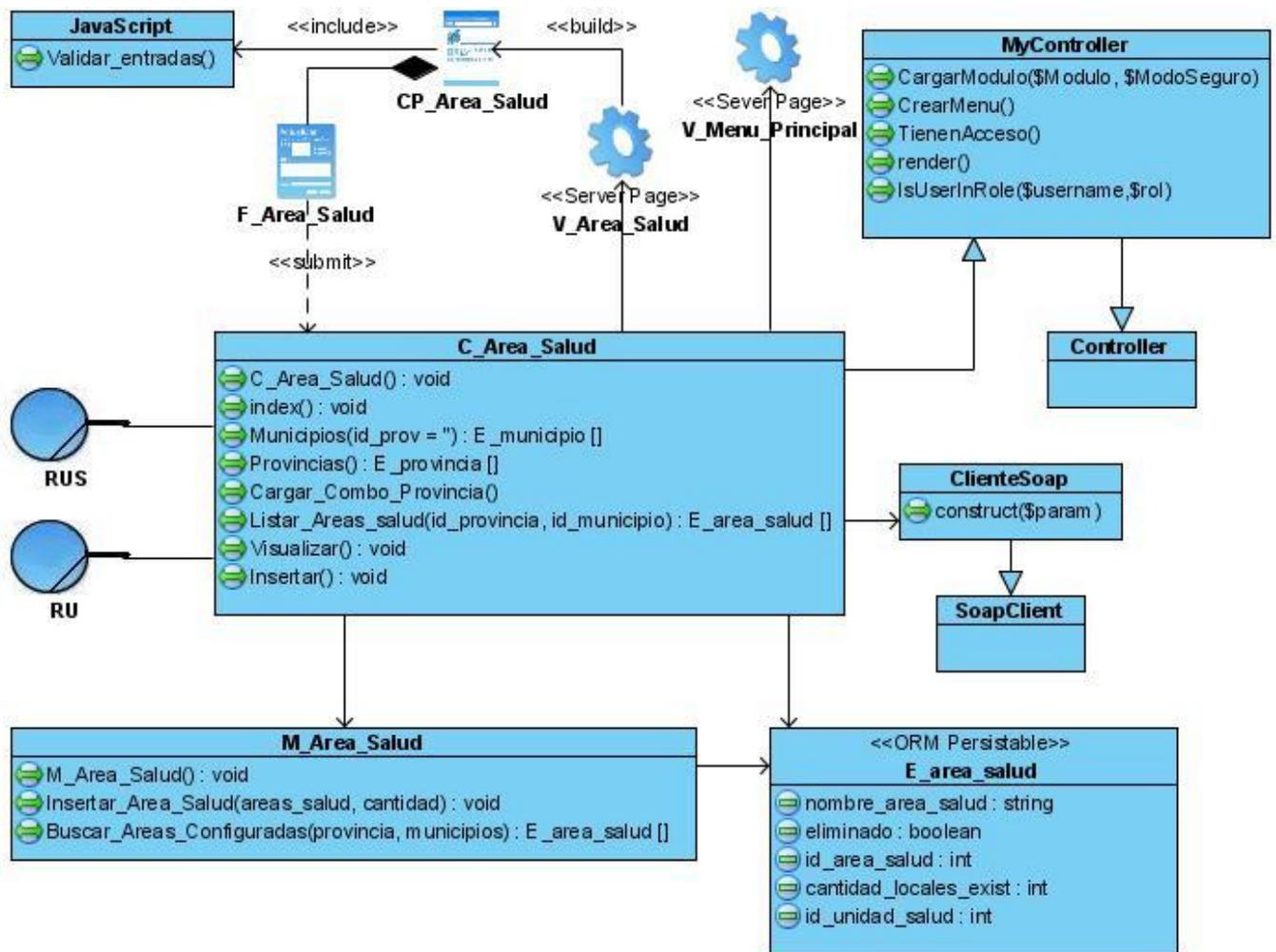


Figura 3.5 Diagrama de clases. Configurar Área de Salud.

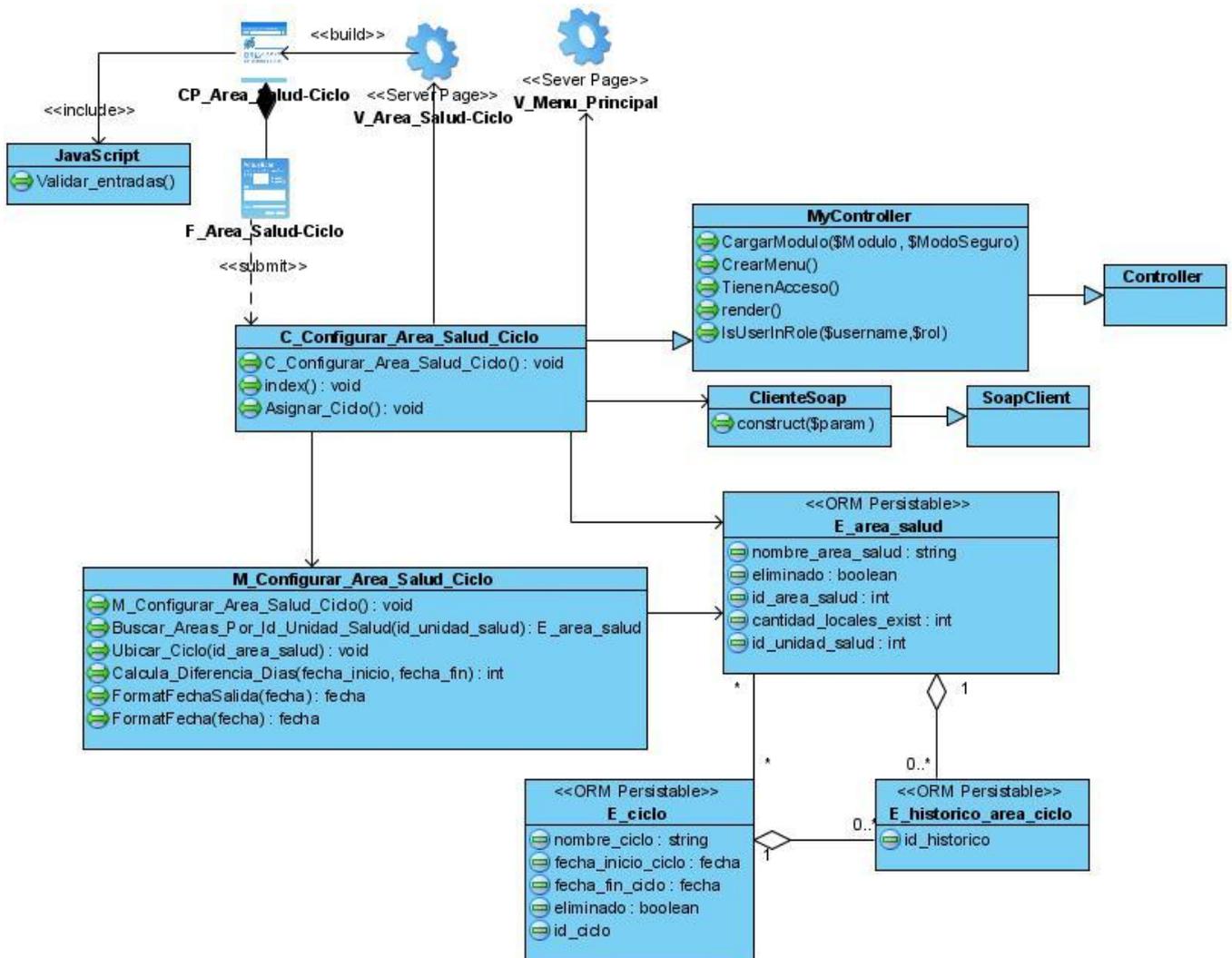


Figura 3.6 Diagrama de clases. Gestionar Área de Salud-Ciclos.

### 3.1.3 Diagramas de secuencia.

Los diagramas de secuencia y los diagramas de colaboración (ambos llamados diagramas de interacción) son dos de los cinco tipos de diagramas de UML que se utilizan para modelar los aspectos dinámicos de los sistemas. Un diagrama de interacción muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento.

Se presentan a continuación los diferentes diagramas de secuencia creados por escenarios para una mejor comprensión del flujo de mensajes.

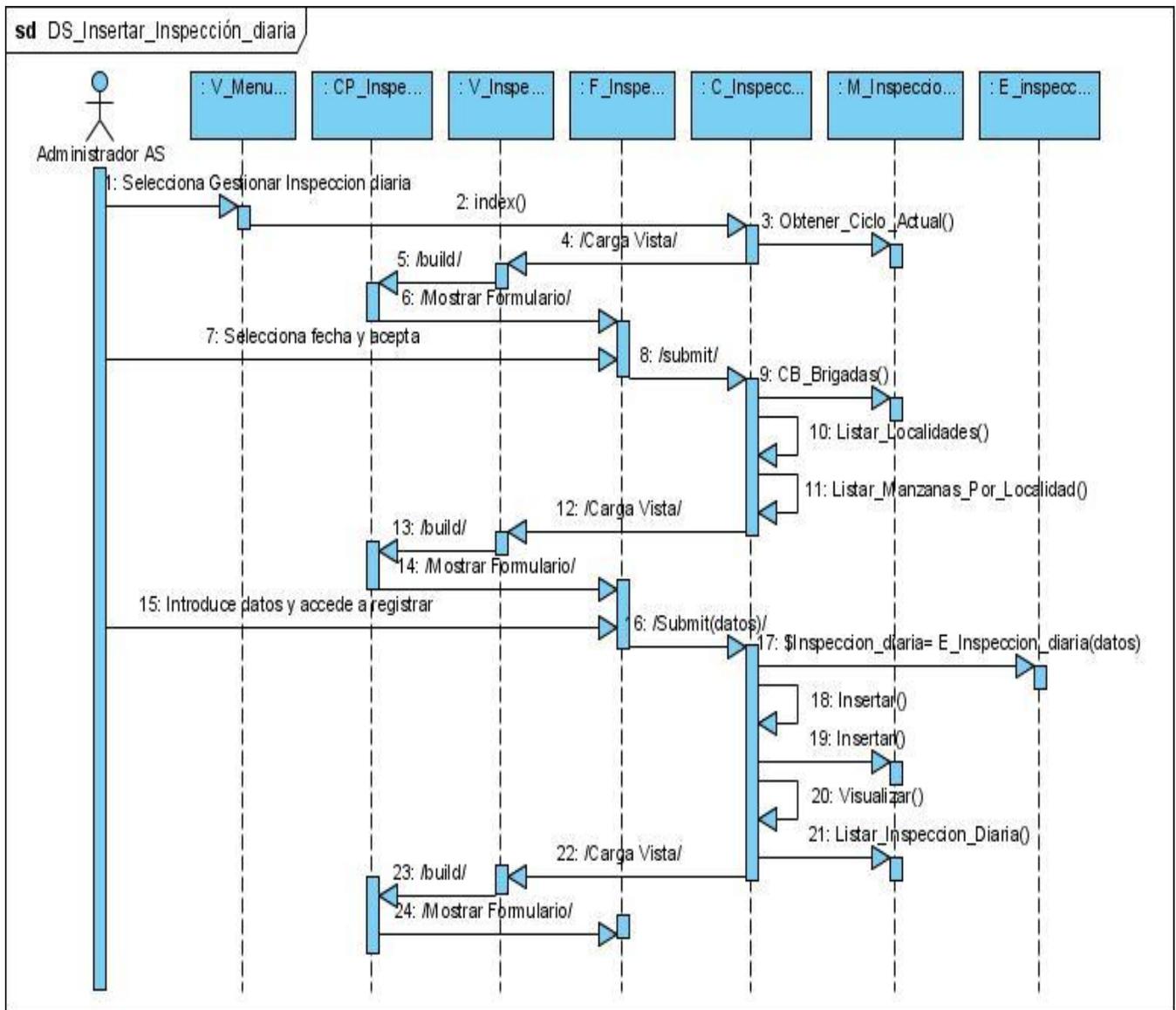


Figura 3.7 Diagrama de secuencia. Insertar información de locales inspeccionados por brigadas.

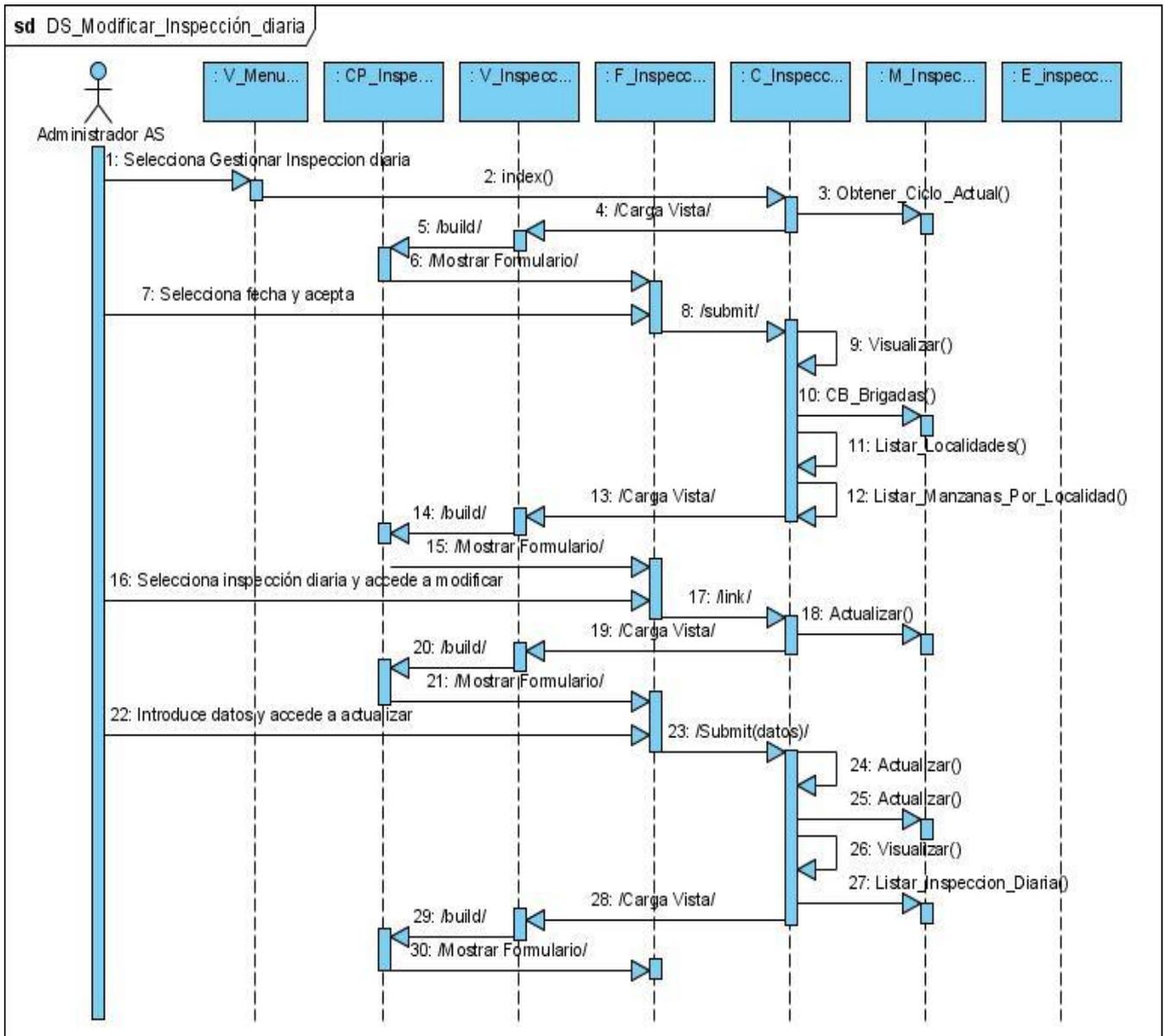


Figura 3.8 Diagrama de secuencia. Modificar información de locales inspeccionados por brigadas.

