

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**FACULTAD 6**



**SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA  
REPRESENTACION Y ANÁLISIS DE ÁREAS URBANAS  
DE RIESGO POR CONTAMINACIÓN**

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS

**Autor:** María Lourdes Sánchez Zequeira

**Tutor:** Msc. Yuniel Eliades Proenza Arias

La Habana, Julio 2015  
“Año 56 de la Revolución”



*"Haz de los obstáculos  
escalones para aquello que  
quieres alcanzar".*

*Charlie  
Chaplin*



## **DECLARACIÓN DE AUTORÍA**

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

**Autor:** María Lourdes Sánchez Zequeira

\_\_\_\_\_

**Tutor:** Msc. Yuniel Eliades Proenza Arias

### **DATOS DEL TUTOR**

**Nombre y apellidos:** Msc. Yuniel Eliades Proenza Arias.

**Correo electrónico:** [yproenza@uci.cu](mailto:yproenza@uci.cu)

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Títulos:** Ingeniero Informático, Máster en Ingeniería de Software e Inteligencia Artificial.

**Breve descripción:** Graduado de Ingeniería Informática en el año 2006 de la Universidad de Holguín y CUJAE. Máster en Ingeniería de Software e Inteligencia Artificial por la Universidad de Málaga en el 2011. Se ha desempeñado como analista y desarrollador de aplicaciones Web y Desktop. Tiene experiencia en el trabajo con Sistemas de Información Geográfica.

### **DATOS DEL AUTOR**

**Nombre y apellidos:** María Lourdes Sánchez Zequeira

**Correo electrónico:** [mlsanchez@estudiantes.uci.cu](mailto:mlsanchez@estudiantes.uci.cu)

**Breve descripción:** Estudiante que cursa el quinto año de la carrera Ingeniería en Ciencias Informáticas en la Universidad de Ciencias Informáticas.

## *Agradecimientos*

*A la UCI por darme la oportunidad de formarme como profesional.*

*A mis padres, mi hermano, y a mi familia en general, por darme todo el apoyo y las ganas de seguir adelante y lograr mis metas; por estar siempre pendiente de mí, y de mi carrera universitaria.*

*Porque a ellos les debo en gran medida quien soy y porque para ellos son mis triunfos.*

*A Arturito, mi niño, por ser la fuerza que me impulsa a seguir mis sueños y por estar día a día junto a mi desde el primer latido de tu corazón, te quiero mi niño, aunque aún seas pequeño algún día sabrás que todo este sacrificio es por ti.*

*A Idel (mi novio), por ser mi compañero, amigo, mi confidente, por todo el apoyo y comprensión que me ha dedicado. Gracias por tener siempre tus brazos extendidos para mí, por estar presente en los momentos felices y difíciles a mi lado, por todo el amor que me has brindado. Te quiero.*

*A mi tutor Yuniel, por guiarme en todo momento, por estar pendiente de cada paso dado, por su apoyo y recomendaciones y por su esfuerzo para lograr que los resultados fueran los mejores, por todo su apoyo, confianza y ayuda.*

*Al tribunal por la seriedad puesta durante la investigación y por la ayuda brindada.*

*A todas las amistades ganadas en estos años, por haber sido como una familia muy grande, porque a pesar de la distancia los llevaré siempre en el corazón.*

*Gracias a todos, los quiero.*

*A los que me preguntaron alguna vez “¿Y la tesis cómo va?”*

*Dedicatoria*

*A Dios por darme la oportunidad de realizar hoy parte de mi sueño y de vivir esta vida maravillosa.*

*A mis queridos padres por haberme apoyado en todo momento, por confiar en mí y darme confianza cuando más lo necesitaba, por tanto cariño y afecto que me han brindado. Por todo el sacrificio hecho hasta el día de hoy. Por darme todo y ser todo para mí.*

*A mi mamá por devolverme la serenidad y la fe con sus consejos cuando asoma la desesperanza, por ser quien me impulsa día a día, te quiero.*

*A mi papá por forjar mi carácter y ser guía en mi camino, te quiero.*

*A mi hermano, por ser mi otra mitad, mi amigo y consejero, por ser el mejor regalo que me han dado mis padres.*

*A Arturito por ser el regalo más grande que me pudo haber dado la vida, sin tu amor y tus tiernas travesuras no hubiese podido lograr esto. Siempre serás la luz que guía mi camino. Gracias por ser mi más bello compromiso.*

*A mi familia que adoro y está a mi lado en la malas cuando todos se van.*

## RESUMEN

Hoy en día el problema de la contaminación ambiental figura entre las principales preocupaciones en la sociedad, debido a los efectos que esta produce en la salud. Sin dudas, los desechos son una de las principales causas de contaminación ambiental. La vinculación de la salud humana y el sistema ambiental se presenta, en la actualidad, como una prioridad mundial. En Cuba, organismos como el Ministerio de Salud Pública, necesitan obtener información geográfica, referente a la ubicación de los entes de contaminación del país.

Específicamente, en las instituciones de sanidad, existen grandes dificultades en el proceso de análisis de la información y ubicación de los entes de contaminación, ya que la mala disposición de residuos, incide en el riesgo epidemiológico que representa la acumulación y vertimiento incontrolado de basura. Es necesario, mantener el control de las enfermedades transmisibles por vectores que habitan en los basureros, con el objetivo de evitar brotes y epidemias.

Por lo que el presente trabajo se dirigió a la creación de un Sistema de Información Geográfica (SIG) para que pueda ser utilizado por los especialistas de esta entidad. Este sistema cuenta con servicios de ubicación de los entes contaminantes, además se pueden realizar búsquedas por provincia y/o municipios, nombre o seleccionar la región deseada, además de brindar una visualización más detallada, aumentando o disminuyendo la imagen del mapa que desea visualizar el usuario. Se realizaron pruebas utilizando el método de caja negra, con el objetivo de validar las funcionalidades de la aplicación.

**Palabras claves:** contaminación ambiental, salud, desechos, entes de contaminación.

## **ABSTRACT**

Today the problem of environmental pollution is one of the main concerns in society, due to the effects it produces on health. Undoubtedly, waste is a major cause of environmental pollution. Linking human health and environmental system is presented today as a global priority. In Cuba, agencies such as the Ministry of Health need to obtain geographical information concerning the location of the bodies of contamination of the country.

Specifically, health institutions, there are great difficulties in the process of analyzing the information and location of the bodies of contamination as the unwillingness of waste affects the epidemiological risk posed by the accumulation and uncontrolled dumping of waste. Necessary to maintain control of vector-borne diseases that live in garbage dumps, in order to prevent outbreaks and epidemics.

So this study addressed the creation of a GIS (Geographical Information System) so it can be used by specialists of this entity. This system has the location services polluters, either by searching by county and / or municipalities, name or category thereof, in addition to providing a more detailed view, increasing or decreasing the map image to be displayed user . They also performed, you try using the black box method, in order to validate the functionality of the application.

**Key words:** environmental pollution, health, waste, pollution entities.

**ÍNDICE**

RESUMEN .....	V
ABSTRACT .....	VI
ÍNDICE DE FIGURAS .....	X
ÍNDICE DE TABLAS .....	XI
INTRODUCCION.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	5
1.1    INTRODUCCIÓN .....	5
1.2    CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA .....	5
1.3    OBJETO DE ESTUDIO: SISTEMAS DE INFORMACIÓN GEOGRÁFICA COMO HERRAMIENTA DE REPRESENTACIÓN Y ANÁLISIS. ....	6
1.4    SITUACIÓN PROBLEMÁTICA.....	9
1.5    ANÁLISIS DE SOLUCIONES EXISTENTES.....	11
1.5.1.  SIG A NIVEL MUNDIAL: SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA EL ANÁLISIS EPIDEMIOLÓGICO (SIGEPI).....	11
1.5.2.  SIG A NIVEL MUNDIAL: SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA EL MAPEO DE ENFERMEDADES (EPIINFO/EPIMAP).....	12
1.5.3.  SIG EN CUBA: SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA GESTIÓN DE LA ESTADÍSTICA DE SALUD DE CUBA (SIG-ESAC). ....	14
1.5.4.  OBSERVACIÓN DE LAS SOLUCIONES EXISTENTES.....	15
1.6    PLATAFORMA, HERRAMIENTAS Y FRAMEWORK DE DESARROLLO .....	16
1.6.1.  GENESIG 1.5 .....	16
1.6.2.  UML 2.0 COMO LENGUAJE UNIFICADO DE MODELADO .....	19
1.6.3.  SISTEMAS GESTORES DE BASES DE DATOS .....	20
1.6.4.  NETBEANS COMO IDE PARA PHP .....	20
1.6.5.  VISUAL PARADIGM COMO HERRAMIENTA CASE.....	21
1.7    LENGUAJES DE PROGRAMACIÓN.....	21
1.7.1.  LENGUAJES WEB DE PROGRAMACIÓN DEL LADO DEL CLIENTE.....	22
1.7.2.  LENGUAJES WEB DE PROGRAMACIÓN DEL LADO DEL SERVIDOR.....	22
1.7.3.  TÉCNICAS Y FORMATOS DE INTERCAMBIO DE DATOS.....	23

---

1.8	METODOLOGÍAS DE DESARROLLO .....	24
1.9	CONCLUSIONES PARCIALES.....	26
<b>CAPÍTULO 2. PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA .....</b>		<b>27</b>
2.1	INTRODUCCIÓN .....	27
2.2	MODELO DEL DOMINIO EN EL DESARROLLO DEL SOFTWARE.....	27
2.2.1.	DESCRIPCIÓN DEL MODELO DEL DOMINIO .....	28
2.2.2.	GLOSARIO DE TÉRMINOS DEL MODELO DEL DOMINIO .....	28
2.3	MODELO DEL SISTEMA.....	29
2.3.1	REQUISITOS.....	29
2.3.2	ESTRATEGIA DE CAPTURA DE REQUISITOS .....	29
2.3.3	LISTADO DE REQUISITOS FUNCIONALES .....	30
2.3.4	REQUISITOS No FUNCIONALES.....	35
2.4	DEFINICIÓN DE CASOS DE USO DEL SISTEMA.....	37
2.4.1	DESCRIPCIÓN DE LOS ACTORES .....	37
2.4.2	DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	38
2.4.3	DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO DEL SISTEMA. ....	38
2.4.4	PATRONES DE CASOS DE USO .....	46
2.5	ARQUITECTURA DE SOFTWARE .....	47
2.6	PATRONES.....	48
2.6.1	PATRONES DE ARQUITECTURA.....	48
2.6.2	PATRONES DE DISEÑO .....	49
2.7	MODELO DE DISEÑO .....	50
2.7.1	DIAGRAMA DE CLASES DEL DISEÑO.....	50
2.7.2	DISEÑO DE LA BASE DE DATOS .....	51
2.8	CONCLUSIONES PARCIALES .....	53
<b>CAPÍTULO 3. CONSTRUCCION Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA</b>		<b>54</b>
3.1	INTRODUCCIÓN .....	54
3.2	MODELO DE IMPLEMENTACIÓN .....	54
3.2.1	DIAGRAMA DE COMPONENTES .....	54
3.3	MODELO DE DESPLIEGUE .....	55
3.4	DESCRIPCIÓN DE LAS PRUEBAS. ....	56

3.4.1 PRUEBAS DE SOFTWARE APLICADAS .....	58
3.4.2 DISEÑO DE LAS PRUEBAS PARA VALIDAR LA SOLUCIÓN PROPUESTA .....	58
3.4.3 RESULTADOS DE LAS PRUEBAS REALIZADAS.....	62
3.5 CONCLUSIONES PARCIALES.....	63
CONCLUSIONES GENERALES .....	64
RECOMENDACIONES.....	65
REFERENCIAS BIBLIOGRÁFICAS.....	66
ANEXOS .....	68

## ÍNDICE DE FIGURAS

Fig. 1 Interfaz gráfica de SigEpi (Epidemiológico, 2001). .....	12
Fig. 2 Interfaz gráfica de EpiInfo (CDC. EpiInfo, 2014).....	14
Fig. 3 Pantalla de inicio del SIG-ESAC con uno de los menús personalizados (Núñez, 2006). .....	15
Fig. 4 Interfaz de la plataforma GeneSIG (Espinosa, y otros, 2012).....	17
Fig. 5. Componentes de MapServer (MapServer, 2015). .....	19
Fig. 6 . Ciclo de vida del proceso unificado ágil (AUP, 2015). .....	25
Fig. 7 Modelo del Dominio. ....	28
Fig. 8 Diagrama de Casos de Uso del Sistema. ....	38
Fig. 9 Diagrama de Clases del Diseño del CU Gestionar Ente Contaminante. ....	51
Fig. 10. Diagrama de Clases Persistentes .....	52
Fig. 11. Diagrama Entidad - Relación.....	52
Fig. 12 Diagrama de componentes CU Gestionar Ente Contaminante.....	55
Fig. 13 Diagrama de Despliegue.....	56

## ÍNDICE DE TABLAS

Tabla 1. Descripción de los actores del sistema. ....	37
Tabla 2. Descripción del Caso de Uso “Gestionar Ente Contaminante”. ....	39
Tabla 3. Diseño Caso Prueba CU: “Gestionar Ente Contaminante”. ....	59
Tabla 4. Variables del Caso de Prueba: “Gestionar Ente Contaminante”. ....	62
Tabla 5. No Conformidades detectadas al aplicar los casos de prueba al sistema.....	69

## **INTRODUCCION**

La preservación del medio ambiente es uno de los desafíos más importantes de la actualidad. En los años recientes se ha dado una gran importancia a los efectos de la contaminación ambiental en los ecosistemas y en la salud del hombre. De hecho, el tema de la contaminación ha pasado a convertirse en un problema vital para la humanidad, debido a la gravedad en que incurren estos al propagarse y causar daños a la salud. Sin dudas, los desechos son una de las principales causas de contaminación ambiental. Los riesgos al medio ambiente y a la salud causados por los residuos sólidos, son un foco de atención a nivel mundial, que ha propiciado que se generen disposiciones regulatorias (leyes, reglamentos y normas); que establecen pautas de conducta a evitar y medidas a seguir para lograr dicho manejo seguro a fin de prevenir riesgos (Teja Pérez, 1987).

La mala disposición de los desechos sólidos, trae como consecuencia que las poblaciones eliminen incontroladamente los residuales en cualquier lugar, y su falta de conocimiento sobre los posibles daños que esto produce, es debido, fundamentalmente al aumento de los posibles criaderos de vectores que transmiten la malaria y el dengue (presencia de cascarones de cocos, latas, botellas y otros, al aire libre). Otra enfermedad, es el parasitismo intestinal, consecuencia del fecalismo al aire libre al no existir sistema de alcantarillado, fosas suficientes y letrinas con condiciones adecuadas; esto favorece la proliferación de vectores y roedores, los cuales transmiten en sus patas, gérmenes y parásitos que son ingeridos por las personas en los alimentos y el agua de tomar, también otro factor es la presencia de micro-vertederos, lo que provoca que la acumulación de basuras a veces por más de 15 días, estimule el aumento de dichos vectores (insectos, moscas, etc. ) y otros animales que transmiten la enfermedad.

En las calles existen los botaderos abiertos que facilitan el acceso a los desechos por parte de animales domésticos y, por consiguiente, la potencial diseminación de enfermedades y contaminantes químicos a través de la cadena alimenticia, puede conllevar a la aparición de enfermedades parasitarias y diarreicas. El polvo llevado desde un botadero abierto por el viento, puede portar agentes patógenos y materiales peligrosos que junto con el humo generado de la quema de basura en estos botaderos, constituye un importante irritante respiratorio y puede hacer que las poblaciones afectadas tengan mucho más susceptibilidad a las enfermedades respiratorias.

Recientemente, la alta dirección del país ha comenzado a combatir abiertamente las manifestaciones de indisciplinas sociales asociadas al vertimiento de basuras y desechos en zonas no autorizadas. Los vehículos que se utilizan no son los más eficaces para la recogida de residuos, siendo los mismos de caja abierta y sin las condiciones técnicas ya que no cuentan con accesorios para la carga y el viraje de los

contenedores, ni con sistema de compactación. No se puede garantizar una forma de recolección, almacenamiento y eliminación adecuada con regularidad.

Se puede agregar también, que los daños al medio ambiente están provocados por la disposición no apropiada de residuos producidas por el vertimiento de residuos sólidos a ríos y arroyos, lo cual están provocando la contaminación de los cursos superficiales y subterráneos de agua, además, de contaminar la población que habita en estos medios, otras especies y los ecosistemas. El sistema de eliminación es deficiente. En el lugar donde se vierten los residuos son quemados sin tener en cuenta que este procedimiento es inadecuado pues daña grandemente el suelo, el aire, la salud y sobre todo la capa de ozono.

El efecto negativo es fácilmente reconocible, el deterioro estético de la ciudad, además, la contaminación produce el envenenamiento e infertilidad de los suelos, cuando se descargan sobre ellos sustancias tóxicas, alterando la composición y capacidades naturales. La acumulación en lugares no aptos, trae consigo un impacto paisajístico negativo, además de tener en algún caso asociado un importante riesgo ambiental; pudiéndose producir accidentes o derrumbes, también constituyen una molestia pública; invaden los caminos, restan estética al panorama, y emiten olores desagradables y polvos irritantes, los que podrían provocar graves daños al ambiente natural.

Es importante tener conocimiento de la ubicación exacta de estos basureros o entes de contaminación para, ante epidemias, poder conocer el posible origen o contrarrestar cualquier incremento asociado a ellos. Por ejemplo, en las batallas de lucha contra el dengue y el cólera se ha tomado como medida principal eliminarlos, pero sin conocer su ubicación, es muy difícil asignar los medios necesarios para ello. Para las Instituciones de Salud Pública es fundamental saber la ubicación exacta de estos, las vulnerabilidades o riesgos de enfermedad que pueden acarrear, y otros datos que pudieran ayudar a prevenir o combatir enfermedades.

Parte de esta información que se maneja se realiza de forma manual, ocasionando deficiencias como la pérdida de información, para responder ante las quejas de la población en tiempo, demora en la toma de decisiones importantes para el sector, por el lento procesamiento de la misma. También es necesario conocer estos datos con antelación, pues se pueden tomar medidas preventivas para que, ante desechos orgánicos, materiales o de cualquier otro tipo, el personal de sanidad elimine estos desechos que tanto molestan a la población.

De lo expuesto anteriormente, se identificó el siguiente **problema de la investigación**:

¿Cómo facilitar la representación y análisis de la información asociada a los entes de contaminación urbanos para prevenir o combatir enfermedades o epidemias en la población cubana?

Como resultado del propio problema se definió como **objeto de estudio** el proceso de representación y análisis de información urbana sobre entes de contaminación para su erradicación y como **campo de acción** a la representación de la información característica de basureros y zanjas para su tratamiento por sanidad. Como **objetivo general** se tiene elaborar un Sistema de Información Geográfica para la representación y análisis de la información de los basureros urbanos y entes de contaminación en poblados cubanos.

Por lo anteriormente mencionado se propone las siguientes **preguntas científicas** a las que se dará respuesta durante el desarrollo de la investigación:

¿Qué factores afectan la correcta identificación y erradicación de basureros urbanos o fuentes de contaminación?

¿Cómo influyen estos entes de contaminación en la propagación de enfermedades o epidemias?

¿Cómo se podría facilitar el proceso de identificación y erradicación de basureros urbanos?

¿Existen acciones, procesos o intenciones de automatización con este objetivo?

¿Cómo influye la utilización de un sistema informático que permita identificar espacialmente estos entes, analizar su información y su impacto sobre las áreas y población cercanas, en su correcta eliminación?

Con el fin de darle cumplimiento al objetivo general y al problema anteriormente planteado se trazaron las siguientes **tareas de la investigación**:

- ✓ Caracterizar los procesos necesarios para la representación y análisis de la información de los entes de contaminación y basureros urbanos.
- ✓ Caracterizar el estado del arte a nivel nacional e internacional de soluciones existentes similares.
- ✓ Caracterizar las herramientas y tecnologías a utilizar en el desarrollo de la solución.
- ✓ Diseñar los artefactos que se involucran en el desarrollo de la solución.
- ✓ Implementar la propuesta de solución.
- ✓ Validar la propuesta de solución.

A lo largo del desarrollo de la investigación se utilizan un conjunto de **métodos científicos** como son:

Los métodos teóricos:

1. *Analítico-Sintético*: Se utiliza en el análisis de las bibliografías especializadas en el tema de la investigación y sintetizar los conceptos fundamentales que sean necesarios para la solución del problema.

2. *Histórico-Lógico*: Se utilizó para el análisis de la evolución histórica de las herramientas similares al Sistema de Información Geográfica a desarrollar, para tenerlos como base para esta investigación.

Los métodos empíricos:

1. *Entrevista*: Se emplea en la fase inicial para el entendimiento del negocio y levantamiento de requisitos. Se realizan una serie de preguntas para lograr el cumplimiento de los objetivos propuestos a satisfacer con este método. Ver anexo 1.

La estructura de este trabajo de investigación consta de introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y anexos:

**Capítulo #1. Fundamentación teórica:** Recoge aspectos concernientes al objeto de estudio, se hace una revisión de la bibliografía donde se analizan los principales conceptos sobre los Sistemas de Información Geográfica como herramienta de representación y análisis de entes de contaminación. Se realiza el estudio de diferentes Sistemas de Información Geográfica dedicados al estudio epidemiológico. Además se reúnen todas las tecnologías y herramientas a utilizar en la solución propuesta.

**Capítulo #2. Presentación de la solución propuesta:** Contiene información asociada al sistema, descripción del modelo de dominio, especificación de los requisitos de software y definición de los casos de uso y patrones de casos de uso utilizados, incluye el diseño de la solución propuesta, la arquitectura utilizada, los patrones de diseño que se usaron, los diagramas de clases del diseño, y el diseño de la base de datos.

**Capítulo #3. Construcción y validación de la solución propuesta:** Contiene los diagramas de despliegue y de componentes del flujo de trabajo de Implementación; y del flujo de trabajo de Pruebas las pruebas realizadas al sistema, con la descripción de los casos de prueba.

Con la culminación de todas las tareas y actividades desarrolladas en el proceso de investigación se esperan los siguientes **resultados**:

- ✓ Un sistema que permita la representación y análisis de la información de los entes de contaminación y mejore el proceso de tratamiento por parte de las instituciones de sanidad.
- ✓ La documentación técnica del proceso ingenieril correspondiente al diseño e implementación del Sistema de Información Geográfica para la representación y análisis de entes contaminantes en áreas urbanas de riesgo por contaminación.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En el presente capítulo, se analizan los principales conceptos sobre los Sistemas de Información Geográfica como herramienta de representación y análisis, con el propósito de brindar al lector un mayor dominio del problema. Por otra parte se realiza el análisis de SIG que realizan procesos similares al sistema que se desea desarrollar, que servirán como punto de partida para el desarrollo de la solución. Además se exponen las principales herramientas, lenguajes y tecnologías que se utilizarán en el desarrollo de la solución.

### 1.2 Conceptos asociados al dominio del problema

A continuación se brindan varios conceptos que se consideran indispensables para el correcto entendimiento de la investigación realizada.

#### **Sistema de información**

“Un sistema de información es un conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a un determinado objetivo” (Rae, 2015).

#### **Sistemas de Información Geográfica**

Un Sistema de Información Geográfica es el conjunto de métodos, herramientas y actividades que actúan coordinada y sistemáticamente para recolectar, almacenar, validar, manipular, integrar, analizar, actualizar, extraer y desplegar información, tanto gráfica como descriptiva de los elementos considerados, con el fin de satisfacer múltiples propósitos (Mateus Pulido, 2009).

#### **Sistemas de Información Geográfica en Salud Pública**

Desde la perspectiva de la salud pública, el concepto de SIG incluye el diseño, desarrollo y utilización de la tecnología de SIG en la descripción y estudio de problemas de salud en las poblaciones, el análisis de la situación de salud, la vigilancia de eventos de salud, el análisis epidemiológico, así como la gestión y toma de decisiones (Carlos, 2003).

En su naturaleza multidisciplinaria, los SIG en salud pública contemplan disciplinas como la epidemiología, la bioestadística, la geografía y la tecnología de la información para responder a sus objetivos. Un paquete de programas de SIG en Salud Pública incluye, además de funciones genéricas,

las específicas para el análisis de datos relacionados con salud y el abordaje de problemas de la salud pública (Boletín Epidemiológico, 2004).

## **Georeferenciación**

Se refiere a la posición en la que se define la localización de un objeto espacial en un sistema de coordenadas (Investigación, 2015).

## **Datos geoespaciales**

Un dato geoespacial es aquel que hace referencia a un espacio geográfico cuya ubicación se conoce bajo un sistema de coordenadas. Los datos geoespaciales se refieren a información sobre la localización geográfica de una entidad. Esto habitualmente implica el uso de unas coordenadas geográficas como los valores latitud o longitud (Torchetti, 2015).

## **Aplicación informática**

Una aplicación informática permite a un usuario utilizar una computadora con un fin específico. Las aplicaciones son parte del software de una computadora, y suelen ejecutarse sobre el sistema operativo (Diccionario, 2015).

## **Aplicaciones web**

Una aplicación web es cualquier aplicación que es accedida vía web por una red como internet o una intranet. En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador (Diccionario, 2015).

## **1.3 Objeto de Estudio: Sistemas de Información Geográfica como herramienta de representación y análisis.**

La información geográfica es un conjunto de datos espaciales que son referenciados, es decir, son localizados en la superficie de la tierra por medio de un sistema de coordenadas determinado. Son requeridos por varios tipos de usuarios con el fin de conocer las características de distintos fenómenos para utilizarlos en investigaciones científicas, administrativas, entre otras. De ahí la importancia de poder almacenarlos y representarlos. Con el objetivo de lograr mayor eficiencia en la gestión y la manipulación de la información se han desarrollado herramientas avanzadas para la gestión de la misma como soluciones a los diferentes problemas que se identifican en disímiles ramas de la sociedad. La cartografía digital permite representar, analizar la información geográfica en formato digital. Ofrece varias ventajas como son la representación de mapas para satisfacer determinadas

necesidades y que la actualización de los mismos se haga de una forma más rápida y efectiva. Precisamente para lograr dicha representación se utilizan los Sistemas de Información Geográfica, no es más que un sistema capaz de editar, interpretar, almacenar y mostrar información geográfica georreferenciada.