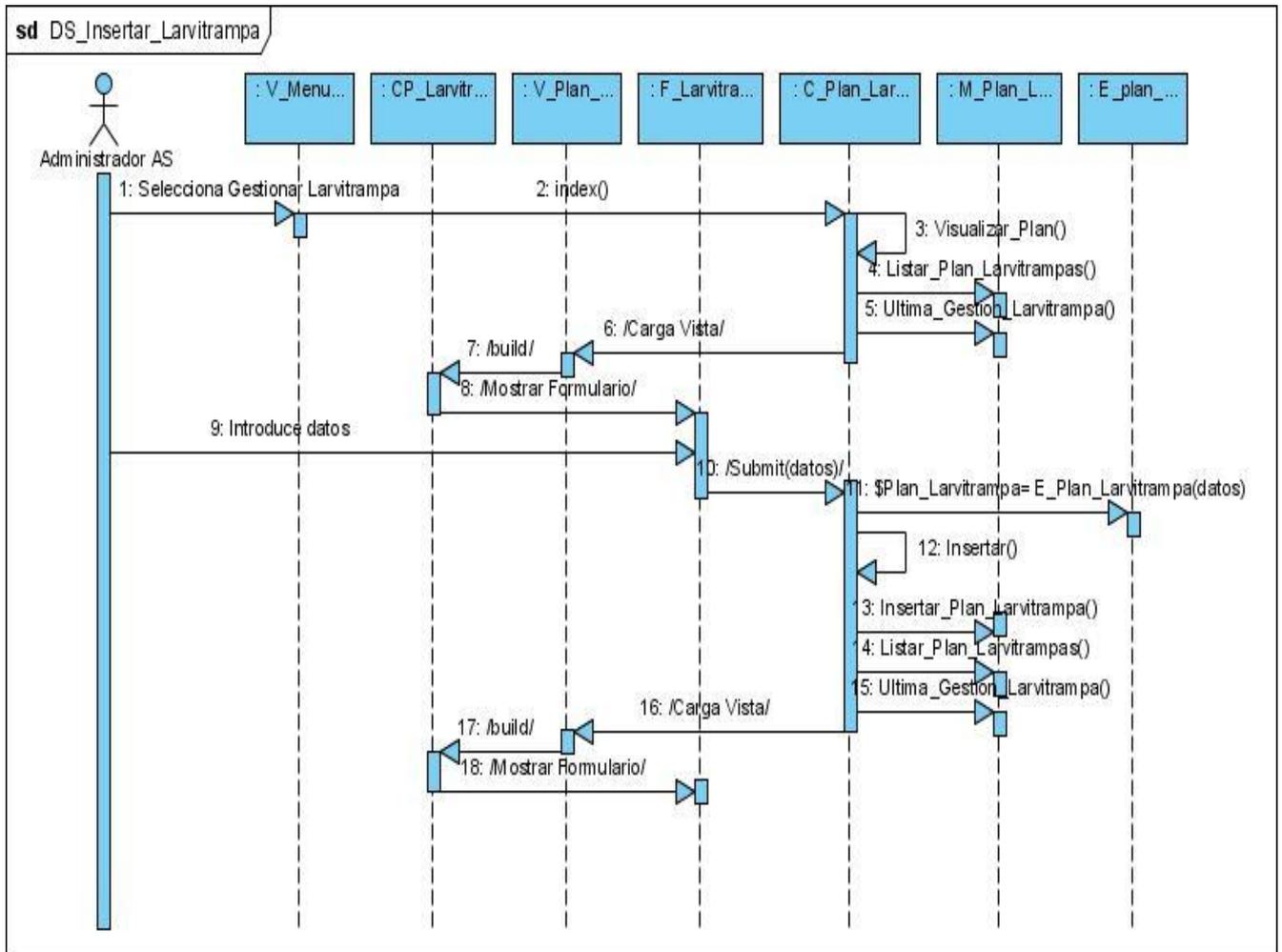


Figura 3.9 Diagrama de secuencia. Insertar plan de larvitrapa.

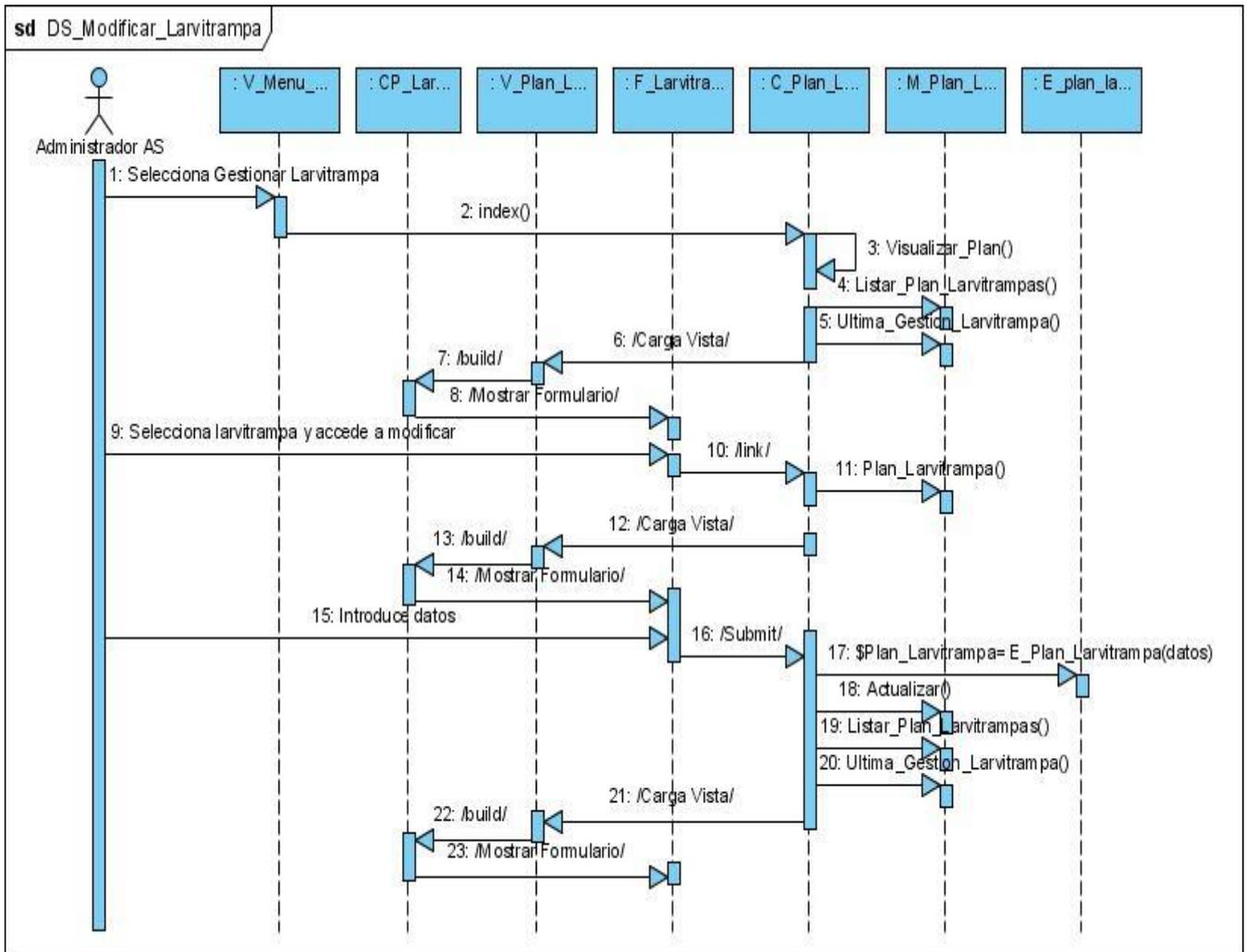


Figura 3.10 Diagrama de secuencia. Modificar plan de larvitrapa.

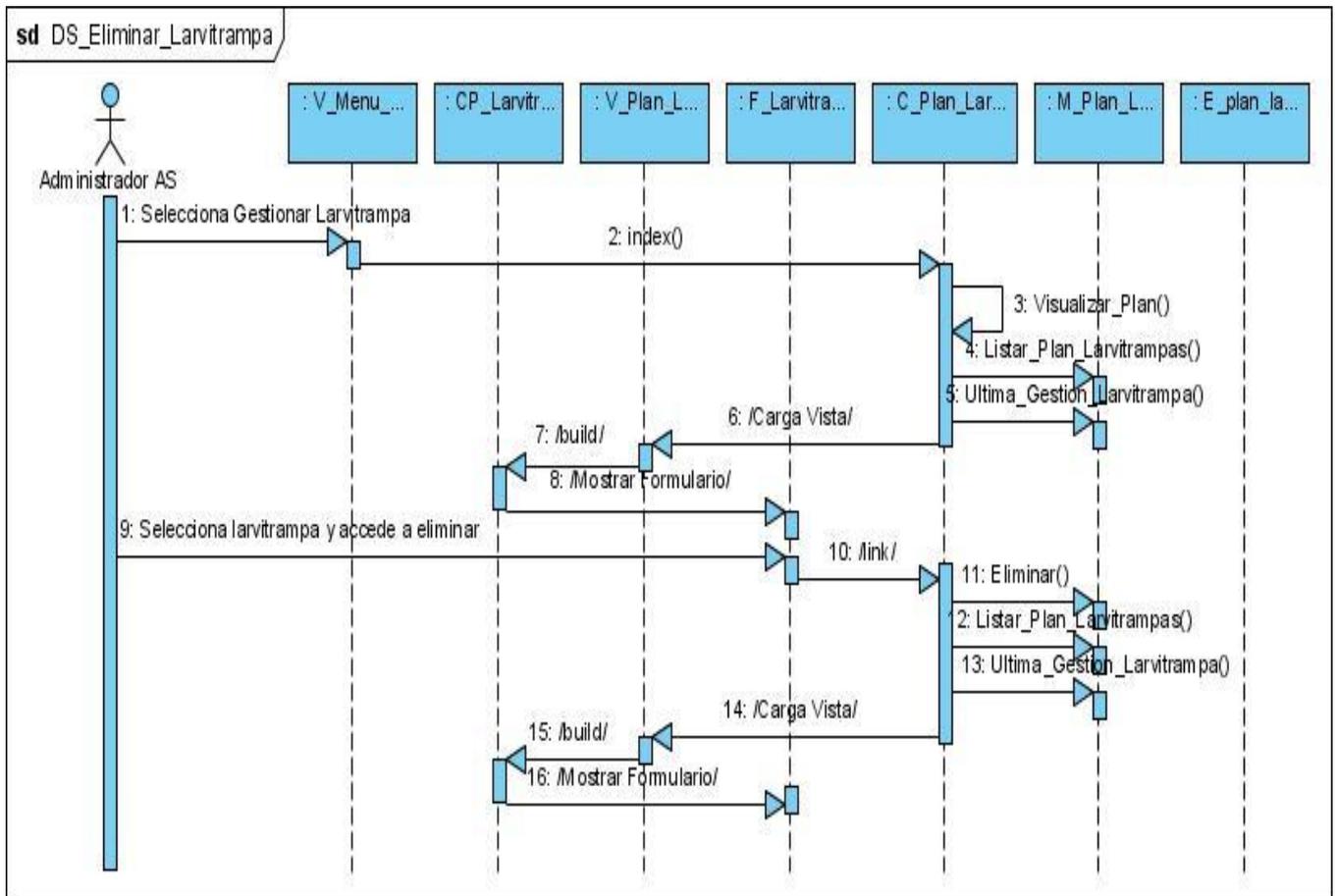


Figura 3.11 Diagrama de secuencia. Eliminar plan larvitrapa.