Algunas de las características que poseen son:

- ✓ Permiten tratar grandes cantidades de información.
- ✓ Permiten trabajar con información de diferentes fuentes y escalas.
- ✓ Permiten modelar información.
- ✓ Permiten el manejo de información georreferenciada.

Un SIG se compone de elementos básicos. Cada uno de ellos cumple un papel determinado dentro del sistema SIG, el cual se caracteriza fundamentalmente por su naturaleza integradora. Cinco son los elementos principales que se contemplan en este aspecto:

- ✓ Datos: Los datos es el componente necesario para el trabajo en un SIG, y los que contienen la información geográfica vital para la propia existencia de los SIG.
- ✓ Métodos: Un conjunto de formulaciones y metodologías a aplicar sobre los datos.
- ✓ Software: Es necesaria una aplicación informática que pueda trabajar con los datos e implemente los métodos anteriores.
- ✓ Hardware: El equipo necesario para ejecutar el software.
- ✓ Personas: Las personas son las encargadas de diseñar y utilizar el software, siendo el motor del sistema SIG.

## Representación de los datos

Existen dos formas de almacenar los datos en un SIG: *vectorial* y *ráster*.

Vectorial: Es una forma de representar la información digital a través de la combinación de puntos, líneas y polígonos. Permite realizar ampliaciones a las imágenes sin que estas pierdan resolución posibilitando mayor calidad de la información mostrada. Es el formato que más se corresponde con la representación de objetos reales llevados al espacio como ríos, carreteras, entre otros.

Ráster: Es una forma de representar la información digital a través de celdas, donde cada una tiene un único valor que se considera representativo para toda la superficie y es asignado según el atributo de

la superficie al que pertenezca. Se utiliza en la representación de fenómenos como la contaminación atmosférica, distribución de temperaturas, localización de especies marinas, entre otros.

Los SIG que se centran en el manejo de datos en formato vectorial son más populares en el mercado. No obstante, los SIG ráster son muy utilizados en estudios medioambientales donde no se requiere una excesiva precisión espacial (contaminación atmosférica, distribución de temperaturas, localización de especies marinas, análisis geológicos, etc.) (SIG, 2015).

Se identifican tres principales clasificaciones de SIG:

SIG de Escritorios: Son aquellos que se utilizan para crear, editar, administrar, analizar y visualizar los datos geográficos sobre entornos de escritorio. Estos según la finalidad a la que vayan destinados se clasifican en Visores SIG, Editores SIG y SIG de análisis.

- ✓ Un Visor SIG permite visualizar información geográfica a través de un formulario al cual se le agregan varias capas de información.
- ✓ Un Editor SIG realiza tratamiento previo de información para su análisis. Antes de adicionar los datos estos se preparan para su uso transformándolos a un formato entendible por el software.
- ✓ Los SIG de Análisis presentan funcionalidades de análisis espacial y modelización cartográfica de procesos.

SIG Web: Son utilizados para mostrar, editar y analizar cartografía en entornos web. Permiten el acceso a grandes volúmenes de datos geográficos. Ofrecen por lo general imágenes aéreas o de satélite. Este tipo de SIG son los más utilizados debido al rápido acceso a través de un navegador web.

SIG Móviles: Son utilizados para crear, manipular, analizar y representar información geográfica en los dispositivos móviles. Estos SIG se pueden instalar en el mismo dispositivo o se pueden acceder a ellos mediante la utilización del Protocolo de Aplicaciones Inalámbricas a través de un navegador.

En los últimos años ha habido un interés creciente de los Ministerios de Salud y otras instituciones del sector en el uso de los SIG como herramienta para fortalecer las capacidades de análisis, gestión, monitoreo y toma de decisiones en salud pública. Este interés se ha visto favorecido por i) el reconocimiento de las capacidades de los SIG para manejar la dimensión de espacio geográfico, integrar datos relacionados con salud proveniente de diversas fuentes, ayudar a visualizar información geográficas de los datos, mostrándolos en mapas, que son un medio más expresivo y visual; ii) la inclusión del tema de los SIG en eventos científicos de epidemiología y salud pública; iii) la publicación

de un número significativo de artículos dedicados a ejemplificar las potencialidades y usos de los SIG en diversos ámbitos, desde los estudios epidemiológicos, la gestión de salud pública y el mejoramiento de la salud comunitaria, entre otros; y, iv) el número creciente de estudios y proyectos en salud, que incluyen el uso de SIG como herramienta de análisis y de comunicación de los resultados obtenidos (Boletín Epidemiológico, 2004).

## 1.4 Situación Problemática

Actualmente el exceso de basura constituye uno de los problemas más acuciantes que la sociedad tiene que afrontar debido al acelerado crecimiento de la población, aumento de la producción y tendencias crecientes en los hábitos de consumo, entre otros.

A pesar de las grandes cantidades de basura, causando deterioro de los recursos naturales y la contaminación del entorno, la dificultad para encontrarles un destino final aceptable, es lo que más suele preocupar; es aquí donde se introduce el término de residuo. Los residuos sólidos urbanos comprenden todo material que sea desechado por la población; pueden ser de origen doméstico, comercial, industrial, desechos de la vía pública y resultantes de la construcción. Estos residuos son los que mayor dificultad tienen en su tratamiento, debido a la diversidad de los mismos y por ser generados en los domicilios, lo que provoca mayor riesgo para los pobladores.

Estos provocan gran impacto en la salud de la población ya que los riesgos pueden ser directos e indirectos. Directos, porque pueden producir enfermedades a las personas que están en contacto con la basura. Indirectos, porque originan la proliferación de vectores, tales como insectos y roedores, potenciales transmisores de enfermedades que incluso pueden provocar la muerte. Además sobre el medio ambiente pueden ocasionar contaminación del suelo, las aguas, la atmósfera y deterioro del paisaje.

La basura suele estar compuesta por diferentes materiales.

*Materia orgánica:* Restos procedentes de la limpieza o la preparación de alimentos junto a la comida que sobra, ramas, paja, restos de animales y plantas.

*Papel y cartón:* Periódicos, revistas, cajas y embalajes, etcétera.

*Plásticos:* Bolsas, embalajes, platos, vasos y cubiertos desechables, y otros.

*Vidrio:* Botellas, frascos diversos, vajilla rota.

*Metales:* Latas, tapas, etcétera.

*Otros:* Barro, arena.

## Tratamiento de los residuos

Se denomina tratamiento de residuos sólidos urbanos al conjunto de operaciones encaminadas a la eliminación de estos residuos o al aprovechamiento de los materiales que contienen.

Los residuos sólidos urbanos constituyen la biomasa residual más aprovechable, ya que está concentrada; es imprescindible su recogida y su transporte.

El concepto de valorizar engloba todas aquellas operaciones mediante las cuales un residuo se vuelve a utilizar total o parcialmente. Básicamente, los sistemas de valoración de los residuos son:

*Reutilización:* Utilización de un residuo como producto final en su forma original, para el mismo o diferente uso.

*Reciclaje:* Proceso que tiene por objeto la recuperación de forma directa o indirecta de los componentes que contienen los residuos.

*Regeneración:* Operación de valorización mediante la cual un residuo es devuelto a sus características originales de forma total o parcial, y que permite su uso en el mismo estado que tenía antes de transformarse en residuo.

*Recuperación:* Operación de valorización mediante la cual se extrae los recursos del residuo.

## **Vertedero controlado o relleno sanitario**

El relleno sanitario consiste en el enterramiento ordenado y sistemático de los residuos sólidos utilizando criterios de ingeniería que permitan su confinamiento en el menor volumen posible y cubriéndolos con una capa de tierra al concluir las operaciones diarias, o según se considere necesario. Cualquier método de tratamiento de los residuos sólidos requerirá de un vertedero controlado para la disposición final.

Una vez retirados los residuos peligrosos y la materia orgánica fermentable, el resto de los residuos contienen materiales que podrán ser reciclados, por ejemplo:

Con el papel, las telas y el cartón se hacen una nueva pasta de papel, lo que contribuye a talar menos árboles.

Con el vidrio se pueden fabricar nuevas botellas y envases sin necesidad de más materias primas y con mucho menor gasto de energía.

Los plásticos se pueden usar para fabricar nueva materia prima y construir objetos diversos.

Los residuos no aprovechables, aunque pudieran ser reciclados, se podrán comprimir y cubrir con tierra de cobertura. Uno de los mayores riesgos es que contaminen las aguas subterráneas y para evitarlo se debe impermeabilizar bien el suelo del vertedero y evitar que las aguas de lluvias y otras salgan sin tratamiento del vertedero, arrastrando contaminantes al exterior.

Otro riesgo está en los malos olores y la concentración de gases explosivos producidos al fermentarse la basura.

Por otro lado, el relleno sanitario constituye la alternativa inmediata al vertedero incontrolado y a los botaderos a cielo abierto. Para la explotación del relleno sanitario como recurso energético es necesario conocer de los basureros su ubicación y volumen estimado, así como las vectores o roedores que pudieran habitar en la basura depositada, con el fin de prevenir o combatir enfermedades ocasionadas por los mismos.

## **1.5 Análisis de soluciones existentes**

El desarrollo y la diversificación de los Sistemas de Información Geográfica (SIG) posibilitan que actualmente estas poderosas herramientas sean aplicables en un campo tan sensible e importante como la Salud Pública.

En este campo, los SIG se han descrito como un conjunto de datos de orden espacial y sanitario que interactúan, permitiendo analizar y sintetizar una gran cantidad de datos con el fin de describir una situación de salud, efectuar análisis epidemiológicos y orientar y evaluar la gestión, las intervenciones y la toma de decisiones en el ámbito de la salud (Carlos, 2003). A continuación se muestran algunas aplicaciones SIG muy utilizadas en estos días.

### **1.5.1. SIG a Nivel Mundial: Sistema de Información Geográfica para el Análisis Epidemiológico (SIGEPI).**

SIGEpi es un sistema de información geográfica desarrollado por el Área de Análisis de Salud y Sistemas de Información (AIS) de la Organización Panamericana de la Salud (OPS). Fue creado para emplearlo en Epidemiología y Salud Pública. Brinda un conjunto de técnicas, instrucciones y métodos para el análisis de datos bioestadísticos, epidemiológicos y geográficos de manera eficiente para el apoyo en la toma de decisiones en salud pública (Carlos, 2003).

En la figura 1 se muestra la interfaz gráfica de SIGEpi, la cual está diseñada para manejar múltiples tipos de documentos en un único entorno de trabajo. Cada tipo de documento se presenta en ventanas específicas, manteniendo el enlace dinámico a algunos de ellos. Los documentos tipo son: Proyecto, Mapa, Tabla, Gráfico, Resultado y Presentación; cada uno de ellos con funciones, menús, botones y herramientas propias (Boletín Epidemiológico, 2004).

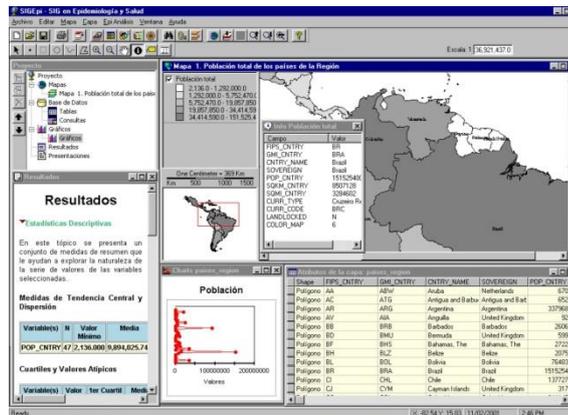


Fig. 1 Interfaz gráfica de SigEpi (Epidemiológico, 2001).

A continuación se presenta un resumen de las principales funciones de procesamiento y análisis que ofrece SIGEpi, según áreas de aplicación (Martínez Piedra, 2001).

#### a. Funciones básicas y procesamiento de datos espaciales

Las funciones de manejo y procesamiento de datos son: la georeferenciación o el trazado de puntos en el mapa a partir de tablas de datos con variables de latitud y longitud; operaciones de geo-procesamiento tales como creación de áreas de influencia para delimitar las áreas de impacto o influjo, entre otras. Otra función esencial de SIGEpi es la creación de mapas temáticos, tales como mapas de valores únicos, de densidad de puntos, de gráficos de barras y pastel, y de intervalos o rangos con distintos métodos de clasificación.

#### b. Métodos útiles en la Práctica de la Salud Pública

A SIGEpi se han incorporado algunos métodos que son útiles en procesos de análisis y toma de decisiones en la práctica de la salud pública, como: la identificación de áreas críticas y prioritarias; la construcción de un índice compuesto de salud, de necesidades básicas insatisfechas en salud, de pobreza, entre otros.