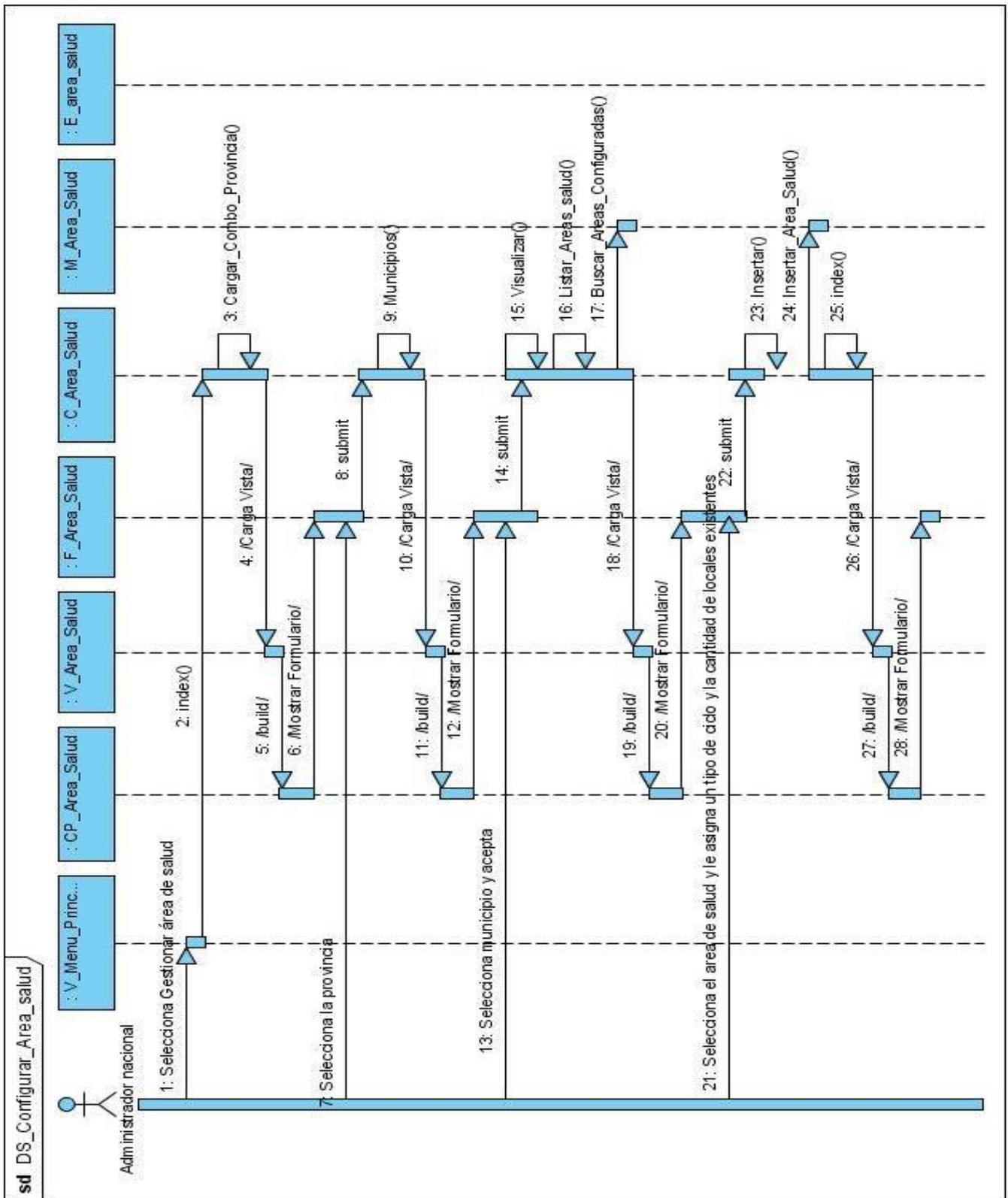


Figura 3.12 Diagrama de secuencia. Configurar manzana.

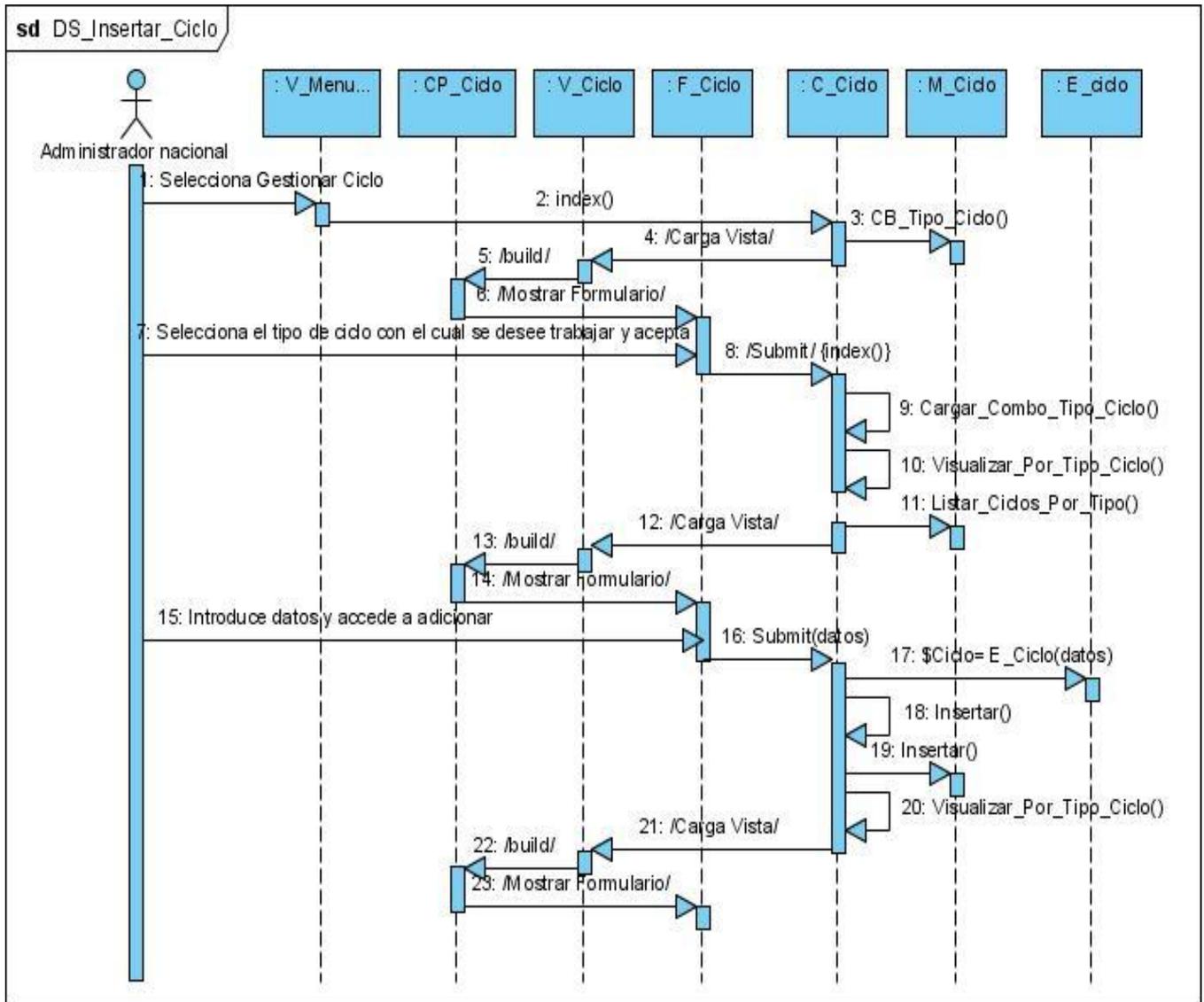


Figura 3.13 Diagrama de secuencia. Insertar ciclo.

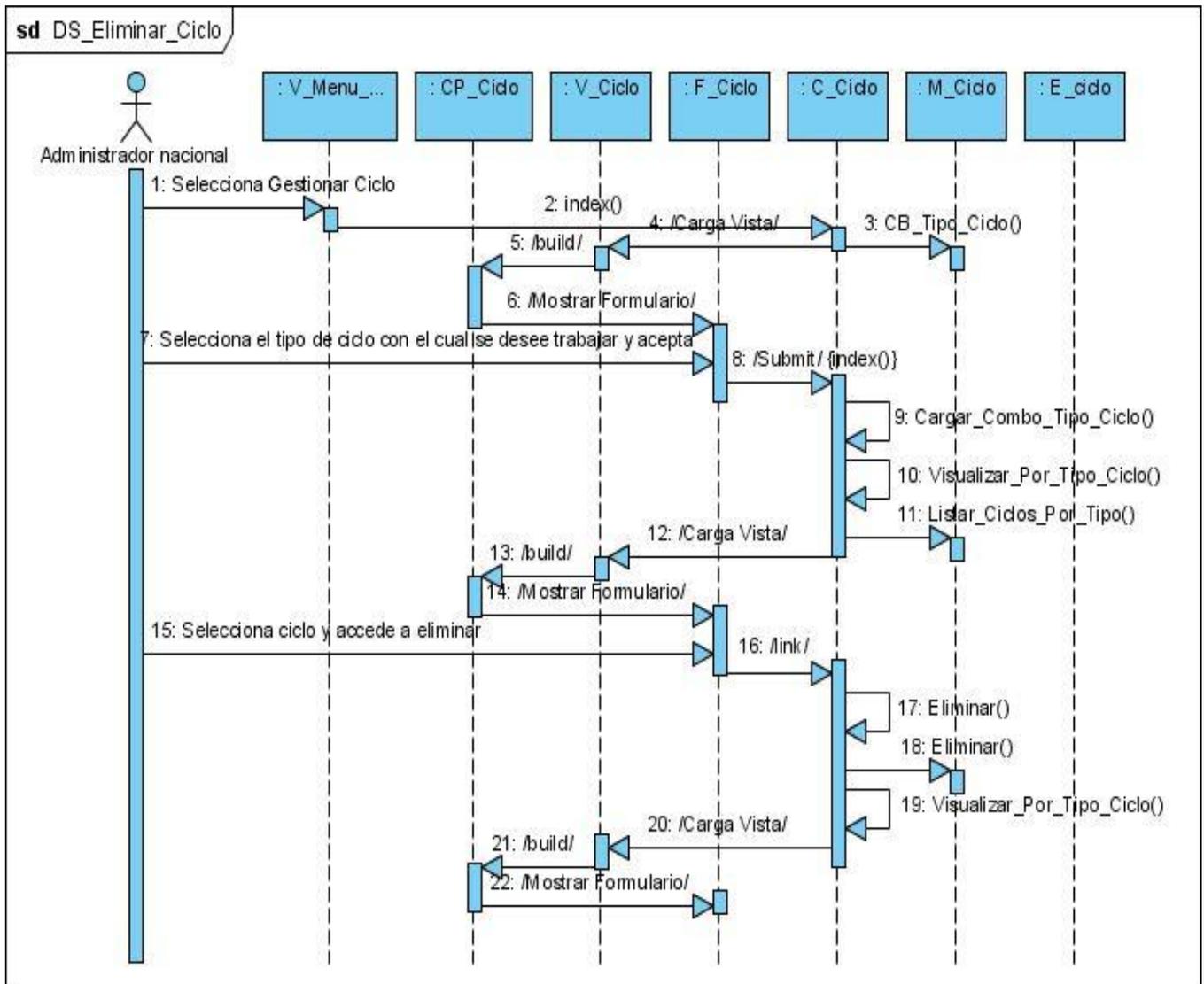


Figura 3.14 Diagrama de secuencia. Eliminar ciclo.

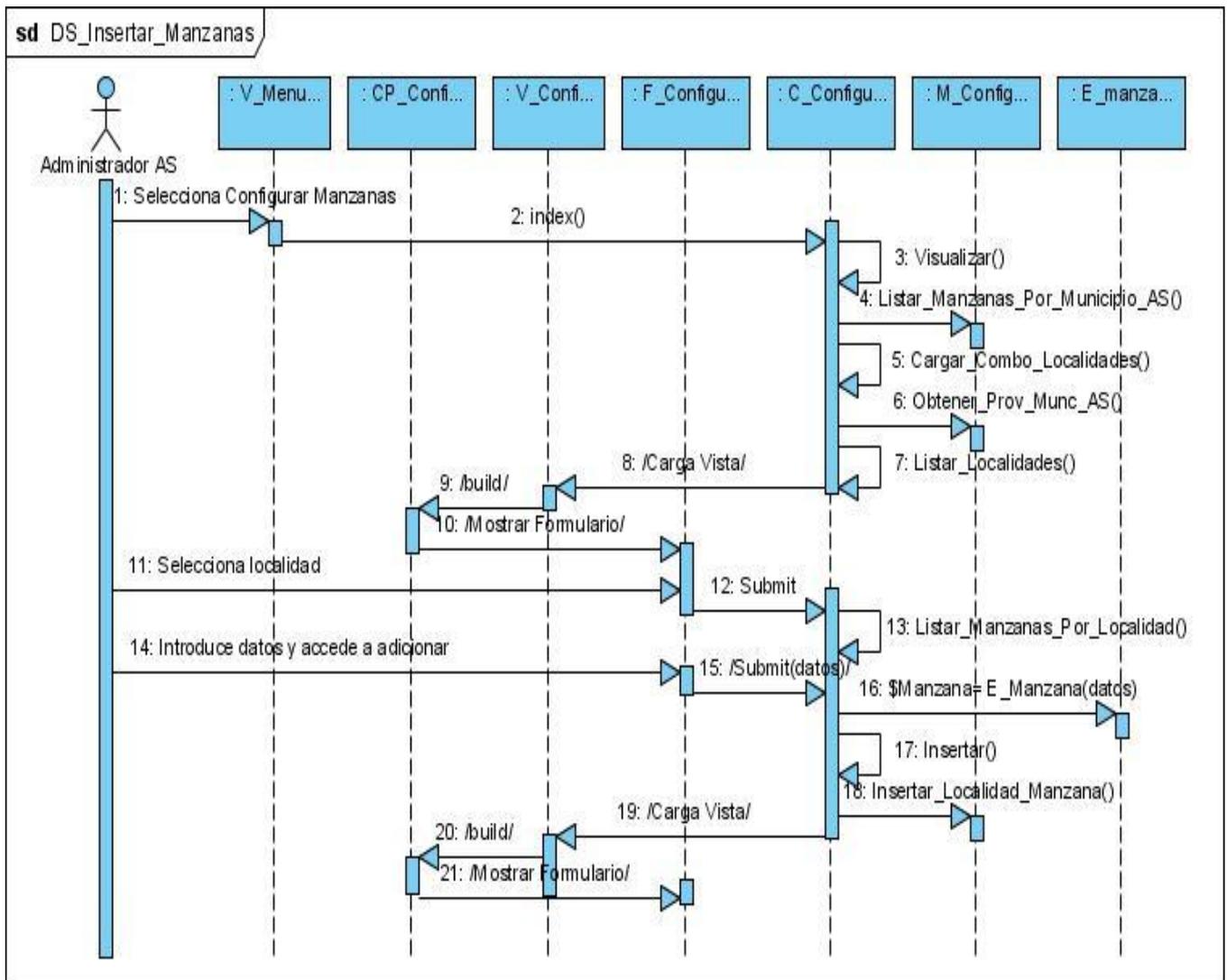


Figura 3.15 Diagrama de secuencia. Insertar manzana.

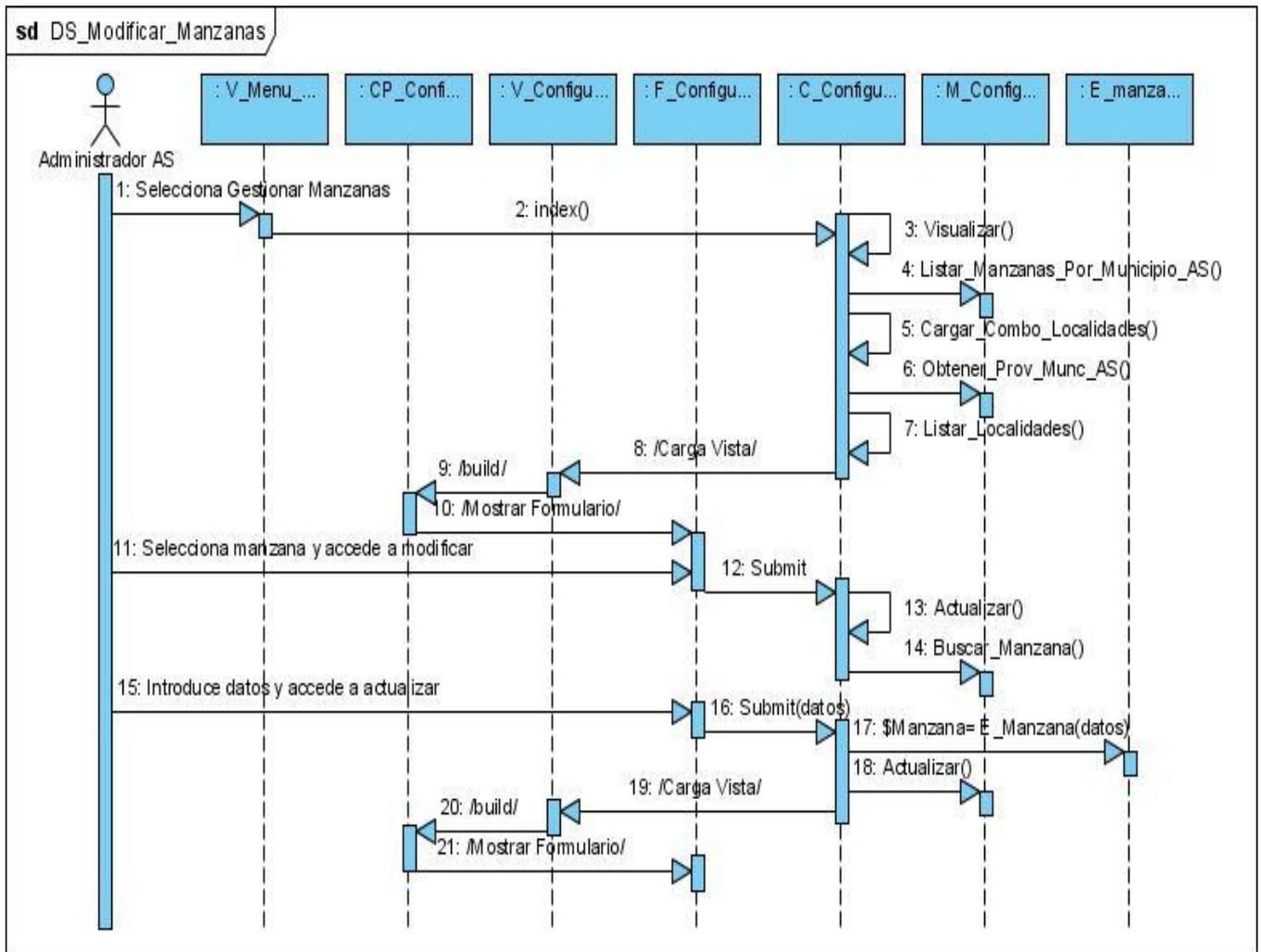


Figura 3.16 Diagrama de secuencia. Modificar manzana.

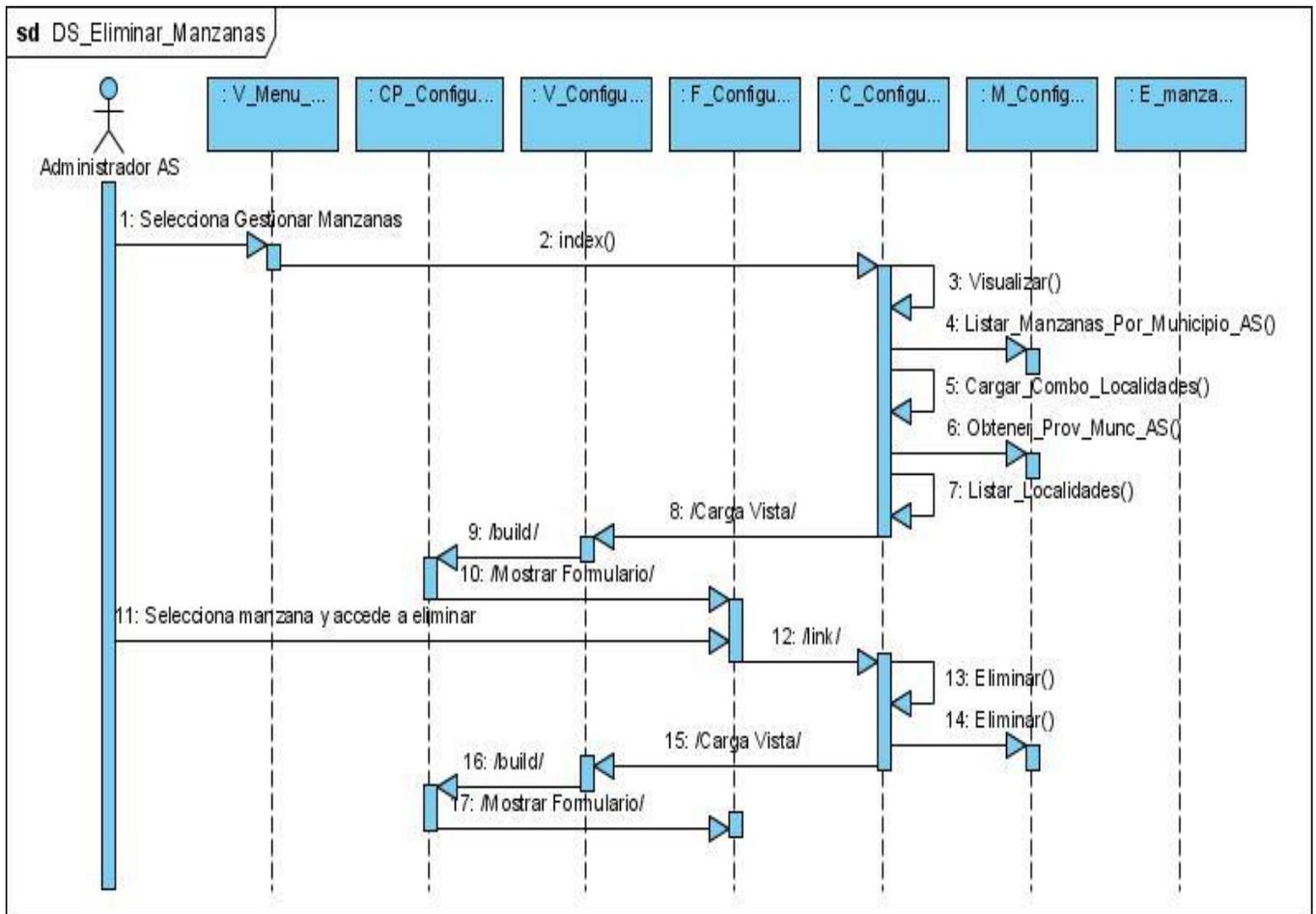


Figura 3.17 Diagrama de secuencia. Eliminar manzanas.

### **Conclusiones.**

Al terminar la etapa de diseño, se refinó el diagrama de clases y sus relaciones. Se presentaron los mensajes que intercambian los objetos para realizar las tareas necesarias, a través de los diagramas de interacción que reflejan la colaboración entre las clases.

Se obtuvo un modelo de diseño que contiene las clases del diseño con los atributos y métodos que permitirán a los implementadores generar gran parte del código.

Con el desarrollo de los diferentes artefactos se ha logrado un diseño preparado para la implementación del software.

**CONCLUSIONES**

Con la investigación se ha obtenido el diseño de una aplicación informática que una vez implementada gestionará los procesos fundamentales del programa de vigilancia y lucha antivectorial en Cuba. Se le proporcionó una adecuada solución a los objetivos y tareas propuestas y se pudieron alcanzar los siguientes resultados:

- Se efectuó un estudio detallado de los procesos desarrollados dentro del Programa de Vigilancia y Lucha Antivectorial en Cuba que permitió comprender el funcionamiento de dichos procesos.
- Se realizó un estudio de los sistemas informatizados existentes vinculados al campo de acción de la investigación, determinando que ninguno cumplía con las necesidades de funcionamiento establecidas por los clientes.
- Se realizó el diseño de la aplicación que facilitó a los programadores realizar la implementación del sistema propuesto.

Al poner en funcionamiento la aplicación se agilizarán los procesos de gestión de la información correspondiente al control de focos de mosquitos *Aedes Aegypti* en el país y se obtendrán una serie de reportes y estadísticas de gran importancia epidemiológicas, permitiendo la toma adecuada de decisiones ante cualquier índice de infestación, para evitar la propagación de enfermedades transmisibles, incrementando así los niveles de salud y satisfacción de la población cubana.

**RECOMENDACIONES**

Para alcanzar un mejor funcionamiento en una fase superior se especifican las siguientes recomendaciones:

- Realizar varias iteraciones para lograr el perfeccionamiento del diseño y así adquirir soluciones más óptimas.
- Para próximas versiones, desarrollar otras funcionalidades relacionadas al proceso de gestión de la información relacionada con el control de otros vectores como roedores, cucarachas y moscas.
- Desarrollar otras funcionalidades correspondientes al trabajo de la Campaña Antivectorial, como la gestión de la información de las radio-batidas y el control de los recursos materiales utilizados en dicha campaña.
- Incluir trabajo con cartografía para la localización de regiones positivas al mosquito *Aedes Aegypti*.
- Incrementar el número de reportes estadísticos que se generan para controlar eficientemente los focos de mosquitos.

**REFERENCIAS BIBLIOGRÁFICAS**

1. Perez, Graells. Las TIC y sus aportaciones a la sociedad. [En línea] 2000. [Citado el: 28 de Octubre de 2007.]  
<http://dewey.uab.es/PMARQUES/tic.htm>.
2. Delgado Ramos, Ariel y Vidal Ledo, María. Infomática en Salud Pública Cubana. [En línea] 2006. [Citado el: 1 de noviembre de 2007.]  
[http://bvs.sld.cu/revistas/spu/vol32\\_3\\_06/spu15306.htm](http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm).
3. Aedes Aegypti. [En línea] [Citado el: 14 de Noviembre de 2007.]  
<http://www.inan.gov.py/articulo.php?ID=48>.
4. Plataforma Estatal de Asociaciones del Profesorado de Tecnología. [En línea] 2007. [Citado el: 23 de Enero de 2008.]  
<http://www.sialatecnologia.org/tecnologia.php>.
5. IBERMACHINES. Sistemas y redes de código abierto. [En línea] 2008. [Citado el: 22 de Enero de 2008.]  
<http://www.ibermachines.com/servidor-web.html>.
6. Sistemas de Gestión de Bases de Datos. [En línea] Agosto de 2004. [Citado el: 1 de Marzo de 2008.]  
[http://www.igac.gov.co:8080/igac\\_web/UserFiles/File/ciaf/TutorialSIG\\_2005\\_26\\_02/paginas/ctr\\_sistema\\_sdegestiondebasededatos.htm](http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/ctr_sistema_sdegestiondebasededatos.htm).
7. Métodos ágiles en el desarrollo de software. [En línea] 17 de octubre de 2006. [Citado el: 5 de Noviembre de 2007.]  
<http://amalgamadeletras.blogspot.com/2006/10/mtodos-giles-en-el-desarrollo-de.html>.
8. Canos, Jose H, Penades, Ma. del Carmen y Letelier, Patricio. Metodologías Ágiles en el Desarrollo de Software. [En línea] [Citado el: 25 de Noviembre de 2007.]  
<http://www.willydev.net/descargas/prev/TodoAgil.Pdf>.
9. Qualitrain. [En línea] 2001. [Citado el: 3 de Diciembre de 2007.]  
<http://www.qualitrain.com.mx/index.php/Procesos-de-software/Metodologias-agiles-de-desarrollo-tercera-parte.html>.
10. ¿Qué es Scrum? [En línea] [Citado el: 15 de Noviembre de 2007.]  
[http://www.baufest.com/spanish/scrum/scrumconference2006/Que\\_es\\_scrum.pdf](http://www.baufest.com/spanish/scrum/scrumconference2006/Que_es_scrum.pdf).
11. Metodologías Ágiles. [En línea] 2008. [Citado el: 25 de Octubre de 2007.]  
<http://www.itbuilder.com.mx/blogs/fabiola.soto/post/Metodologias-Agiles.aspx>.
12. GenBIZ. [En línea] [Citado el: 22 de Febrero de 2008.]  
<http://www.puntogen.cl/implementacion.html>.
13. Gómez, Karel. Documento de Arquitectura de Software. Facultad 7. Habana : s.n., 2008.

14. Patrones y Antipatrones: una Introducción - Parte II. [En línea] 2008. [Citado el: 12 de Febrero de 2008.]  
[http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_3317/default.aspx](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317/default.aspx).
15. Herramientas Case. [En línea] [Citado el: 3 de Diciembre de 2007.]  
<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
16. Sitio de descargas de software. [En línea] [Citado el: 15 de Enero de 2007.]  
[http://www.freedownloadmanager.org/es/downloads/paradigma\\_uml\\_libre/](http://www.freedownloadmanager.org/es/downloads/paradigma_uml_libre/).
17. Identificación de procesos de negocio. [En línea] 2004. [Citado el: 13 de Noviembre de 2007.]  
<http://www.cujae.edu.cu/ediciones/Revistas/Industrial/Vol-XXV/3-2004/83-88%20Identificaci%C3%B3n%20de%20procesos.pdf>.
18. Especificación de Requerimientos Parte 1. [En línea] 2008. [Citado el: 10 de Mayo de 2008.]  
Disponible en:  
[https://www.u-cursos.cl/ingenieria/2007/1/IN55A/1/material\\_docente/objeto/119018](https://www.u-cursos.cl/ingenieria/2007/1/IN55A/1/material_docente/objeto/119018).
19. Ortiz, Yunaysy. Especificacion de Requisitos 1.1.Vectores. Habana : s.n., 2008.
20. Administración de requerimientos con casos de usos. [En línea] 2000-2008. [Citado el: 12 de Enero de 2008.]  
<http://www.milestone.com.mx/CursoAdmReqCasosDeUso.htm>.
21. Disciplina Análisis y Diseño. La Habana : s.n., 2008.
22. El diseño de software. [En línea] 2004. [Citado el: 2 de Mayo de 2008.]  
<http://www.info-ab.uclm.es/asignaturas/42579/pdf/01-Capitulo1.pdf>.
23. Montovani, Pablo y Piñero, Álvaro. Patrones de diseño. [En línea] [Citado el: 2 de Mayo de 2008.]  
<http://dc.exa.unrc.edu.ar/nuevodc/materias/sistemas/2007/Patrones/1181572570/Singleton-FactoryMethod1.ppt>.
24. Saavedra Gutierrez, Jorge A. El mundo informático. [En línea] [Citado el: 10 de Marzo de 2008.]  
<http://jorgesaavedra.wordpress.com/2007/05/08/patrones-grasp-patrones-de-software-para-la-asignacion-general-de-responsabilidadparte-ii/>.