### 1.5.2. SIG a Nivel Mundial: Sistema de Información Geográfica para el Mapeo de Enfermedades (EpiInfo/EpiMap).

A su vez, la División de Vigilancia en Salud Pública e Informática del CDC<sup>1</sup> de Atlanta, Estados Unidos, ha desarrollado la herramienta para el mapeo de enfermedades EpiMap/EpiInfo, como parte del

<sup>1</sup> Centro de Control y Prevención de Enfermedades

paquete de análisis epidemiológico. Es una herramienta analítica indispensable para cualquier equipo y unidad de salud dedicada a realizar investigaciones y estudios epidemiológicos, contiene una amplia colección de métodos que responde a la mayor parte de las necesidades de estudios epidemiológicos, estudios de brotes y otros procesos de la epidemiología y salud pública (CDC. EpiInfo, 2014).

En la figura 2 se muestra la interfaz de EpiInfo, la cual está conformada por un conjunto de programas estrechamente relacionados, MakeView permite la creación de cuestionarios y la base de datos, Enter ofrece las funciones para la entrada, edición, validación y consistencia de los datos auxiliándose del cuestionario diseñado. Una vez que los datos están introducidos y validados, el programa Analysis permite realizar el análisis estadístico y epidemiológico de los datos, presentando los resultados en forma de tablas y gráficas, EpiReport se ocupa de combinar las salidas del análisis y otros procesos a los datos, permitiendo su presentación en forma profesional, lo que puede ser guardado como archivo en formato HTML para su publicación y distribución en Internet. EpiMap es el programa que permite el despliegue en mapas de los resultados de los análisis hechos en EpiInfo, así como otros datos e indicadores. Contiene un número mínimo de funciones de un SIG, permitiendo la georeferenciación de datos de individuos y eventos de salud a partir de variables con referente geográfico como la dirección de residencia o trabajo, y otros referidos a áreas geográficas. Permite además la creación de mapas temáticos de intervalos, de símbolos graduados, de valores únicos y densidad de puntos.

EpiInfo y EpiMap han sido desarrollados con el objetivo de brindar a los servicios de salud pública, epidemiólogos y otros profesionales de la salud, una herramienta de análisis epidemiológico de bajo costo y fácil de utilizar, capaz de administrar datos, realizar análisis epidemiológico de datos de salud y desplegar los resultados en forma de mapas. EpiInfo/EpiMap es un sistema de programas cuyo diseño y desarrollo ha estado guiado por la necesidad de cubrir las principales funciones y procedimientos de la vigilancia de enfermedades y diferentes tipos de estudios epidemiológicos, respondiendo a demandas del CDC y la OMS<sup>2</sup> (Boletín Epidemiológico, 2004).

---

<sup>2</sup> Organización Mundial de la Salud.



Fig. 2 Interfaz gráfica de EpiInfo (CDC. EpiInfo, 2014).

### 1.5.3. SIG en Cuba: Sistema de Información Geográfica para la Gestión de la Estadística de Salud de Cuba (SIG-ESAC).

SIG -ESAC se crea con el objetivo de facilitar la gestión de la estadística de salud del país. El mismo permite cartografiar y realizar diferentes tipos de análisis de significativos indicadores de salud como son: morbilidad, mortalidad, demográficos, recursos y servicios.

Se crea principalmente para la cartografía bioestadística, aunque se le añaden algunas herramientas para estudios epidemiológicos. Tiene mucha utilidad para las oficinas de estadística provincial y nacional del MINSAP, ya que ayuda al estudio de la distribución espacial de diferentes objetivos, lo que facilita la identificación de zonas geográficas y grupos de población que presentan más riesgo de padecer o morir por alguna patología, entre otros fines.

En la figura 3 se puede observar la pantalla de inicio del sistema con los diferentes menús creados para la cartografía. SIG-ESAC posee cuatro menús con los cuales, con solo ejecutar un clic, los usuarios logran la representación del índice de salud deseado, ya sea el de alguna provincia específica o del país completo, siempre utilizando la división político-administrativa de municipio como el nivel más elemental para la cartografía (Núñez, 2006).

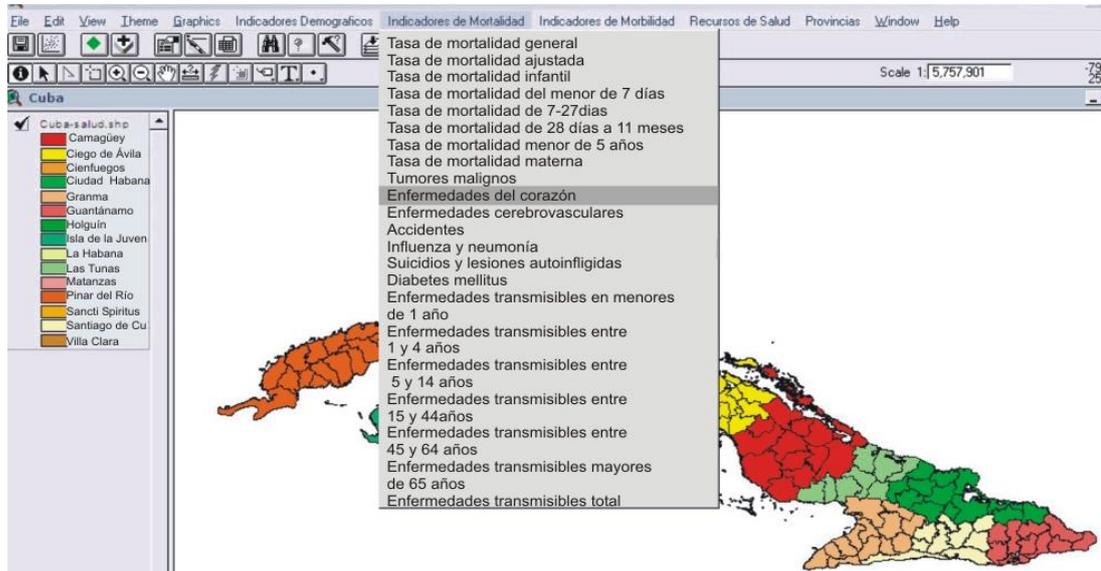


Fig. 3 Pantalla de inicio del SIG-ESAC con uno de los menús personalizados (Núñez, 2006).

#### 1.5.4. Observación de las soluciones existentes

El estudio de un conjunto de Sistemas de Información Geográfica dedicados al análisis epidemiológico, al mapeo de enfermedades o estadísticas de salud, correspondientes a nivel mundial así como en Cuba, permitió identificar un conjunto de funcionalidades básicas que debe poseer la aplicación que se desea construir como la creación de áreas de influencia para delimitar las áreas de impacto de las epidemias. Otra función esencial que se identificó es la creación de mapas temáticos, para cada uno de los posibles riesgos que pueden acarrear los basureros, lo que facilita la identificación de zonas geográficas y grupos de población que presentan más riesgo de padecer o morir por alguna patología, entre otros fines.

Se debe puntualizar que los SIG correspondientes a nivel nacional e internacional solo podrían servir de base para la identificación del objetivo antes mencionado, no para su utilización, ya que son aplicaciones específicas de uno u otro lugar en particular, presentando características muy específicas de información socioeconómica de interés para una u otra entidad específica, de acuerdo a las características propias de cada país o región.

Después de realizado dicho análisis, se comprueba que los sistemas existentes no cumplen en su totalidad con los requisitos que se proponen para la elaboración del sistema acorde a sus necesidades, ya que se necesita tener la ubicación exacta de los entes de contaminación para su erradicación sino

que ofrecen características y conceptos útiles para el desarrollo de la propuesta de solución, por lo que se concluye que no se utilizaran para dar respuesta a la problemática establecida anteriormente.

Para el desarrollo de dicha propuesta de solución, se hizo necesario realizar el estudio y selección de las metodologías, herramientas y tecnologías que se pueden utilizar para obtener un producto con la calidad requerida y con un óptimo funcionamiento las cuales se exponen a continuación.

## 1.6 Plataforma, herramientas y framework de desarrollo

A continuación se detalla la plataforma que se utiliza para el desarrollo de la solución propuesta, así como las herramientas y framework de desarrollo utilizados con este fin.

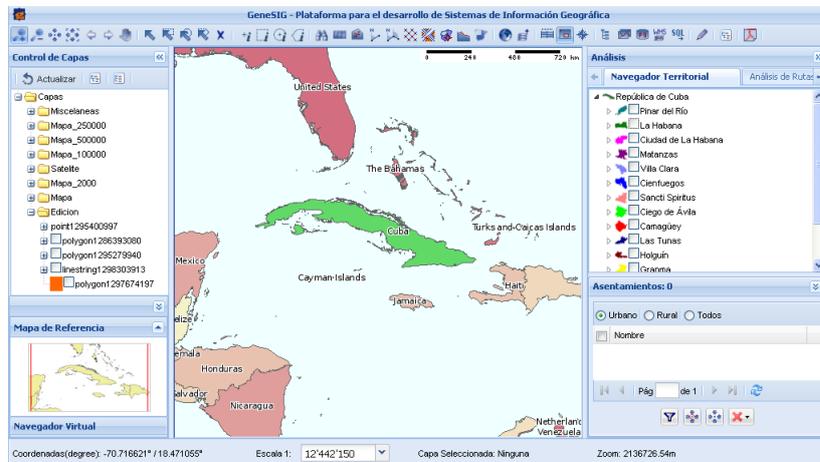
### 1.6.1. GeneSIG 1.5

La plataforma GeneSIG surge como necesidad de contar con un producto soberano que sirva como soporte al desarrollo de aplicaciones de SIG en entornos libres. En los últimos tiempos se ha incrementado exponencialmente el uso de los SIG para la toma de decisiones, relacionado íntimamente con el incremento de la efectividad y la disminución de los costos que se obtienen con la puesta en marcha de un sistema de este tipo. Tiene como objetivo fundamental realizar la representación geoespacial de la información asociada a negocios específicos, permitiendo además realizar análisis sobre dicha información. GeneSIG fue desarrollada por el centro de desarrollo GEYSED en conjunto con XETID perteneciente a la Universidad de las Ciencias Informáticas, implementada con herramientas y tecnologías libres, cumpliendo con la necesidad actual de migración para lograr la independencia tecnológica. Su construcción está basada en estándares OpenGIS<sup>3</sup>. En la figura 4 se muestra la interfaz de la plataforma con todas sus funcionalidades, además puede implantarse personalizaciones en diferentes servicios o ramas como pueden ser:

- ✓ Servicio informativo de transporte urbano, del correo postal, de turismo de ciudad y mejores ofertas de ventas.
- ✓ Control Epidemiológico.
- ✓ Defensa Civil.
- ✓ Otros.

---

<sup>3</sup> Estándar internacional orientado a Sistemas de Información Geográfica.



**Fig. 4 Interfaz de la plataforma GeneSIG (Espinosa, y otros, 2012).**

GeneSIG está desarrollado sobre las siguientes tecnologías y herramientas: CartoWeb como framework, PHP 5.3 del lado del servidor y ExtJS 3.0 del lado del cliente, implementado en los entornos integrados de desarrollo Netbeans 8.0, haciendo uso de la herramienta CASE<sup>4</sup> Visual Paradigm 8.0, sobre Sistemas Operativos Ubuntu 12.04, MapServer 6.0 como servidor de mapas y Servidor Web Apache 2.2. La utilización de GeneSIG para la construcción del sistema permite reducir el tiempo de desarrollo, debido a la cantidad de módulos ya implementados que realizan las funcionalidades más comunes en un SIG. Es importante destacar la madurez de la plataforma además de la aceptación entre los desarrolladores que la utilizan, sin dejar de mencionar que sus módulos se pueden adaptar a diferentes negocios.

### ExtJS 3.0

ExtJS es una librería JavaScript que permite construir aplicaciones complejas. Tiene la funcionalidad de crear interfaces de usuario funcionales. Es compatible con los navegadores más usados a nivel mundial y fundamental dentro de la plataforma GeneSIG, usada en la construcción de componentes para interfaces de usuario de gran rendimiento, es responsable de la programación del lado del cliente. Garantiza un balance entre cliente y el servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo. Gestiona la comunicación asíncrona y su uso es relativamente fácil (Espinosa, y otros, 2012).

<sup>4</sup> Computer Aided Software Engineering (Ingeniería de Software Asistida por Computadora).

## **CartoWeb**

CartoWeb es un framework para el desarrollo de Sistemas de Información Geográfica de entorno Web, adecuado para la creación de aplicaciones avanzadas y altamente personalizable. Desarrollado por Camptocamp S.A<sup>5</sup>, distribuido bajo licencia GPL<sup>6</sup> y escrito en PHP. Muestra su verdadero poder cuando está asociado a PostgreSQL / PostGIS. CartoWeb puede ser configurado como un servicio Web SOAP<sup>7</sup> que permite tener los servidores Web y los servidores de datos de forma distribuida (Camptocamp SA., 2008). Este framework brinda a GeneSIG la posibilidad de trabajar con mapas entre otras funcionalidades, además de una estructura física de componentes que en principio es la misma que propone el Cartoweb. Esto permite la reutilización de dichos componentes en otros proyectos y la separación de la lógica del servidor (cartoserver), encargado del diálogo con el servidor de mapas y la provisión de servicios a un cliente (cartoclient) (Espinosa, y otros, 2012).