**BIBLIOGRAFÍA**

- Administración de requerimientos con casos de usos. [En línea] 2000-2008. [Citado el: 12 de Enero de 2008.]  
<http://www.milestone.com.mx/CursoAdmReqCasosDeUso.htm>.
- Aedes Aegypti. [En línea] [Citado el: 14 de Noviembre de 2007.]  
<http://www.inan.gov.py/articulo.php?ID=48>.
- Canos, Jose H, Penades, Ma. del Carmen y Letelier, Patricio. Metodologías Ágiles en el Desarrollo de Software. [En línea] [Citado el: 25 de Noviembre de 2007.]  
<http://www.willydev.net/descargas/prev/TodoAgil.Pdf>.
- Code Java. [En línea] [Citado el: 28 de Marzo de 2008.]  
<http://www.codejava.org/?idxpagina=9&idxnota=32795&destacada=1>.
- de la Torre, Aníbal. Lenguajes del lado servidor o cliente. [En línea] 2006. [Citado el: 5 de Marzo de 2008.]  
[http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html)
- Delgado Ramos, Ariel y Vidal Ledo, María. Infomática en Salud Pública Cubana. [En línea] 2006. [Citado el: 1 de noviembre de 2007.]  
[http://bvs.sld.cu/revistas/spu/vol32\\_3\\_06/spu15306.htm](http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm)
- Disciplina Análisis y Diseño. La Habana : s.n., 2008.
- El concepto de los Sitios Web. [En línea] 2008. [Citado el: 3 de Diciembre de 2007.]  
<http://www.informaticamilenium.com.mx/paginas/espanol/preguntas/concepto.htm>.
- Especificación de Requerimientos Parte 1. [En línea] 2008. [Citado el: 10 de Mayo de 2008.]  
[https://www.u-cursos.cl/ingenieria/2007/1/IN55A/1/material\\_docente/objeto/119018](https://www.u-cursos.cl/ingenieria/2007/1/IN55A/1/material_docente/objeto/119018).
- Gallego Vázquez, José Antonio. Desarrollo web con PHP y MySQL. Madrid : Ediciones Anaya Multimedia, 2003.
- GenBIZ. [En línea] [Citado el: 22 de Febrero de 2008.]  
<http://www.puntogen.cl/implementacion.html>.
- Gómez, Karel. Documento de Arquitectura de Software. Facultad 7. Habana : s.n., 2008.
- Guía Breve de Servicios Web. [En línea] 9 de Enero de 2008. [Citado el: 23 de Marzo de 2008.]  
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
- Herramientas Case. [En línea] [Citado el: 3 de Diciembre de 2007.]  
<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
- IBERMACHINES. Sistemas y redes de código abierto. [En línea] 2008. [Citado el: 22 de Enero de 2008.]

<http://www.ibermachines.com/servidor-web.html>.

- Identificación de procesos de negocio. [En línea] 2004. [Citado el: 13 de Noviembre de 2007.]  
<http://www.cujae.edu.cu/ediciones/Revistas/Industrial/Vol-XXV/3-2004/83-88%20Identificaci%C3%B3n%20de%20procesos.pdf>.
- Integración del Visual Paradigm. [En línea] [Citado el: 2 de Febrero de 2004.]  
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=vp>.
- Introducción a los servicios web y Microsoft .Net. [En línea] 9 de Julio de 2002. [Citado el: 15 de Marzo de 2008.]  
<http://geneura.ugr.es/~jmerelo/ws/>.
- Lago, Ramiro. Patrón "Modelo-Vista-Controlador". [En línea] Abril de 2007. [Citado el: 18 de Mayo de 2008.]  
<http://www.proactiva-calidad.com/java/patrones/mvc.html>.
- Lenguajes de Programación. [En línea] 2006. [Citado el: 17 de Abril de 2008.]  
<http://www.lenguajes-de-programacion.com/programacion-web.shtml>.
- Lenguajes de programación que deberías aprender. [En línea] 2007. [Citado el: 15 de Mayo de 2008.]  
<http://www.tufuncion.com/diferentes-lenguajes-programacion>.
- Model-View-Controller. [En línea] Sun Microsystems, 2002. [Citado el: 20 de Abril de 2008.]  
<http://java.sun.com/blueprints/patterns/MVC.html>.
- Montovani, Pablo y Piñero, Alvaro. Patrones de diseño. [En línea] [Citado el: 2 de Mayo de 2008.]  
<http://dc.exa.unrc.edu.ar/nuevodic/materias/sistemas/2007/Patrones/1181572570/Singleton-FactoryMethod1.ppt>.
- Ortiz, Yunaysy. Especificación de Requisitos 1.1.Vectores. Habana : s.n., 2008.
- Otras metodologías ágiles. [En línea] [Citado el: 3 de Marzo de 2008.]  
[http://www.xelphos.com.ar/xop\\_ma\\_otras.html](http://www.xelphos.com.ar/xop_ma_otras.html).
- Patrones y Antipatrones: una Introducción - Parte II. [En línea] 2008. [Citado el: 12 de Febrero de 2008.]  
[http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_3317/default.aspx](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317/default.aspx).
- Perez, Graells. Las TIC y sus aportaciones a la sociedad. [En línea] 2000. [Citado el: 28 de Octubre de 2007.]  
<http://dewey.uab.es/PMARQUES/tic.htm>.
- Plataforma Estatal de Asociaciones del Profesorado de Tecnología. [En línea] 2007. [Citado el: 23 de Enero de 2008.]

- <http://www.sialatecnologia.org/tecnologia.php>.
- ¿Qué es la entomología? [En línea] 1995. [Citado el: 20 de Mayo de 2008.]  
[http://scriptusnaturae.8m.com/II\\_ento/entomologia.htm](http://scriptusnaturae.8m.com/II_ento/entomologia.htm).
  - ¿Qué es Scrum? [En línea] [Citado el: 15 de Noviembre de 2007.]  
[http://www.baufest.com/spanish/scrum/scrumconference2006/Que\\_es\\_scrum.pdf](http://www.baufest.com/spanish/scrum/scrumconference2006/Que_es_scrum.pdf).
  - Rational Rose Enterprise. [En línea] [Citado el: 15 de Abril de 2008.]  
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
  - Requerimientos Técnicos y Funcionales de acuerdo al ISO 12207 . [En línea] 2008. [Citado el: 5 de Mayo de 2008.]  
<http://www.scribd.com/doc/202538/F1-Requerimientos-Tecnicos-y-Funcionales-de-acuerdo-al-ISO-12207>.
  - Saavedra Gutierrez, Jorge A. El mundo infomático. [En línea] [Citado el: 10 de Marzo de 2008.]  
<http://jorgesaaavedra.wordpress.com/2007/05/08/patrones-grasp-patrones-de-software-para-la-asignacion-general-de-responsabilidadparte-ii/>.
  - Sistemas de Gestión de Bases de Datos. [En línea] Agosto de 2004. [Citado el: 1 de Marzo de 2008.]  
[http://www.igac.gov.co:8080/igac\\_web/UserFiles/File/ciaf/TutorialSIG\\_2005\\_26\\_02/paginas/ctr\\_sistema\\_sede\\_gestion\\_de\\_bases\\_de\\_datos.htm](http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/ctr_sistema_sede_gestion_de_bases_de_datos.htm).
  - Tecnología. [En línea] [Citado el: 18 de Enero de 2008.]  
<http://www.elpais.com/tecnologia/>.
  - Up to Down. [En línea] 2008. [Citado el: 17 de Mayo de 2008.] <http://php.uptodown.com/>.
  - Visual Paradigm. [En línea] [Citado el: 2 de Octubre de 2007.]  
<http://www.visual-paradigm.com/product/vpum/>.
  - Welling, Luke y Thomson, Laura. Desarrollo web con PHP y MySQL. s.l. : Anaya Multimedia.

ANEXOS

Anexo 1. Modelo de negocio.

Diagramas de actividades de los casos de uso del negocio.

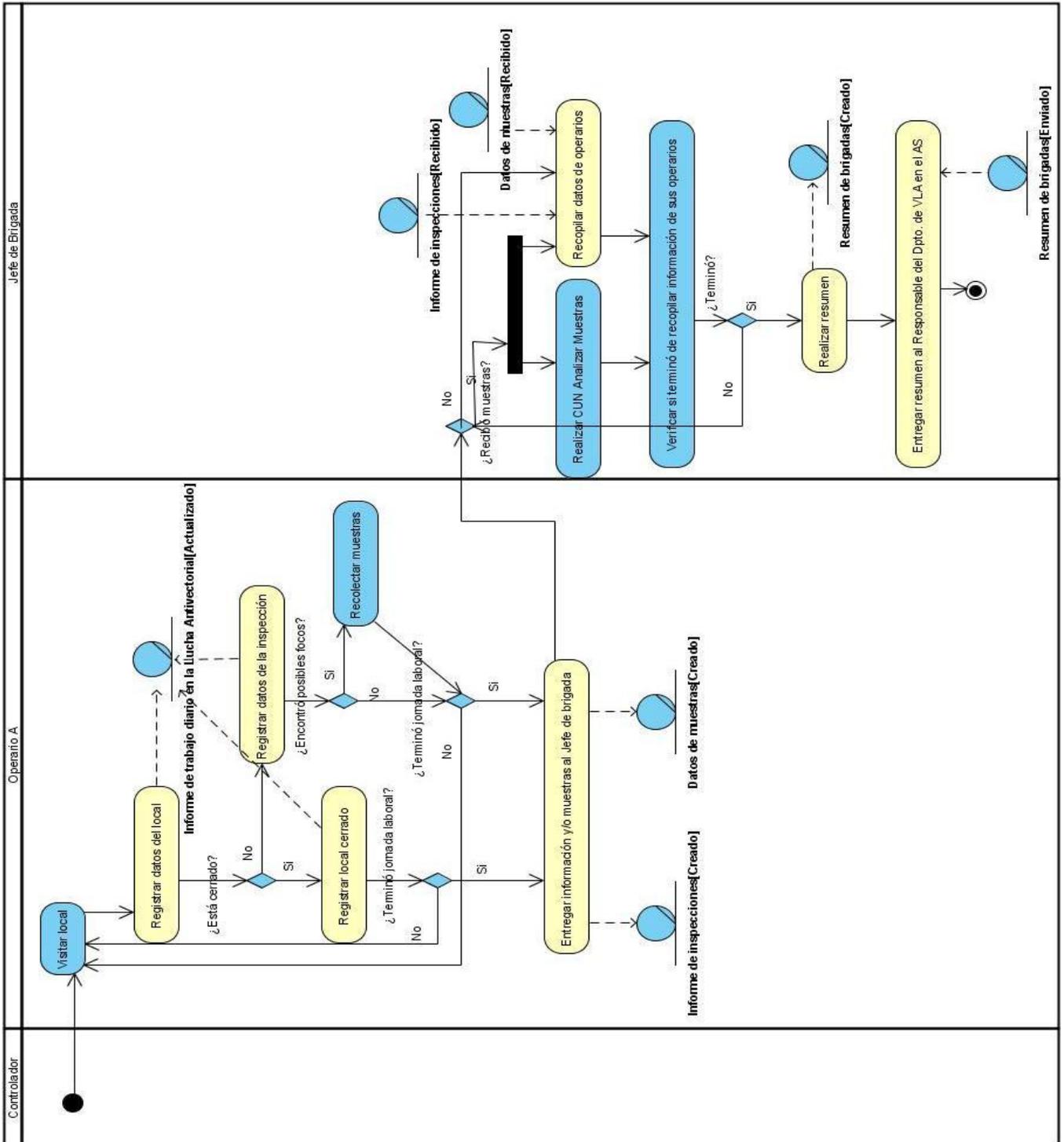


Figura A1.1 Diagrama de actividades. Recopilar información de inspecciones a locales.

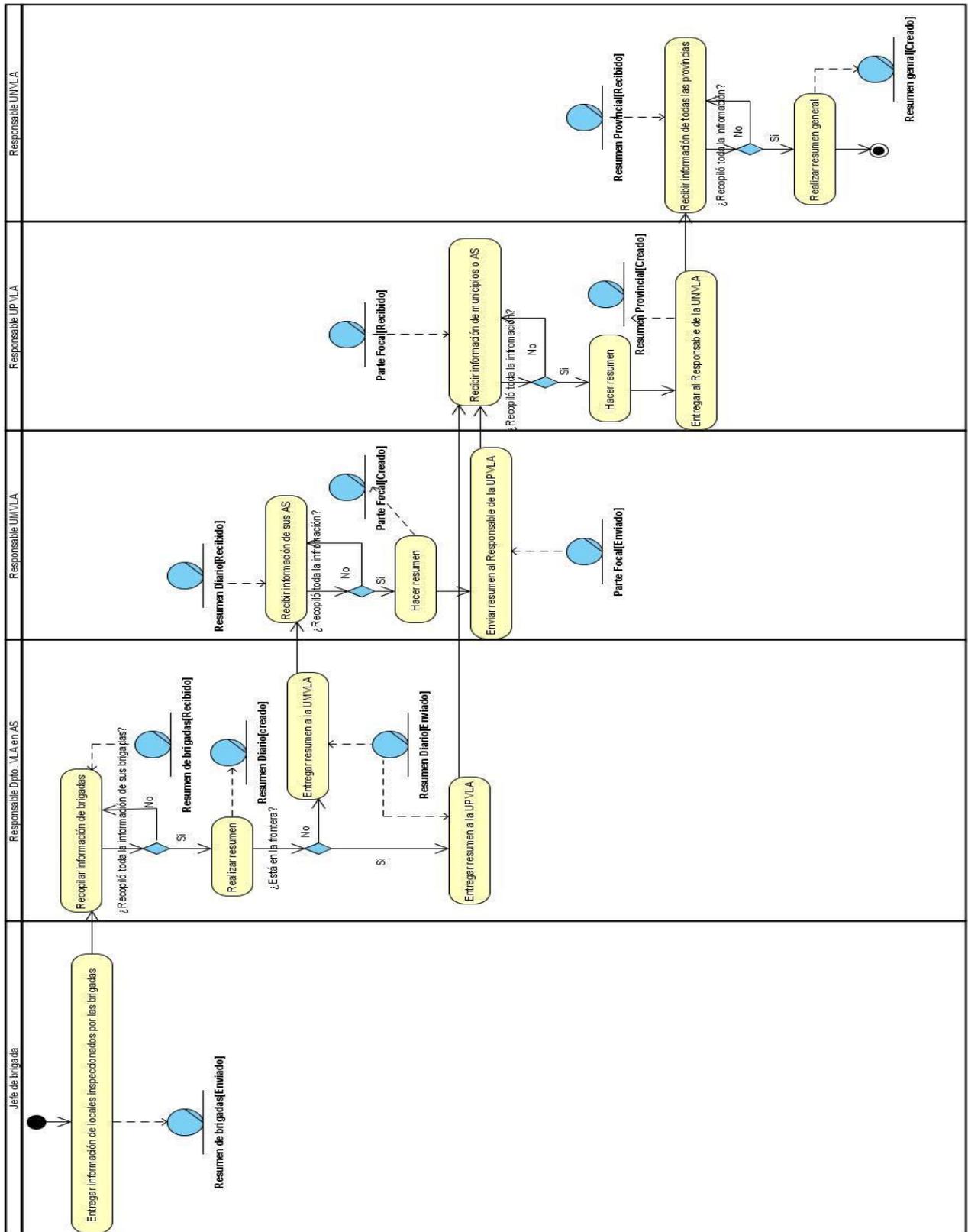


Figura A1.2 Diagrama de actividades. Recopilar información en los diferentes niveles.