## **Servidor de Mapas: MapServer 6.0**

MapServer es una plataforma de código abierto para la publicación de datos espaciales y aplicaciones cartográficas interactivas para la web (MapServer, 2015). Adquiere el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología IMS<sup>8</sup>. Usualmente MapServer trabaja detrás de un servidor Web, este es el que recibe los requerimientos de mapas y los pasa a MapServer para que los cree. MapServer genera la imagen del mapa requerido y la entrega al servidor Web Apache el cual la transmite de vuelta al usuario (Moreta, 2009). Esto es lo que se muestra en la figura 5. La selección de este servidor de mapas se basa en su rendimiento, además de ser tecnología libre se ajusta perfectamente a Sistemas de Información Geográfica Web.

---

<sup>5</sup> Empresa orientada a servicios e integración de aplicaciones de software libre para soluciones geoespaciales. Sociedad Anónima

<sup>6</sup> General Public License (Licencia Pública General).

<sup>7</sup> Simple Object Access Protocol (Protocolo Simple de Acceso a Objetos).

<sup>8</sup> Internet Map Server (Servidor de Mapas de Internet).

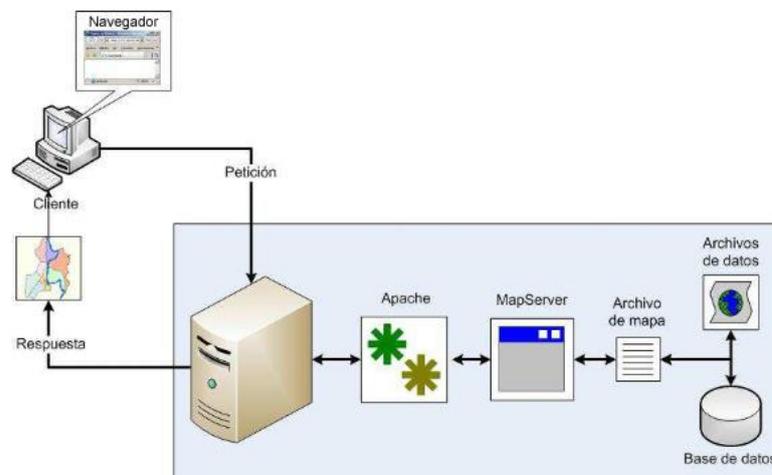


Fig. 5. Componentes de MapServer (MapServer, 2015).

## Servidor Web: Apache 2.2

Apache es un servidor Web de código abierto para plataformas Unix, Microsoft Windows, Macintosh, potente y estable, sencillo para las tareas más comunes de mantenimiento. Es altamente configurable, su robustez y estabilidad. Es un servidor seguro, eficiente y extensible que proporciona servicios HTTP en sincronía con los estándares HTTP actuales (The Apache Software Apache, 2012). El servidor de mapas MapServer necesita un servidor Web Apache para su funcionamiento en plataformas de entorno libre sobre la cual se desarrolla el SIG, además de poseer características como robustez, licencia de código abierto entre otras.

### 1.6.2. UML 2.0 como Lenguaje Unificado de Modelado

La falta de estandarización en la forma de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores. Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema.

Los diseños logrados usando UML se pueden realizar en cualquier lenguaje orientado a objetos ya que este se aplica a una multitud de diferentes tipos de sistemas, dominios, y métodos o procesos. UML se seleccionó como lenguaje de modelado para el desarrollo de la aplicación puesto que es el lenguaje estándar de modelado más popular para crear planos de software, además de visualizar, especificar, construir, documentar y detallar los artefactos del sistema. Es un lenguaje que permite modelar estructuras complejas utilizando técnicas orientadas a objetos.

## 1.6.3. Sistemas Gestores de Bases de Datos

### PostgreSQL 9.1

PostgreSQL es un potente gestor de base de datos, bajo licencia BSD<sup>9</sup>. Cuenta con más de 15 años de desarrollo activo y arquitectura probada que ha ganado mucha reputación por su confidencialidad e integridad en los datos (PostgreSQL, 2013). Su uso se debe a que este gestor está diseñado para ambientes de altos volúmenes de datos. Al ser multiplataforma está disponible en casi cualquier sistema operativo. En términos de eficiencia y recursos, es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, lo que posibilita atender un mayor número de peticiones concurrentemente. GeneSIG propone su uso para el desarrollo de sistemas con dicha plataforma.

### PostGIS 1.5

PostGIS es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial y así posibilitando su uso en Sistemas de Información Geográfica. Publicado bajo Licencia Pública General de GNU<sup>10</sup> (MappingGis, 2015). PostGIS es muy utilizado actualmente ya que ha demostrado estabilidad y eficiencia en relación con otros productos, demostrando superioridad sobre extensiones geográficas como la de MySQL, es similar a la versión geográfica del potente gestor de bases de datos Oracle. Este sistema ha sido certificado por el Open Geospatial Consortium (OGC) lo que garantiza la interoperabilidad con otros sistemas. PostGIS proporciona formas de almacenado que favorecen y facilitan su uso. Soporta estructura de datos para líneas, puntos y polígonos, las cuales son muy usadas en sistemas como el que se propone.

## 1.6.4. NetBeans como IDE para PHP

NetBeans es un IDE<sup>11</sup> que posee un excelente completamiento de código. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso (NetBeans IDE, 2015). Permite desarrollar aplicaciones para diferentes entornos como escritorio y Web, utilizando como lenguaje Java, C/C++, PHP, JavaScript entre otros. Presenta mejoras

---

<sup>9</sup> Berkeley Software Distribution (Distribución de software Berkeley).

<sup>10</sup> Licencia Pública General

<sup>11</sup> Integrated development environment (Entorno Integrado de Desarrollo).

para SOA<sup>12</sup> y UML<sup>13</sup>. Después de haber analizado varias herramientas de este tipo, e identificar características similares, como licencias, completamiento de código e integración con los lenguajes de programación empleados para la construcción del SIG, se decide su uso de esta forma se eleva la productividad y se garantiza la calidad del producto final (Palau, y otros, 2012).

## 1.6.5. Visual Paradigm como Herramienta Case

Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software permiten incrementar la productividad y el control de la calidad en cualquier proceso de elaboración de software, pues transforman la actividad de desarrollar software en un proceso automatizado (Valle, 2009). Visual Paradigm 8.0 utiliza UML como lenguaje de modelado. Esta herramienta está pensada para usuarios interesados en construir sistemas de software fiables con el uso del paradigma orientado a objetos, incluyendo actividades entre las que se destacan: ingeniería de software, análisis de sistemas y análisis de negocios. La utilización de la herramienta en cuestión ayudará a aumentar la productividad durante el desarrollo del SIG, automatizando parte del proceso, además, la documentación será enriquecida con diagramas para lograr mejor entendimiento del funcionamiento del sistema por futuros desarrolladores.

## 1.7 Lenguajes de Programación

Las máquinas de cómputo, necesitan un lenguaje el cuál interpretan para llevar a cabo instrucciones, mediante las cuales se controla su comportamiento y se realizan diferentes operaciones. Algunos de ellos son Pascal, Python, C++, Java y PHP<sup>14</sup>. El lenguaje de programación está conformado por una serie de reglas sintácticas y semánticas que son utilizadas por el programador y a través de las cuales creará un programa. Todas estas instrucciones escritas por el desarrollador y que forman dicho programa es a lo que se denomina código fuente (Lanzillotta, 2005).

Los lenguajes de programación pueden ser interpretados o compilados. Los interpretados, usan otro programa (Intérprete) para traducir las instrucciones a código máquina. Por lo general son más lentos que los compilados ya que necesitan un intermediario entre él y la máquina, pero ganan en portabilidad ya que haciendo uso de su intérprete pueden ejecutarse en múltiples plataformas. Los lenguajes compilados solo necesitan un compilador que traduzca el código a código máquina, generando

---

<sup>12</sup> Service Oriented Architecture (Arquitectura Orientada a Servicios).

<sup>13</sup> Unified Modeling Language (Lenguaje Unificado de Modelado).

<sup>14</sup> Hypertext Preprocessor (Preprocesador de Hipertexto).

instrucciones compiladas que se ejecutan directamente en el ordenador, suelen ser mucho más rápidos en tiempo de ejecución.

## 1.7.1. Lenguajes web de programación del lado del cliente

Los lenguajes del lado del cliente basan su procesamiento en el cliente web, es decir, que se ejecutan en el navegador del usuario. En este sentido existen varios lenguajes, pero la presente investigación se asume JavaScript para la solución que se propone.

### JavaScript

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.

Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador aunque existen implementaciones de dicho lenguaje del lado del servidor conocidos como SSJS<sup>15</sup> (Espinosa, y otros, 2012).

El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único con que cuenta este lenguaje, es el propio navegador.

Debido a su compatibilidad con la mayoría de los navegadores modernos, es actualmente el lenguaje más utilizado. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, en ocasiones con ligereza. Una de sus ventajas radica en que las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia. Es un lenguaje de programación interpretado (Palau, y otros, 2012).

## 1.7.2. Lenguajes web de programación del lado del servidor

Los lenguajes de lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. GeneSIG está programado en PHP 5 y está enfocado al desarrollo de aplicaciones web en el mismo

---

<sup>15</sup> acrónimo de Server-side JavaScript

lenguaje de programación. La versión mínima de PHP requerida para ejecutar GeneSIG es PHP 5. 2 (Espinosa, y otros, 2012).

## **PHP 5. 3**

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, y con extensa documentación. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores (Palau, y otros, 2012).

### **PHP/ Mapscript**

*“Las librerías Mapscript de PHP constituyen la vía de comunicación de las aplicaciones SIG con el servidor de mapas Mapserver y por otra parte rompen en cierta medida ésta rigidez de la representación de mapas a través de los ficheros “Mymapfile. map” permitiendo modificar el mapfile en tiempo de ejecución, a los cuales se les conoce como mapfile dinámicos, facilitando de esta forma la creación de aplicaciones con un grado de personalización mayor, eventualmente no alcanzado con aplicaciones del MapServer” (Espinosa, 2000).*

## **1.7.3. Técnicas y Formatos de intercambio de datos**

### **JSON**

JSON<sup>16</sup> es un formato ligero de intercambio de datos. Para los seres humanos es fácil leerlo e interpretarlo, al igual que para las máquinas generarlos. Es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidas por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos (Json, 2011). La comunicación en el sistema entre el cliente y el servidor se realiza utilizando el formato JSON, y de esta forma se unifican los lenguajes de programación JavaScript y PHP para dar solución a diferentes tareas, cada uno en el contexto que se desarrolla (Palau, y otros, 2012). La claridad y simplicidad para manipular los datos transferidos son características fundamentales a tener en cuenta en el desarrollo de aplicaciones Web.

---

<sup>16</sup> JavaScript Object Notation, Notación de Objetos de JavaScript

## **AJAX**

El Ajax<sup>17</sup> es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML (Espinosa, y otros, 2012).

## **1.8 Metodologías de desarrollo**

En la actualidad la construcción de una aplicación informática eficiente que cumpla con los requerimientos planteados es una tarea intensa y difícil de cumplir. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de los mismos con el fin de hacerlo más predecible y eficiente. Estas son usadas para estructurar, planificar y controlar el proceso de desarrollo. Estas garantizan el cumplimiento de la planificación de producción y tienen como objetivos guiar a los desarrolladores en la creación de un producto de calidad que cumpla con los requerimientos planteados por el cliente.

### **Agile Unified Process (AUP) o Proceso Unificado Ágil**

Es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP (AUP, 2015).

La metodología AUP es más simple que RUP porque reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP. Dispone de cuatro fases igual que RUP: Inicio, Elaboración, Construcción y Transición (AUP, 2015).

---

<sup>17</sup> acrónimo de Asynchronous JavaScript And XML

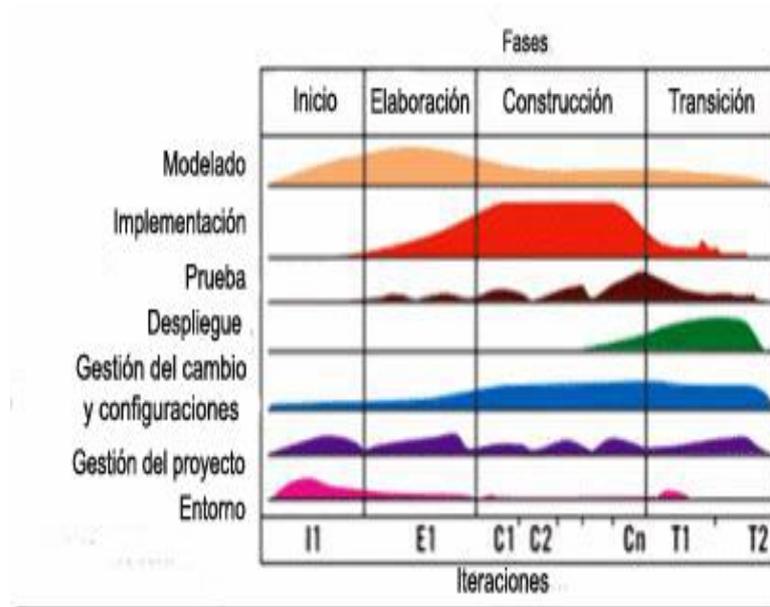


Fig. 6 . Ciclo de vida del proceso unificado ágil (AUP, 2015).

## Descripción de los flujos de trabajo ingenieriles

El Modelado es el flujo de trabajo que tiene el objetivo de entender el negocio de la organización, el problema que se aborda en el proyecto y determinar una solución viable para resolver el problema de dominio. El flujo de trabajo Implementación tiene como objetivo transformar su(s) modelo(s) en código ejecutable y realizar un nivel básico de las pruebas. El flujo de trabajo de Prueba tiene como objetivo realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, verificando que se cumplan los requerimientos. Por último dentro de los flujos de trabajo ingenieriles se tiene el Despliegue, cuyo objetivo es el plan para la prestación del sistema y la ejecución de dicho plan, para que el sistema quede a disposición de los usuarios finales (AUP, 2015).

Debido a las características antes expuestas se decide utilizar la metodología AUP, como metodología de desarrollo, pues la misma es ideal para la guía del proceso de desarrollo de software sin añadir más trabajo que el desarrollo en sí, ya que opta por un paradigma de trabajo con entregables esenciales y específicos para el entendimiento de la solución final.

## **1.9 Conclusiones parciales**

Con el estudio realizado, de los SIG seleccionados en este capítulo, se llega a la conclusión de que ninguno responde a las necesidades del MINSAP, específicamente de las instituciones de sanidad, determinándose de esta forma que ninguno puede ser parte íntegra de la solución, pero que permitió identificar un conjunto de funcionalidades básicas que debe poseer la aplicación que se desea construir, por lo que se decide utilizar como base la plataforma GENESIG ya que proporciona una estructura a la cual se le pueden seguir añadiendo, quitando o modificando funcionalidades, en pos de realizar personalizaciones para un negocio específico que lo requiera, que es el caso del sistema que se quiere desarrollar. Durante el diseño y la implementación de la solución, las herramientas y tecnologías utilizadas serán las descritas en el presente capítulo.

### CAPÍTULO 2. PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

#### 2.1 Introducción

En este capítulo se realiza un análisis de la solución propuesta. Al no ser identificados claramente los procesos de negocio se plantea la conceptualización del entorno mediante un modelo de dominio. Se analizan y relacionan los conceptos y entidades que están presentes donde funcionará el sistema. Se identifican los requisitos funcionales y no funcionales con los que contará el sistema. Se analizan y detallan los casos de uso y se describen los actores. Además incluye el diseño de la solución propuesta, la arquitectura utilizada, los patrones de diseño que se usaron, los diagramas de clases del diseño, y el diseño de la base de datos.

#### 2.2 Modelo del dominio en el Desarrollo del Software.

El modelo de dominio no representa el sistema a desarrollar, permite un mayor entendimiento de la situación existente en un entorno determinado. *“La etapa orientada a objetos esencial del análisis o investigación es la descomposición de un dominio de interés en clases conceptuales individuales u objetos. El modelo de dominio describe conceptos del mundo real, no componentes software” (Larman, 2003).*

Se decide utilizar el modelo de dominio debido a que no se encuentran bien definidos los procesos necesarios para la realización de un SIG para la ubicación de entes de contaminación. Por otra parte no se cuenta con un cliente real, lo que imposibilita conocer, con un nivel amplio de detalle, los procedimientos existentes para el desarrollo de un sistema determinado.

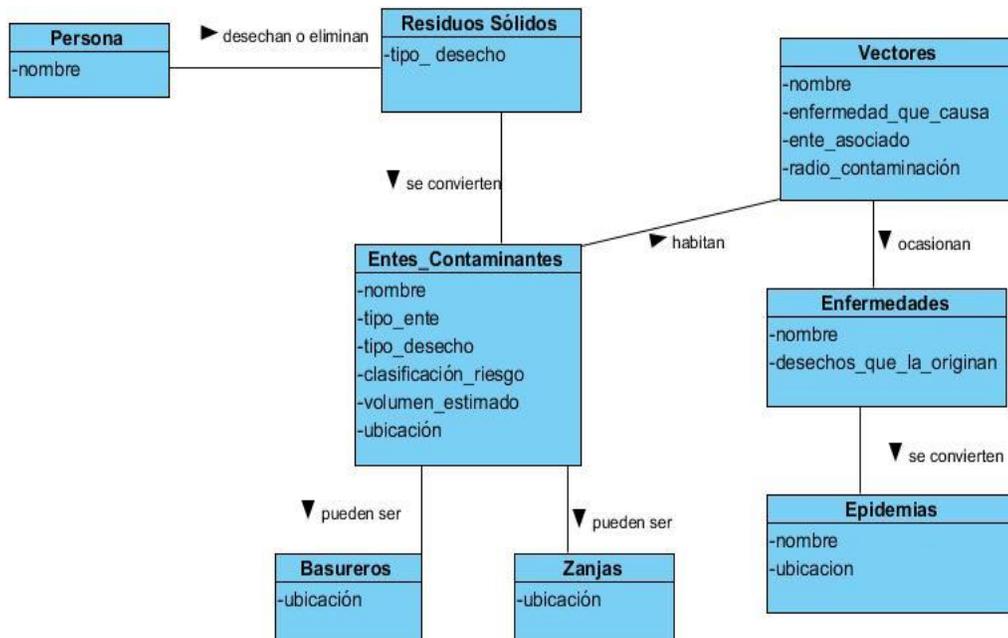


Fig. 7 Modelo del Dominio.

### 2.2.1. Descripción del Modelo del Dominio

Las personas o habitantes eliminan incontroladamente residuos sólidos sin pensar en los daños que pueden causar con su actitud. Estos desechos van acumulándose hasta que se convierten en entes contaminantes con grandes volúmenes de basuras. Estos entes se pueden clasificar en basureros o zanjas, en los que habitan innumerables vectores con el riesgo de ocasionar enfermedades, estas a su vez se pueden convertir en epidemias ya que los vectores pueden transitar e infectar a varias personas en una zona determinada.

### 2.2.2. Glosario de Términos del Modelo del Dominio

**Persona:** Habitante de una población que elimina incontroladamente residuos sólidos.

**Residuos Sólidos:** Material que sea desechado por la población; pueden ser de origen doméstico, comercial, industrial, desechos de la vía pública y resultantes de la construcción. Estos residuos son los que mayor dificultad tienen en su tratamiento, debido a la diversidad de los mismos y por ser generados en los domicilios.

**Entes Contaminantes:** Son las acumulaciones de desperdicios humanos y escombros que pueden estar constituidas por uno o varios depósitos localizados dentro de los límites de una zona del mismo origen y con características similares. Pueden clasificarse en basureros y zanjas.

**Basureros:** Sitio en donde se arroja y amontona la basura.

**Zanjas:** Excavación larga y estrecha hecha en la tierra, que puede contener agua, en donde se amontona la basura que desecha la población.

**Enfermedades:** Es una descripción en la salud comunitaria que ocurre cuando un padecimiento afecta a un número de individuos superior al esperado en una población durante un tiempo determinado.

**Epidemias:** Extensión de una enfermedad infecciosa a gran escala, y que afecta a la mayoría de los individuos de la zona en la que se desarrolla. En otras palabras, es un incremento significativamente elevado en el número de casos de una enfermedad con respecto al número de casos esperados.

**Vectores:** Es un agente que puede propagar o transmitir una enfermedad de un organismo a otro.

### 2.3 Modelo del Sistema

Luego de haber relacionado las clases que conforman el dominio del sistema, así como la descripción de los principales conceptos que utilizan los usuarios y con los que debe trabajar la aplicación a través del modelo de dominio, se representan los principales artefactos del flujo de trabajo Modelado, que tiene como objetivo principal guiar el desarrollo hacia el sistema correcto que cumpla con las necesidades planteadas por el cliente.

#### 2.3.1 Requisitos

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir (PRESSMAN, 2005). Definen lo que el sistema tiene que hacer y los servicios que debe proporcionar al usuario. Para que un requisito se considere satisfactorio debe cumplir las siguientes características: implementación independiente, consistente y no ambiguo, preciso, verificable, que pueda ser leído y modificable. Existen dos grandes categorías en las que pueden clasificarse los requisitos, estas son: requisitos funcionales y requisitos no funcionales.

#### 2.3.2 Estrategia de captura de requisitos

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto. El proceso de captura se basa en averiguar, normalmente en circunstancias difíciles, lo que se debe construir.

Para la captura de requisitos se utilizó como método de apoyo *la entrevista*. Esta estrategia es una de las técnicas más usadas en la captura de requisitos. A través del contacto directo con ambas partes (MINSAP y el equipo de desarrollo) posibilita de manera eficiente reunir toda la información relevante para la realización del software, y de esta forma se garantiza que la solución se adecue a las necesidades del cliente. Por lo que se puede decir que con la utilización de esta técnica se logra mejorar

la calidad del software pues si se cumple con todos los requisitos el software posee lo que el cliente desea por lo tanto tendrá buena calidad.

### 2.3.3 Listado de requisitos funcionales

A continuación se detallan los requisitos funcionales que la solución propuesta debe cumplir como respuesta al problema identificado.

#### Módulo de autenticación.

**RF 1. Autenticar usuario.** El sistema debe permitir que un usuario se autentique. La misma se puede ejecutar por los siguientes criterios de entrada:

- Usuario (Formato: Alfanumérico, Obligatorio: Sí).
- Contraseña (Formato: Alfanumérico, Obligatorio: Sí).

**RF 2. Cerrar Sesión.** Esta funcionalidad le permite al usuario terminar su sesión en la aplicación.

#### Módulo administración.

**RF 3. Adicionar usuario.** El sistema debe permitir adicionar nuevos usuarios que puedan acceder a la aplicación. Para esta funcionalidad se requieren los siguientes criterios de entrada:

- Nombre (Formato: Alfanumérico, Obligatorio: Sí).
- Usuario (Formato: Alfanumérico, Obligatorio: Sí).
- Contraseña (Formato: Alfanumérico, Obligatorio: Sí).
- Confirmar contraseña (Formato: Alfanumérico, Obligatorio: Sí).
- Rol (Formato: Lista desplegable, Obligatorio: Sí).

Existirán dos roles en el sistema que podrán realizar acciones en dependencia de sus permisos:

- Consultor: Podrá realizar todas las funcionalidades correspondientes a los módulos de autenticación, navegación y reporte y del módulo de administración tendrá solamente acceso a editar su perfil.
- Especialista: Podrá realizar todas las acciones que realiza el consultor y además tendrá acceso al módulo de administración, gestión y análisis.

El sistema mostrará un mensaje indicando que el usuario fue adicionado:

- Mensaje (Formato: Alfabético).

**RF 4. Modificar usuario.** El sistema debe permitir modificar los datos de los usuarios que tienen acceso a la aplicación. Para esta funcionalidad se requieren los siguientes criterios de entrada:

## Capítulo 2. Presentación de la solución propuesta

---

- Nombre (Formato: Alfanumérico, Obligatorio: Sí).
- Usuario (Formato: Alfanumérico, Obligatorio: Sí).
- Contraseña (Formato: Alfanumérico, Obligatorio: Sí).
- Confirmar contraseña (Formato: Alfanumérico, Obligatorio: Sí).
- Rol (Formato: Lista desplegable, Obligatorio: Sí).

El sistema mostrará un mensaje indicando que el usuario fue modificado:

- Mensaje (Formato: Alfabético).

**RF 5. Eliminar usuario.** El sistema debe permitir eliminar los usuarios que tienen acceso a la aplicación.

Para esta funcionalidad se requieren los siguientes criterios de entrada:

- Usuario (Formato: Alfanumérico, Obligatorio: Sí).

El sistema mostrará un mensaje confirmando que el usuario fue eliminado:

- Mensaje (Formato: Alfabético).

**RF 6. Editar perfil.** El sistema debe permitir modificar los datos del usuario que está autenticado en la aplicación. Para esta funcionalidad se requieren los siguientes criterios de entrada:

- Nombre (Formato: Alfanumérico, Obligatorio: Sí).
- Contraseña (Formato: Alfanumérico, Obligatorio: Sí).
- Confirmar contraseña (Formato: Alfanumérico, Obligatorio: Sí).

El sistema mostrará un mensaje indicando que el usuario fue modificado:

- Mensaje (Formato: Alfabético).

### **Módulo de navegación.**

**RF 7. Acercar determinada región del mapa.** Esta funcionalidad permite aumentar el tamaño del mapa y disminuye la escala, ubicando en el centro del mapa el punto donde el usuario realizó la operación de acercar. Para realizar esta funcionalidad se requiere que el usuario seleccione en el mapa el punto o la región que desea acercar.

**RF 8. Alejar determinada región del mapa.** Esta funcionalidad permite disminuir el tamaño del mapa en caso que el usuario lo haya aumentado con anterioridad. Una vez disminuido el tamaño se aumenta la escala del mapa. Para realizar esta funcionalidad se requiere que el usuario seleccione en el mapa el punto que desea alejar. El sistema mostrará dicha región a un menor nivel de visualización.