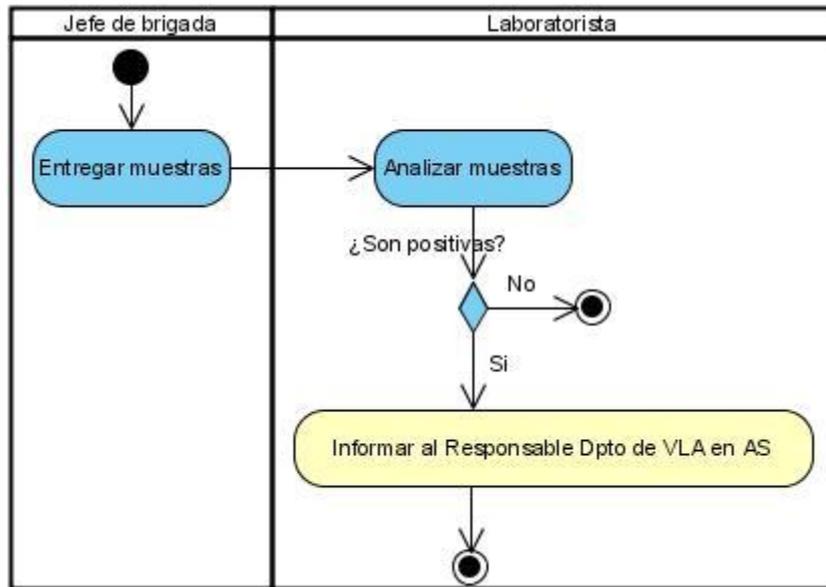


Figura A1.3 Diagrama de actividades. Enviar muestras.

*Modelo de objetos.*

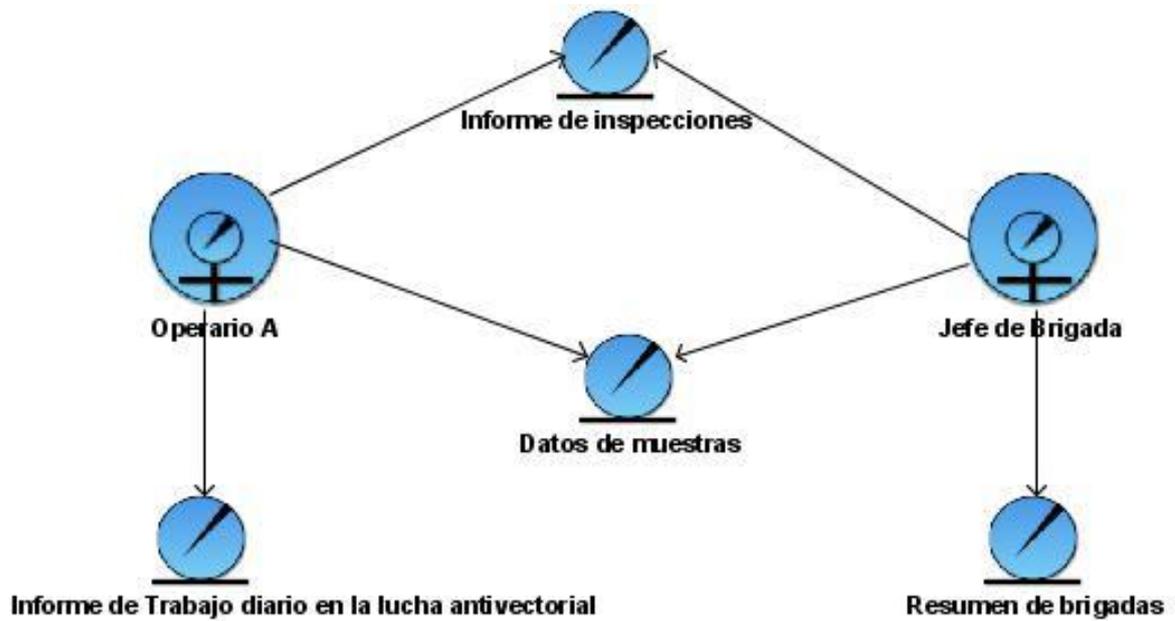


Figura A1.4 Diagrama de objeto. Recopilar información de inspecciones a locales.

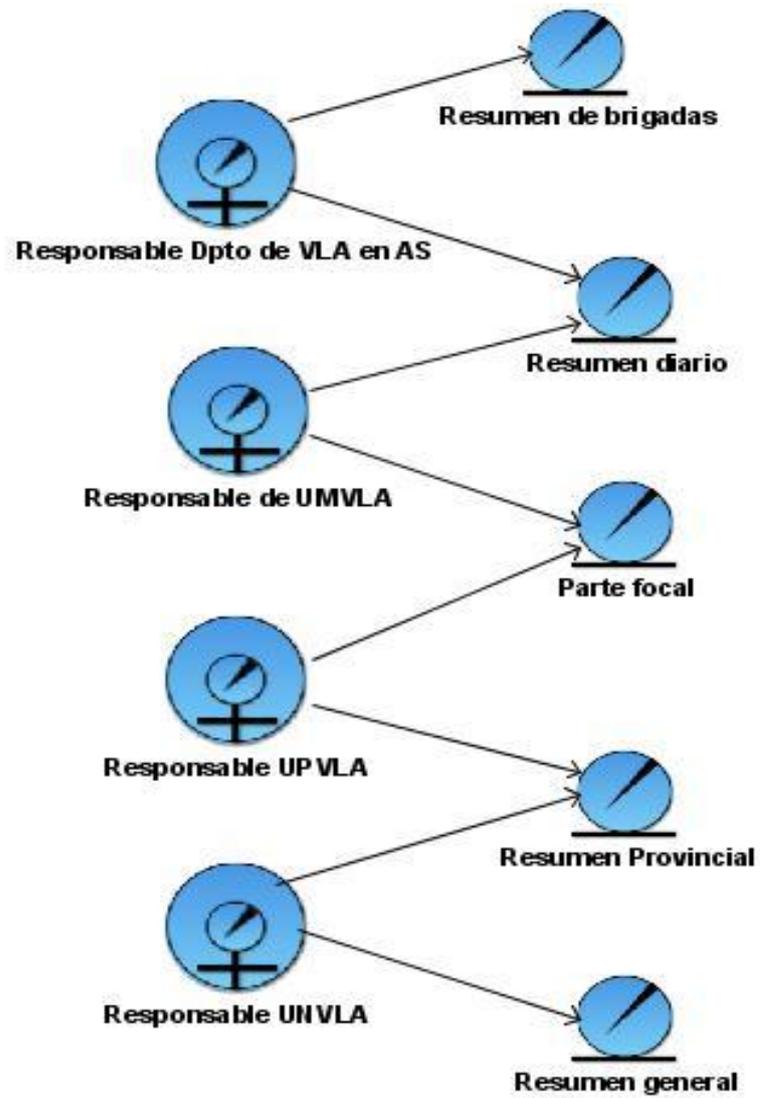


Figura A1.5 Diagrama de objeto. Recopilar información en los diferentes niveles.

Anexo 2. Modelo de sistema.

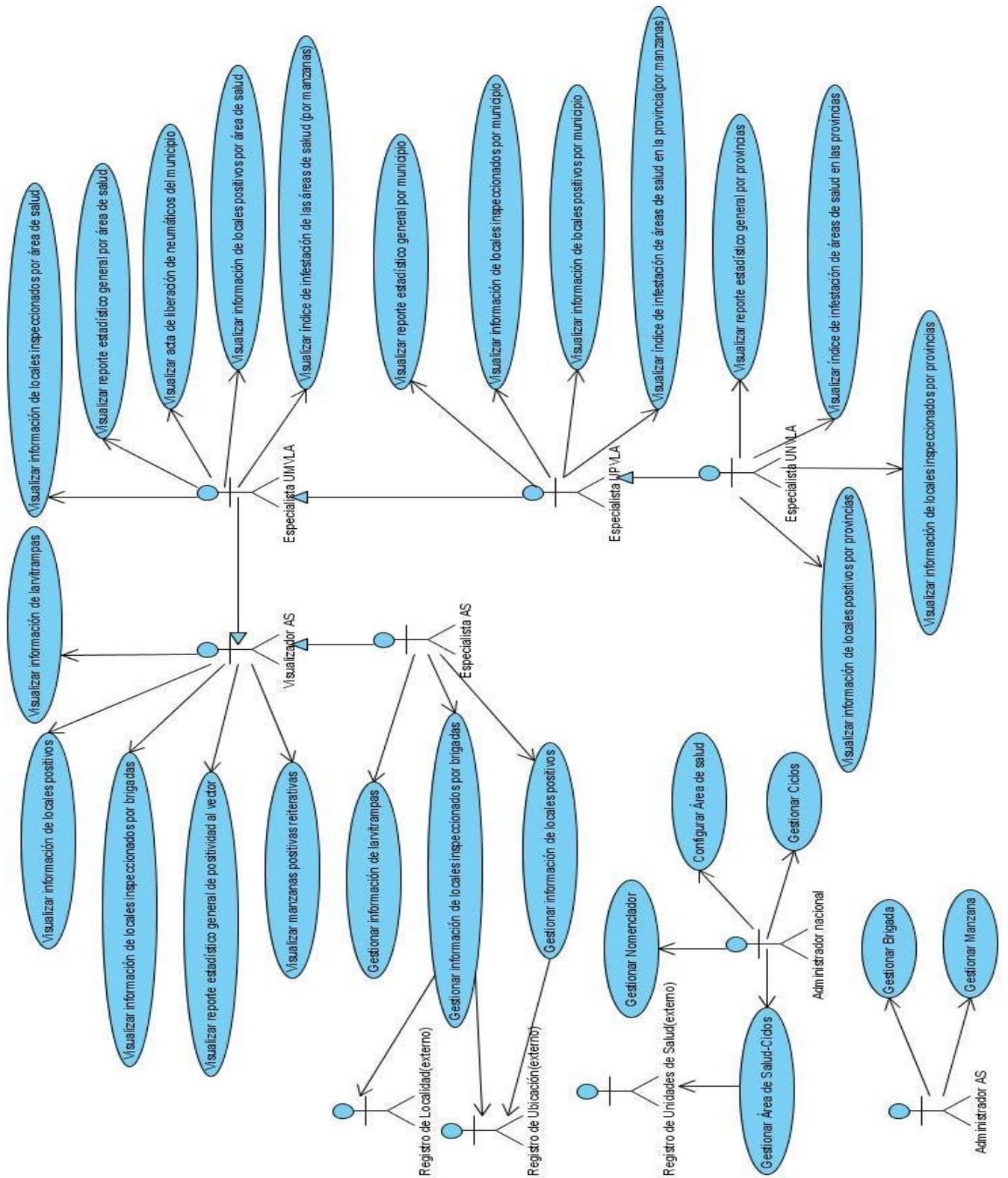


Figura A2.1 Diagrama de casos de usos del sistema.

Los casos de usos críticos fueron explicados de forma extendida en el Capítulo 2.

### A2.2 Listado de casos de uso.

<b>CU_1</b>	<b>Gestionar información de locales inspeccionados por brigadas.</b>
<b>Actor</b>	Especialista AS.
<b>Descripción</b>	Posibilita insertar, eliminar o modificar un local inspeccionado en dependencia de la brigada.
<b>Referencia</b>	R1

**Tabla A2.1 Caso de uso: Gestionar información de locales inspeccionados por brigadas.**

<b>CU_2</b>	<b>Visualizar información de locales inspeccionados por brigadas.</b>
<b>Actor</b>	Visualizador AS.
<b>Descripción</b>	Posibilita visualizar los datos de los locales inspeccionados por una brigada.
<b>Referencia</b>	R1.2

**Tabla A2.2 Caso de uso: Visualizar información de locales inspeccionados por brigadas.**

<b>CU_3</b>	<b>Gestionar información de locales positivos.</b>
<b>Actor</b>	Especialista AS.
<b>Descripción</b>	Admite la posibilidad de insertar, eliminar o modificar un local positivo.
<b>Referencia</b>	R2

**Tabla A2.3 Caso de uso: Gestionar información de locales positivos.**

<b>CU_4</b>	<b>Visualizar información de locales positivos.</b>
<b>Actor</b>	Visualizador AS.
<b>Descripción</b>	Visualiza los datos de los locales positivos.
<b>Referencia</b>	R2.2

**Tabla A2.4 Caso de uso: Visualizar información de locales positivos.**

<b>CU_5</b>	<b>Gestionar información de larvitrapas.</b>
<b>Actor</b>	Especialista AS.
<b>Descripción</b>	Posibilita insertar, eliminar o modificar datos determinados de una larvitrapa.
<b>Referencia</b>	R3

**Tabla A2.5 Caso de uso: Gestionar información de larvitrapas.**

<b>CU_6</b>	<b>Visualizar información de larvitrapas.</b>
<b>Actor</b>	Visualizador AS.
<b>Descripción</b>	Permite visualizar los datos referentes a cada larvitrapa.
<b>Referencia</b>	R3.3

**Tabla A2.6 Caso de uso: Visualizar información de larvitrapas.**

<b>CU_7</b>	<b>Visualizar Reporte estadístico general de positividad al vector.</b>
<b>Actor</b>	Visualizador AS.
<b>Descripción</b>	Permite visualizar los datos de los reportes estadísticos de positividad al vector.
<b>Referencia</b>	R4

**Tabla A2.7 Caso de uso: Visualizar Reporte estadístico general de positividad al vector.**

<b>CU_8</b>	<b>Visualizar Manzanas positivas reiterativas.</b>
<b>Actor</b>	Visualizador AS.
<b>Descripción</b>	Posibilita visualizar los datos de las manzanas positivas reiterativas.
<b>Referencia</b>	R5

**Tabla A2.8 Caso de uso: Visualizar Manzanas positivas reiterativas.**

<b>CU_9</b>	<b>Visualizar información de reporte estadístico general por área de salud.</b>
<b>Actor</b>	Especialista UMVLA.
<b>Descripción</b>	Posibilita visualizar los datos referentes a los reportes estadísticos realizados por área de salud.
<b>Referencia</b>	R6.

**Tabla A2.9 Caso de uso: Visualizar información de reporte estadístico general por área de salud.**

<b>CU_10</b>	<b>Visualizar información de locales inspeccionados por área de salud.</b>
<b>Actor</b>	Especialista UMVLA.
<b>Descripción</b>	Permite visualizar los datos referentes a los locales inspeccionados por área de salud.
<b>Referencia</b>	R7.

**Tabla A2.10 Caso de uso: Visualizar información de locales inspeccionados por área de salud.**

<b>CU_11</b>	<b>Visualizar información de locales positivos por Áreas de Salud.</b>
<b>Actor</b>	Especialista UMVLA.
<b>Descripción</b>	Posibilita visualizar los datos de los locales positivos según un área de salud determinada.
<b>Referencia</b>	R8.

**Tabla A2.11 Caso de uso: Visualizar información de locales positivos por Áreas de Salud.**

<b>CU_12</b>	<b>Visualizar índice de infestación de las áreas de salud (por manzanas).</b>
<b>Actor</b>	Especialista UMVLA.
<b>Descripción</b>	Permite visualizar los datos de los índices de infestación de las áreas de salud.
<b>Referencia</b>	R9.

**Tabla A2.12 Caso de uso: Visualizar índice de infestación de las áreas de salud (por manzanas).**

<b>CU_13</b>	<b>Visualizar reporte estadístico general por municipios.</b>
<b>Actor</b>	Especialista UPVLA.
<b>Descripción</b>	Permite visualizar los datos de los reportes estadísticos generales por municipio.
<b>Referencia</b>	R10.

**Tabla A2.13 Caso de uso: Visualizar reporte estadístico general por municipios.**

<b>CU_14</b>	<b>Visualizar información de locales inspeccionados por municipios.</b>
<b>Actor</b>	Especialista UPVLA.
<b>Descripción</b>	Permite visualizar los datos de los locales inspeccionados por municipios.
<b>Referencia</b>	R11.

**Tabla A2.14 Caso de uso: Visualizar información de locales inspeccionados por municipios.**

<b>CU_15</b>	<b>Visualizar información de locales positivos por municipios.</b>
<b>Actor</b>	Especialista UPVLA.
<b>Descripción</b>	Permite visualizar los datos de los locales positivos por municipios.
<b>Referencia</b>	R12.

**Tabla A2.15 Caso de uso: Visualizar información de locales positivos por municipios.**

<b>CU_16</b>	<b>Visualizar índice de infestación de áreas de salud en la provincia (por manzanas).</b>
<b>Actor</b>	Especialista UPVLA.
<b>Descripción</b>	Su principal objetivo es visualizar los datos de los índices de infestación de áreas de salud en la provincia (por manzanas).
<b>Referencia</b>	R13.

**Tabla A2.16 Caso de uso: Visualizar índice de infestación de áreas de salud en la provincia (por manzanas).**

<b>CU_17</b>	<b>Visualizar reporte estadístico general por provincias.</b>
<b>Actor</b>	Especialista UNVLA.
<b>Descripción</b>	Tiene como objetivo visualizar los datos de los reportes estadísticos generales por provincia.
<b>Referencia</b>	R14.

**Tabla A2.17 Caso de uso: Visualizar reporte estadístico general por provincias.**

<b>CU_18</b>	<b>Visualizar información de locales inspeccionados por provincias.</b>
<b>Actor</b>	Especialista UNVLA.
<b>Descripción</b>	Posibilita visualizar la información de los locales inspeccionados por provincias.
<b>Referencia</b>	R15.

**Tabla A2.18 Caso de uso: Visualizar información de locales inspeccionados por provincias.**

<b>CU_19</b>	<b>Visualizar información de locales positivos por provincias.</b>
<b>Actor</b>	Especialista UNVLA.
<b>Descripción</b>	Posibilita visualizar la información de los locales positivos por provincias.
<b>Referencia</b>	R16.

**Tabla A2.19 Caso de uso: Visualizar información de locales positivos por provincias.**

<b>CU_20</b>	<b>Visualizar índice de infestación de áreas de salud en la provincia.</b>
<b>Actor</b>	Especialista UNVLA.
<b>Descripción</b>	Permite visualizar los datos de los índices de infestación de áreas de salud en la provincia.
<b>Referencia</b>	R17.

**Tabla A2.20 Caso de uso: Visualizar índice de infestación de áreas de salud en la provincia.**

<b>CU_21</b>	<b>Gestionar Nomenclador</b>
<b>Actor</b>	Administrador Nacional.
<b>Descripción</b>	Su principal objetivo es insertar, eliminar o modificar un nomenclador con los datos requeridos.
<b>Referencia</b>	R18

**Tabla A2.21 Caso de uso: Gestionar Nomenclador.**

<b>CU_22</b>	<b>Gestionar Ciclo</b>
<b>Actor</b>	Administrador Nacional.
<b>Descripción</b>	Su principal objetivo es insertar, eliminar o modificar un ciclo, en dependencia de la fecha.
<b>Referencia</b>	R19

**Tabla A2.22 Caso de uso: Gestionar Ciclo.**

<b>CU_23</b>	<b>Gestionar Área de Salud-Ciclo.</b>
<b>Actor</b>	Administrador Nacional.
<b>Descripción</b>	Permite asignar, eliminar o modificar un ciclo a un área de salud determinada.
<b>Referencia</b>	R20

**Tabla A2.23 Caso de uso: Gestionar Área de Salud-Ciclo.**

<b>CU_24</b>	<b>Gestionar Manzana.</b>
<b>Actor</b>	Administrador AS.
<b>Descripción</b>	Permite insertar, eliminar o modificar datos correspondientes a una brigada.
<b>Referencia</b>	R21

**Tabla A2.24 Caso de uso: Gestionar Manzana.**

Anexo 3. Diseño del sistema.

Diagramas de clase. Casos de usos secundarios.

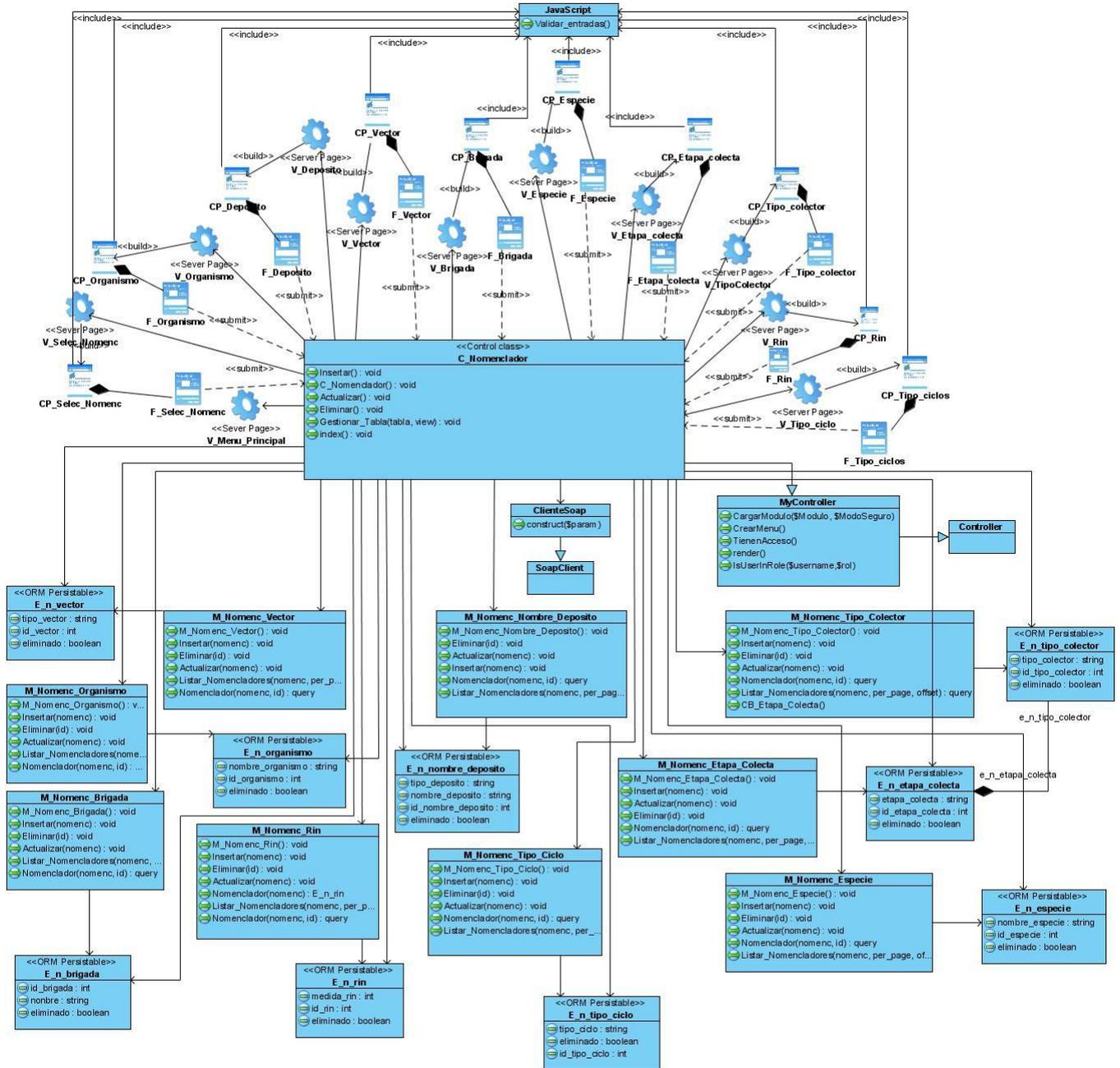


Figura A3.1. Diagrama de clases. Gestionar Nomenclador.

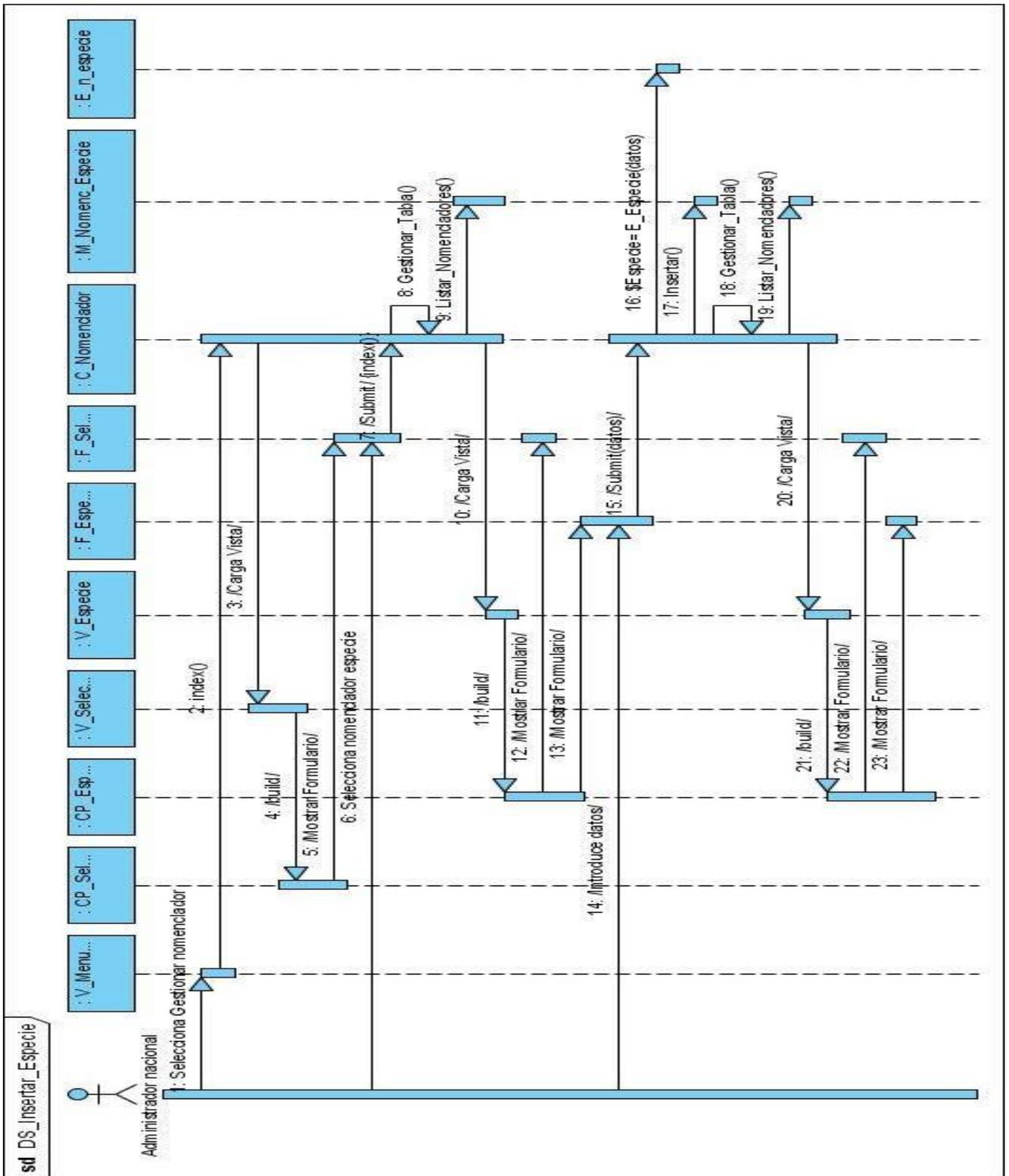


Figura A3.2 Diagrama de secuencia. Insertar especie.