**RF 9. Centrar el mapa.** Esta funcionalidad permite ubicar en el centro del mapa una región previamente seleccionada al dar clic en el mismo, sin modificar su escala. Para realizar esta funcionalidad el usuario debe seleccionar la opción correspondiente. El sistema mostrará el mapa en su escala máxima.

## *Capítulo 2. Presentación de la solución propuesta*

---

**RF 10. Mover el mapa.** Esta funcionalidad permite mover el mapa variando con el puntero del ratón la posición de la vista que se presenta, sin modificar su escala. Para realizar esta funcionalidad el usuario debe seleccionar la opción correspondiente. El sistema mostrará el mapa en su escala máxima.

### **Módulo de gestión.**

**RF 11. Adicionar información geográfica de Ente Contaminante.** Esta funcionalidad permite que el usuario pueda ubicar en el mapa un nuevo ente contaminante, así como adicionar la información asociada al mismo. Estos podrán ser: basureros o zanjas. Para ello deberá escoger la opción correspondiente y el sistema mostrará una ventana en la que se podrán introducir los datos del ente:

- Nombre (Formato: Alfanumérico, Obligatorio: Sí).
- Tipo de ente (Formato: Lista desplegable, Obligatorio: Sí).
- Tipo de desecho (Formato: Lista desplegable, Obligatorio: Sí).
- Clasificación de riesgo (Formato: Lista desplegable, Obligatorio: Sí).
- Volumen estimado (Formato: Numérico, Obligatorio: Sí).
- Ubicación Geográfica (Formato: Punto o región, Obligatorio: Sí).

**RF 12. Modificar información geográfica de Ente Contaminante.** Esta funcionalidad permite que el usuario pueda modificar los datos referentes a los entes, así como su ubicación geográfica. Para ello deberá escoger la opción correspondiente, seleccionar el ente que desea modificar y el sistema mostrará una ventana en la que se podrán cambiar los datos correspondientes:

- Nombre (Formato: Alfanumérico, Obligatorio: Sí).
- Tipo de ente (Formato: Lista desplegable, Obligatorio: Sí).
- Tipo de desecho (Formato: Lista desplegable, Obligatorio: Sí).
- Clasificación de riesgo (Formato: Lista desplegable, Obligatorio: Sí).
- Volumen estimado (Formato: Numérico, Obligatorio: Sí).
- Ubicación Geográfica (Formato: Punto o región, Obligatorio: Sí).

**RF 13. Eliminar información geográfica de Ente Contaminante.** Esta funcionalidad permite que el usuario pueda eliminar los entes que tiene representados en el mapa. Para ello deberá escoger la opción correspondiente y luego seleccionar el ente que desea eliminar. Una vez el usuario haya confirmado la eliminación, el sistema mostrará el mapa sin dicho ente.

**RF 14. Adicionar información de Enfermedad.** Esta funcionalidad permite que el usuario adicione una nueva enfermedad, así como la información asociada a la misma. Para ello deberá escoger la opción

## *Capítulo 2. Presentación de la solución propuesta*

---

correspondiente y el sistema mostrará una ventana en la que se podrán introducir los datos de la enfermedad:

- Nombre de enfermedad (Formato: Alfabético, Obligatorio: Sí).
- Desechos que la originan (Formato: Campo de selección, Obligatorio: Sí).

**RF 15. Modificar información de Enfermedad.** Esta funcionalidad permite que el usuario pueda modificar los datos referentes a las enfermedades. Para ello deberá escoger la opción correspondiente, seleccionar la enfermedad que desea modificar y el sistema mostrará una ventana en la que se podrán introducir los datos correspondientes:

- Nombre de enfermedad (Formato: Alfabético, Obligatorio: Sí).
- Desechos que la originan (Formato: Campo de selección, Obligatorio: Sí).

**RF 16. Eliminar información de Enfermedad.** Esta funcionalidad permite que el usuario pueda eliminar las enfermedades. Para ello deberá seleccionar la enfermedad que desea eliminar y luego escoger la opción correspondiente. Una vez el usuario haya confirmado la eliminación, el sistema mostrará el listado de enfermedades sin dicha enfermedad.

**RF 17. Adicionar información geográfica de Epidemia.** Esta funcionalidad permite que el usuario pueda ubicar en el mapa una nueva epidemia. Para ello deberá escoger la opción correspondiente y el sistema mostrará una ventana donde se podrán introducir los datos de la epidemia:

- Nombre de la epidemia (Formato: Alfabético, Obligatorio: Sí).
- Ubicación geográfica (Formato: Punto o región, Obligatorio: Sí).

**RF 18. Modificar información geográfica de Epidemia.** Esta funcionalidad permite que el usuario pueda modificar en el mapa una epidemia. Para ello deberá escoger la opción correspondiente y el sistema mostrará una ventana donde se podrán modificar los datos de la epidemia:

- Nombre de la epidemia (Formato: Alfabético, Obligatorio: Sí).
- Ubicación geográfica (Formato: Punto o región, Obligatorio: Sí).

**RF 19. Eliminar información geográfica de Epidemia.** Esta funcionalidad permite que el usuario pueda eliminar las epidemias que tiene representadas en el mapa. Para ello deberá escoger la opción correspondiente y luego seleccionar la epidemia que desea eliminar. Una vez el usuario haya confirmado la eliminación, el sistema mostrará el mapa sin dicha epidemia.

## *Capítulo 2. Presentación de la solución propuesta*

---

**RF 20. Adicionar información de Vector.** Esta funcionalidad permite que el usuario pueda adicionar nuevos vectores contaminantes. Para ello deberá escoger la opción correspondiente y el sistema mostrará una ventana donde se podrán introducir los datos de los vectores:

- Nombre del vector contaminante (Formato: Alfabético, Obligatorio: Sí).
- Enfermedad transmitida (Formato: Lista Despegable, Obligatorio: Sí).
- Ente Contaminante asociado (Formato: Lista Despegable, Obligatorio: Sí).
- Radio de contaminación (Formato: Numérico, Obligatorio: Sí).

**RF 21. Modificar información de Vector.** Esta funcionalidad permite que el usuario pueda modificar los vectores ya existentes. Para ello deberá escoger la opción correspondiente y el sistema mostrará una ventana donde se podrán modificar los datos de los vectores:

- Nombre del vector contaminante (Formato: Alfabético, Obligatorio: Sí).
- Enfermedad transmitida (Formato: Lista Despegable, Obligatorio: Sí).
- Ente Contaminante asociado (Formato: Lista Despegable, Obligatorio: Sí).
- Radio de contaminación (Formato: Numérico, Obligatorio: Sí).

**RF 22. Eliminar información de Vector.** Esta funcionalidad permite que el usuario pueda eliminar los vectores contaminantes. Para ello deberá seleccionar el vector que desea eliminar y luego escoger la opción correspondiente. Una vez el usuario haya confirmado la eliminación, el sistema mostrará el listado de vectores sin dicho vector.

**RF 23. Ubicar Elemento en el mapa.** Esta funcionalidad permite que el usuario pueda localizar un elemento ya sea un ente contaminante o una epidemia mediante la selección de una región a partir del trazado con el ratón sobre el mapa. Para ello el usuario deberá seleccionar la opción correspondiente y luego el elemento que desea ubicar. El sistema mostrará el elemento ubicado en el mapa.

### **Módulo de análisis.**

**RF 24. Realizar tematización.** Esta funcionalidad permite que el usuario pueda crear un mapa temático en función del criterio(s) de análisis que haya seleccionado. Estos criterios son:

- Riesgo de Enfermedad.
- Riesgo Biológico.
- Sin Riesgo.
- Peligro Químico.
- Riesgo Radioactivo.

Las tematizaciones, serán mostradas en la mayor escala del mapa en el momento de realizar la misma. Una vez definidos los criterios, estos se mostrarán con un ícono diferente para cada uno de ellos en el mapa, contiene una leyenda donde el usuario podrá identificar cada riesgo que está representado en el mapa, además el usuario podrá modificar la cantidad de criterios que quiere mostrar en caso que lo desee. El sistema mostrará el mapa tematizado por íconos en dependencia de los criterios que haya seleccionado el usuario.

### **Módulo de reportes.**

**RF 25. Mostrar la cantidad de basureros por un área determinada.** Esta funcionalidad permite obtener un reporte donde se muestre un listado con todos los basureros asociados a un área determinada indicada por el usuario, además se muestra el riesgo de enfermedad asociada a dicha área determinado por un radio de acción que abarca al ente. El usuario deberá seleccionar la opción correspondiente y el sistema mostrará el reporte.

**RF 26. Ver ubicación en el mapa.** Esta funcionalidad permite que el usuario pueda ver la ubicación del ente contaminante seleccionado en el mapa. Para ello el usuario deberá seleccionar el ente que desea ver la ubicación y luego la opción correspondiente. El sistema mostrará la región que comprende al ente seleccionado a menor escala en el mapa.

### **2.3.4 Requisitos No Funcionales**

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. (PRESSMAN, 2005).

Los requisitos no funcionales pueden ser más críticos que los funcionales, puesto que si un requisito funcional no se cumple, el sistema se degrada, pierde eficacia, y puede no responder a la totalidad de los requisitos del usuario, pero en cambio si un requisito no funcional no se cumple, el sistema puede inutilizarse.

#### **RNF 1. Usabilidad**

El sistema debe permitir un alto nivel de facilidades de uso, basado en el cumplimiento de los siguientes aspectos:

- ✓ El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.
- ✓ Las funcionalidades principales del sistema estarán orientadas a íconos para un mayor reconocimiento por parte del usuario.

### **RNF 2. Confiabilidad**

- ✓ La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.

### **RNF 3. Restricciones de diseño**

- ✓ El diseño deberá ser sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema.
- ✓ El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.
- ✓ El producto se desarrollará sobre la Plataforma GeneSIG que constituye una plataforma de desarrollo para ser personalizada como aplicaciones a la medida, pudiéndose incorporar a ésta nuevas funcionalidades.

### **RNF 4. Interfaz**

#### **Interfaces de usuario**

- ✓ El sistema debe poseer una interfaz gráfica uniforme incluyendo pantallas, menús y opciones.
- ✓ Tanto los títulos de los componentes de la interfaz, como los mensajes para interactuar con los usuarios, deberán ser en idioma español y tener una apariencia uniforme guiada por los colores negro y azul claro.
- ✓ Los mensajes definidos deberán ser lo suficientemente informativos para dar a conocer la severidad de los mismos.
- ✓ Los colores de la región de los entes contaminantes se podrán apreciar en rojo. Las regiones identificadas como epidemias se representarán en verde.

#### **Interfaces hardware**

Para las PCs clientes:

- ✓ Se requiere tengan tarjeta de red.
- ✓ Al menos 128 MB de memoria RAM.
- ✓ Procesador 512 MHz como mínimo.

Para los servidores:

- ✓ Se requiere tarjeta de red.
- ✓ El Servidor de Mapas debe tener como mínimo 2 GB de RAM y 40 GB de disco duro.
- ✓ El Servidor de base de datos debe tener como mínimo 2 GB de RAM y 40 GB de disco duro.
- ✓ Procesador de 3 GHz como mínimo.

#### **Interfaces software**

La construcción de la aplicación funcionará bajo los conceptos de la arquitectura cliente/servidor. Por tanto se deben tener como requerimientos mínimos de software:

Para las PCs clientes:

- ✓ Un navegador como Mozilla Firefox, Safari u otro navegador.

Para los Servidores:

- ✓ Sistema operativo GNU/Linux Ubuntu Server 12.04.
- ✓ Servidor Web Apache 2.2 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- ✓ PostgreSQL 9.1 como Sistema Gestor de Base de Datos.
- ✓ PostGIS 1.5. como extensión de PostgreSQL como soporte de datos espaciales.
- ✓ MapServer 6.0 con extensión PHP mapscript.

### Interfaces de comunicación

- ✓ El producto garantizará mediante su interfaz la configuración del entorno de trabajo mediante funcionalidades propias como ocultar y mostrar paneles, así como elementos para cambiar las vistas, las escalas y las capas que serán visibles en la interacción.

### RNF 5. Requisitos de licencia

De acuerdo a los tipos de licencias de los componentes y herramientas que propone utilizar para el desarrollo del producto se puede catalogar legalmente esta arquitectura de modelo libre, permitiendo la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la autorización de sus respectivos titulares.

## 2.4 Definición de casos de uso del sistema

### 2.4.1 Descripción de los actores

Los actores se definen como los roles que puede tener un usuario; pueden ser humanos, otros sistemas, máquinas, hardware, etc. que interactúan con un sistema, para de esta forma intercambiar datos. Los mismo juegan un papel muy importante en los casos de uso, a continuación se describen los actores que posee el sistema propuesto.