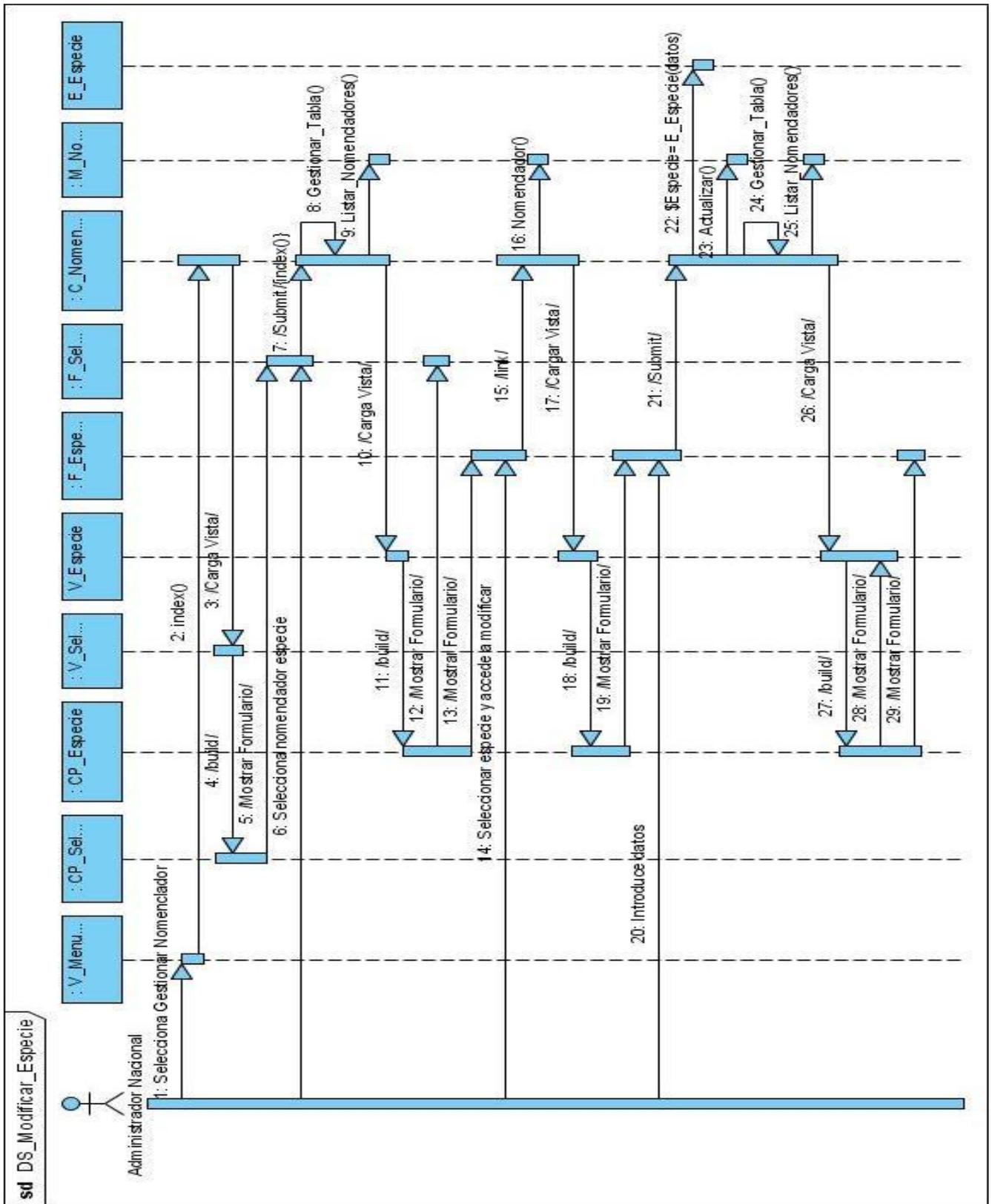


Figura A3.3 Diagrama de secuencia. Modificar especie.

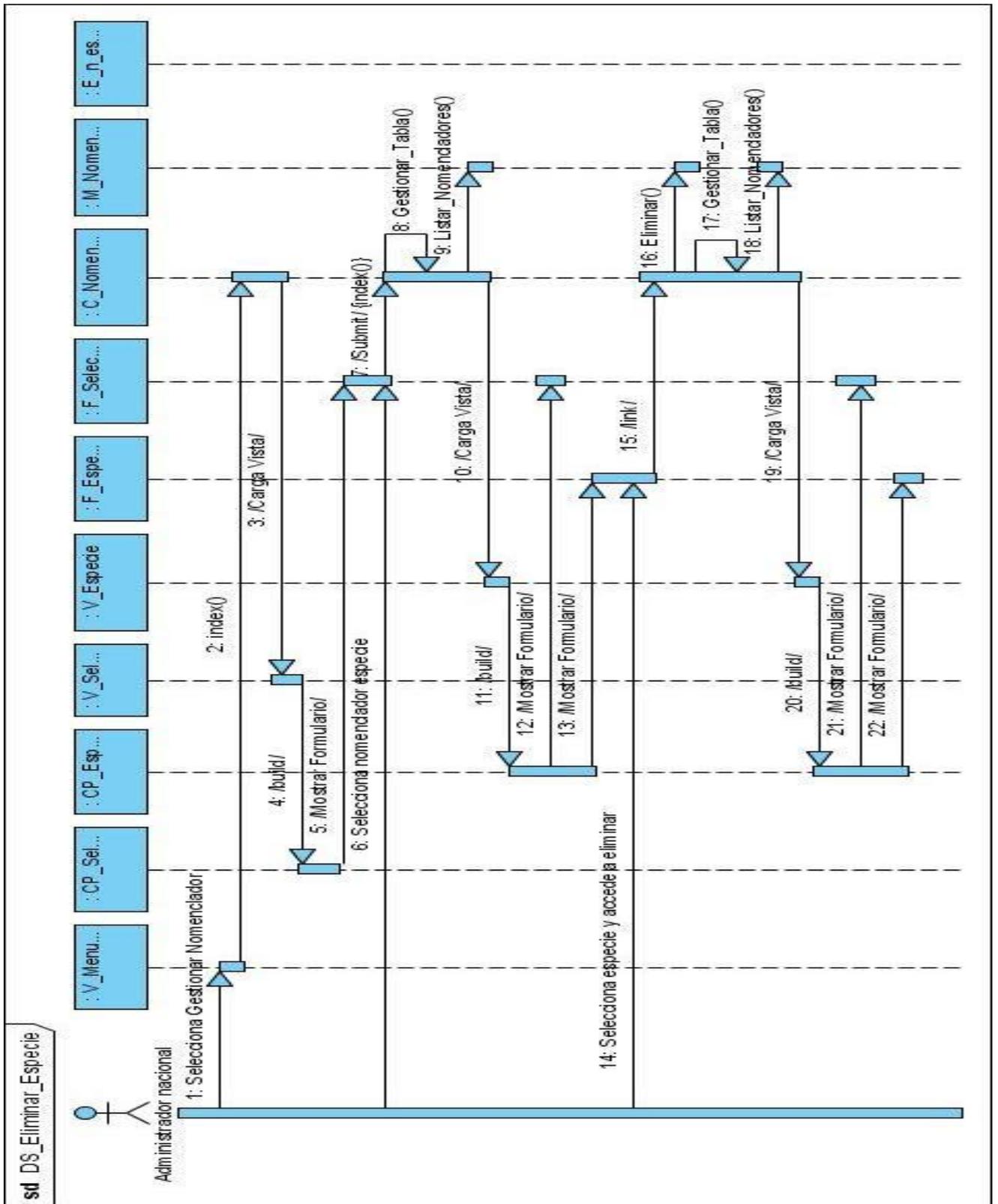


Figura A3.4 Diagrama de secuencia. Eliminar especie.

**GLOSARIO DE TÉRMINOS.**

**Aedes Aegypti:** Aedes Aegypti Linnaeus, 1762, es un mosquito cuyo origen se ubica geográficamente en la Región Etiópica (África), que nuclea la mayor cantidad de especies del Subgénero Stegomyia Theobald, 1901, al cual este culícido pertenece.

**Agente patógeno:** Es toda aquella entidad biológica capaz de producir enfermedad o daño en la biología de un hospedero (humano, animal o vegetal) sensiblemente predispuesto.

**Aplicación:** Programa informático que proporciona servicios de alto nivel al usuario, generalmente utilizando otros programas más básicos que se sitúan por debajo.

**Arbovirosis:** Término que comprende el conjunto de enfermedades producidas por los virus transmitidos por artrópodos.

**Artrópodo:** Tipo de los animales invertebrados de cuerpo con simetría bilateral, formado por una serie lineal de segmentos y provisto de apéndices articulados o artejos.

**Automatización:** Conversión de determinados procesos corporales o psíquicos en automáticos. Aplicación de procedimientos automáticos a un aparato, proceso o sistema.

**Bacteria:** Organismo microscópico sin núcleo diferenciado, que se multiplica por división simple o por esporas. Las bacterias son agentes de numerosas enfermedades infecciosas.

**Base de datos:** Conjunto de datos no redundantes, almacenados en un soporte informático, organizados de forma independiente de su utilización y accesibles simultáneamente por distintos usuarios y aplicaciones. La diferencia de una BD respecto a otro sistema de almacenamiento de datos es que se almacenan en la BD de forma que cumplen tres requisitos básicos: no redundancia, independencia y concurrencia.

**C:** Lenguaje de programación estructurado, de aplicación general.

**C++:** Lenguaje C orientado a objetos.

**Ciclo de vida:** Conjunto organizado de actividades y fases, tanto técnicas como de gestión que son necesarias desarrollar a lo largo de la vida del sistema, desde que se decide su necesidad hasta que

el sistema deja de ser utilizado. Las fases son: planificación, análisis, diseño, construcción e implantación.

**Código:** Cada una de las secuencias de caracteres que transforman los elementos de un repertorio en otro.

**Culícido:** Se dice de los insectos dípteros del suborden de los nematóceros, provistos de una probóscide que contiene cuatro o más cerdas fuertes, las cuales utilizan las hembras para perforar la piel del hombre y los animales y chupar la sangre de que se alimentan. Los machos viven de jugos vegetales. Se desarrollan en el agua, en cuya superficie depositan sus huevos las hembras.

**Dengue:** Enfermedad epidémica caracterizada por fiebre, dolores en los miembros y un exantema seguido de descamación.

**Digitalización:** Expresión de una información en dígitos para su tratamiento informático.

**Entomología:** Palabra que proviene de los términos griegos "entomos" que significa insectos y "logos" que significa ciencia, la Entomología sería la ciencia que estudia los insectos.

**Epidemia:** Enfermedad infecciosa que durante un período de tiempo ataca, simultáneamente y en un mismo territorio, a gran número de personas.

**Epidemiología:** Es la disciplina científica que estudia la distribución, frecuencia, determinantes, relaciones, predicciones y control de los factores relacionados con la salud y enfermedad. La epidemiología en sentido estricto, ocupa un lugar especial en la intersección entre las ciencias biomédicas y las ciencias sociales y aplica los métodos y principios de estas ciencias al estudio de la salud y la enfermedad en poblaciones humanas determinadas. Existe una epidemiología veterinaria y podría hablarse de una epidemiología zoológica y botánica, íntimamente relacionada con la ecología.

**Flameo:** Acción y efecto de flamear. Quemar con alcohol u otro líquido inflamable en superficies.

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

**Hipertexto:** Asociación de información (texto, gráficos, sonido) organizada según una estructura de referencias que permite al usuario saltar de un concepto a otro relacionado con el primero, utilizando dispositivos interactivos y una interfaz gráfica visual.

**IBM (International Business Machines):** Es una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática. Tiene su sede en Armonk (Estados Unidos) y está constituida como tal desde el 15 de junio de 1911.

**INFOMED:** Nombre que identifica a la primera red electrónica cubana de información para la salud y surgió como parte de un proyecto del Centro Nacional de Información de Ciencias Médicas de Cuba. Es el Portal de Salud Cubano y la red de personas e instituciones que comparten el propósito de facilitar el acceso a la información de salud en Cuba.

**Informatización:** Aplicación de sistemas y equipos informáticos al tratamiento de información.

**Insecto:** Referido a un animal artrópodo, que tiene respiración por tráqueas, el cuerpo dividido en cabeza, tórax y abdomen, dos antenas, seis patas y dos o cuatro alas.

**Internet:** Red informática de comunicación internacional que permite el intercambio de todo tipo de información entre sus usuarios. El nombre proviene del acrónimo de las palabras inglesas International Network (red internacional).

**Malacología:** Parte de la zoología que estudia los moluscos.

**Malaria:** Del italiano "malaria", mala aria (mal aire), también denominada fiebre palúdica o paludismo, es una enfermedad producida por parásitos del género Plasmodium.

**Nematodo:** Referido a un gusano, que tiene el cuerpo sin segmentaciones y con forma cilíndrica y alargada, y que está provisto de un aparato digestivo formado por un tubo recto que va desde la boca al ano, y vive generalmente como parásito de otros animales.

**Parásito:** Referido a un organismo animal o vegetal, que vive a costa de otro de distinta especie, alimentándose de él y causándole algún perjuicio.

**Patógeno:** Que origina y desarrolla una enfermedad.

**Protista:** Referido a un organismo, que se caracteriza por su pequeño tamaño y por carecer de órganos y de tejidos diferenciados.

**Reservorio:** Población de seres vivos u organismo que alojan de forma natural el germen de una enfermedad que puede propagarse como epidemia: Algunos animales son reservorios del paludismo, que pasa indirectamente a las personas cuando un mosquito pica al animal y después al individuo.

**Software:** Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

**Tecnología de la información y las comunicaciones (TIC):** Conjunto de avances tecnológicos que proporcionan la informática, las telecomunicaciones y las tecnologías audiovisuales, que comprenden los desarrollos relacionados con los ordenadores, Internet, la telefonía, las aplicaciones multimedia y la realidad virtual. Estas tecnologías básicamente proporcionan información, herramientas para su proceso y canales de comunicación.

**Vertebrado:** Referido a un animal, que tiene un esqueleto con columna vertebral y cráneo, y un sistema nervioso central constituido por la médula espinal y el encéfalo.

**Virus:** Microorganismo de estructura sencilla, capaz de reproducirse en el seno de células vivas específicas, que está compuesto fundamentalmente por ácido ribonucleico o ácido desoxirribonucleico y por proteínas, y que es el causante de muchas enfermedades.