**Tabla 1. Descripción de los actores del sistema.**

<b>Consultor</b>	Usuario que podrá realizar todas las funcionalidades correspondientes a los módulos de autenticación, navegación, reportes y del módulo de administración tendrá acceso solamente a editar su perfil.
------------------	---

<b>Especialista</b>	Usuario que podrá realizar todas las acciones que realiza el consultor y además tendrá acceso al módulo de administración, gestión y análisis.
---------------------	--

## 2.4.2 Diagrama de Casos de uso del sistema

El diagrama de CUS es el modelo de las funciones deseadas para el sistema y su entorno, que sirve como contrato entre el cliente y los desarrolladores. Estas funciones por lo general agrupan uno o más requisitos funcionales definidos en el flujo de trabajo de Modelado, como lo propone la metodología AUP. Una vez recopilados los requisitos funcionales del sistema, se muestra el diagrama de caso de uso del sistema, en el cual se representan los actores y su relación con el sistema:

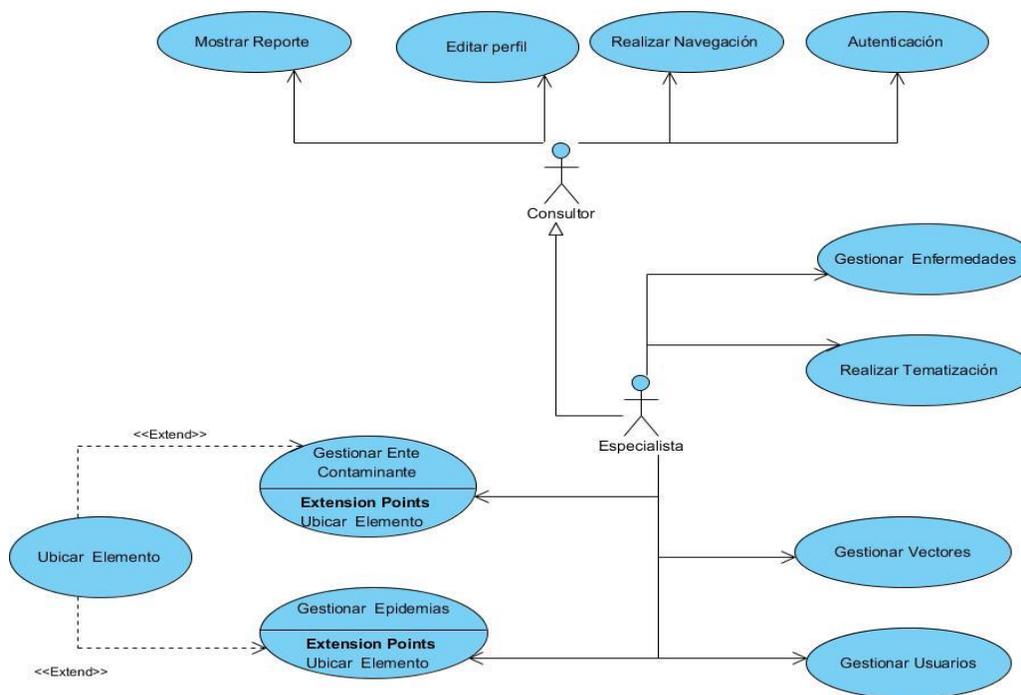


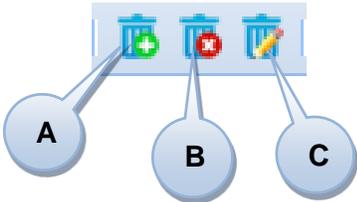
Fig. 8 Diagrama de Casos de Uso del Sistema.

## 2.4.3 Descripción textual de los casos de uso del sistema.

Teniendo en cuenta que los casos de uso son los que dirigen todo el proceso de desarrollo y debido a que la mayoría de las actividades como el análisis, diseño y prueba se llevan a cabo partiendo de los casos de uso, se hace necesario realizar una descripción de los mismos. Debido a que es muy amplia la información, se muestra a continuación la descripción del caso de uso más crítico del sistema, los demás pueden ser consultados en el expediente de proyecto.

## *Capítulo 2. Presentación de la solución propuesta*

**Tabla 2. Descripción del Caso de Uso “Gestionar Ente Contaminante”.**

<b>Caso de Uso:</b>	Gestionar Ente Contaminante.	
<b>Actores:</b>	Especialista	
<b>Propósito:</b>	Este caso de uso se lleva a cabo con el objetivo de insertar nuevos entes contaminantes, así como editar o eliminar los ya existentes.	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción correspondiente y termina cuando el sistema guarda los datos correspondientes a dicha acción.	
<b>Precondiciones:</b>	El especialista tiene que estar autenticado en la aplicación.	
<b>Referencias:</b>	RF 11, RF 12, RF 13.	
<b>Prioridad:</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El especialista selecciona la opción para Insertar, Eliminar o Listar los entes contaminantes.	<p>2. El sistema muestra un menú con las acciones que el usuario puede realizar. Ver Interfaz 13.</p> <ul style="list-style-type: none"> <li>- Si selecciona “Insertar ente contaminante” (A), ver sección “Insertar ente contaminante”.</li> <li>- Si selecciona “Eliminar ente contaminante” (B), ver sección “Eliminar ente contaminante”.</li> <li>- Si selecciona “Listar entes contaminantes” (C), ver sección “Listar entes contaminantes”.</li> </ul>	
	3. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el especialista y guarda los datos.	
<b>Prototipo de Interfaz</b>		
<p><b>Interfaz 13</b></p> 		

## Capítulo 2. Presentación de la solución propuesta

Sección "Insertar Ente contaminante"	
Acción del Actor	Respuesta del Sistema
1. El especialista selecciona la opción "Insertar ente contaminante" (A). Ver Interfaz 13.	2. El sistema muestra la ventana para que se introduzcan los datos del nuevo ente contaminante. Ver Interfaz 14.
3. El especialista introduce los datos correspondientes y presiona el botón "Ubicar en el mapa". (E). Ver interfaz 14.	4. El sistema esconde la ventana "Insertar ente contaminante" para que el especialista seleccione en el mapa el ente que desea ubicar. Ver interfaz 15.
5. El especialista marca en el mapa la región donde desea ubicar el ente contaminante.	6. El sistema muestra un mensaje para guardar los cambios realizados. Ver interfaz 16.
7. El especialista presiona el botón "Aceptar". (F). Ver interfaz 17.	8. El sistema procesa la información y muestra un mensaje indicando que el ente fue adicionado. Ver interfaz 18.

### Prototipo de interfaz

#### Interfaz 14

**Insertar ente contaminante**

Nombre:

Tipo de ente: --Seleccionar--

Tipo de desecho: --Seleccionar--

Clasificación de Riesgo: --Seleccionar--

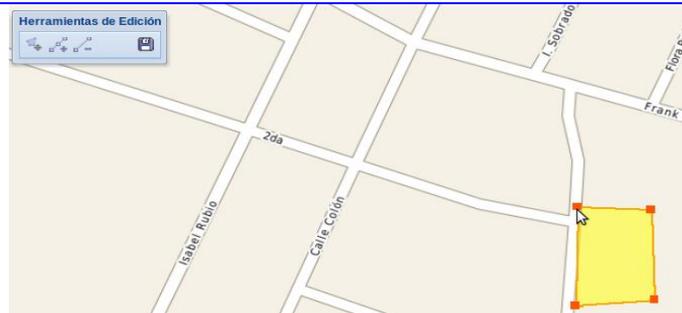
Volumen (m<sup>3</sup>): 0

Información geográfica:

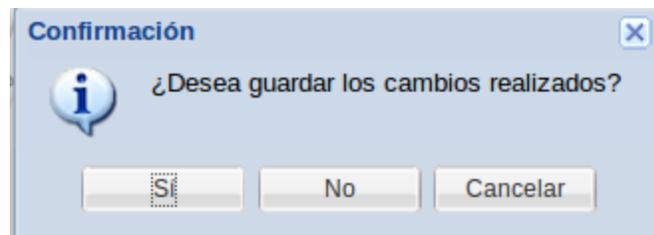
**E**

#### Interfaz 15

## Capítulo 2. Presentación de la solución propuesta



**Interfaz 16**



**Interfaz 17**

**Insertar ente contaminante**

Nombre:

Tipo de ente:

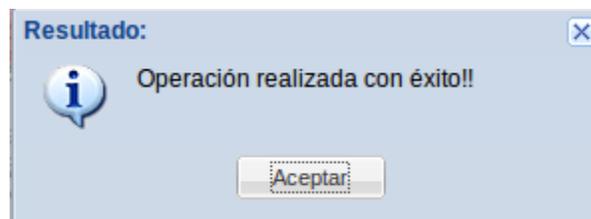
Tipo de desecho:

Clasificación de Riesgo:

Volumen (m<sup>3</sup>):

Información geográfica:

**Interfaz 18**



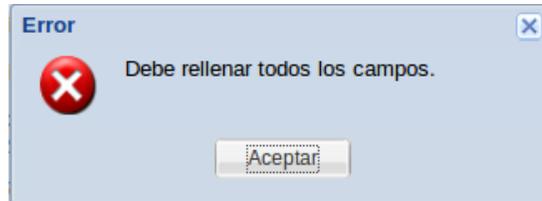
**Flujos Alternos**

## *Capítulo 2. Presentación de la solución propuesta*

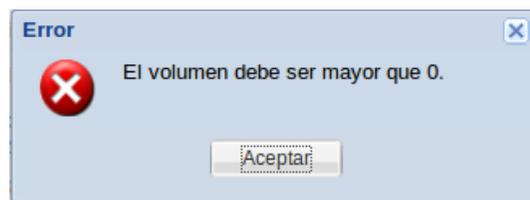
Acción del Actor	Respuesta del Sistema
3. El especialista deja campos en blanco.	4. El sistema muestra un mensaje indicando que hay campos con valores inválidos para que llene los campos que están en blanco. Ver interfaz 19.
3. El especialista no introduce un valor para el volumen mayor que 0.	4. El sistema muestra un mensaje indicando que el volumen debe ser mayor que 0. Ver interfaz 20.
3. El especialista presiona el botón “Cancelar” (D). Ver interfaz 17.	4. El sistema oculta la ventana “Insertar ente contaminante”.
5. El especialista no selecciona en el mapa un área válida para ubicar el ente contaminante.	6. El sistema muestra un mensaje indicando que debe seleccionar un área válida.

### **Prototipo de interfaz**

#### **Interfaz 19**



#### **Interfaz 20**



### **Sección “Eliminar Ente”**

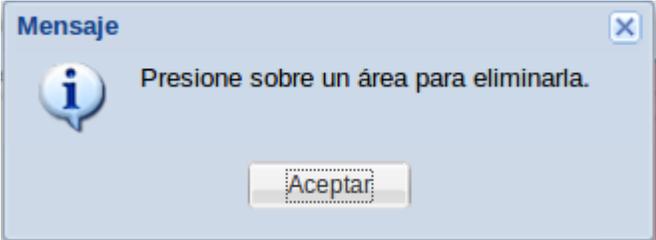
Acción del Actor	Respuesta del Sistema
1. El especialista selecciona la opción “Eliminar Ente Contaminante” (C). Ver interfaz 13.	2. El sistema muestra un mensaje indicando que debe seleccionar un área para eliminarla. Ver interfaz 21.
3. El especialista selecciona el área del ente que desea eliminar.	4. El sistema muestra un mensaje indicando que el ente fue eliminado.

## Capítulo 2. Presentación de la solución propuesta

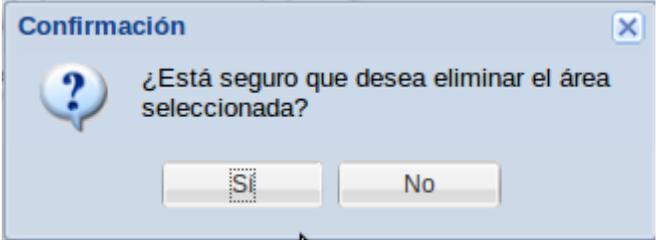
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El especialista no selecciona el ente.	4. El sistema muestra un mensaje indicando que debe seleccionar un área válida. Ver interfaz 21.
5. El especialista cierra el mensaje de información.	6. El sistema oculta el mensaje y muestra el mapa para que el especialista seleccione el ente que desea eliminar.
7. El especialista selecciona el ente que desea eliminar.	8. El sistema muestra un mensaje de confirmación para que el especialista confirme que es el área correcta a eliminar. Ver interfaz 22.
9. El especialista confirma el área a eliminar y cierra el mensaje.	10. El sistema cierra el mensaje de confirmación y elimina el ente seleccionado y muestra un mensaje indicando que el ente fue eliminado. Ver interfaz 23.

**Prototipo de interfaz**

**Interfaz 21**

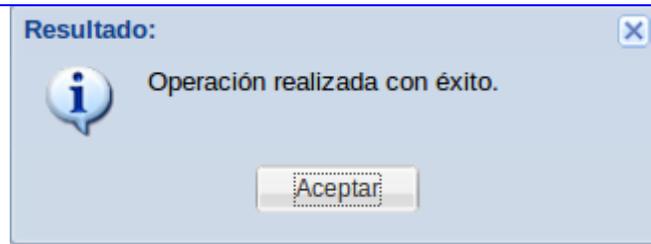


**Interfaz 22**



**Interfaz 23**

## Capítulo 2. Presentación de la solución propuesta



### Sección "Listar Entes contaminantes"

#### Acción del Actor

1. El especialista selecciona la opción "Listar entes contaminantes" (C). Ver interfaz 13.

#### Respuesta del Sistema

2. El sistema muestra la ventana "Listado de entes contaminantes". Ver interfaz 24.

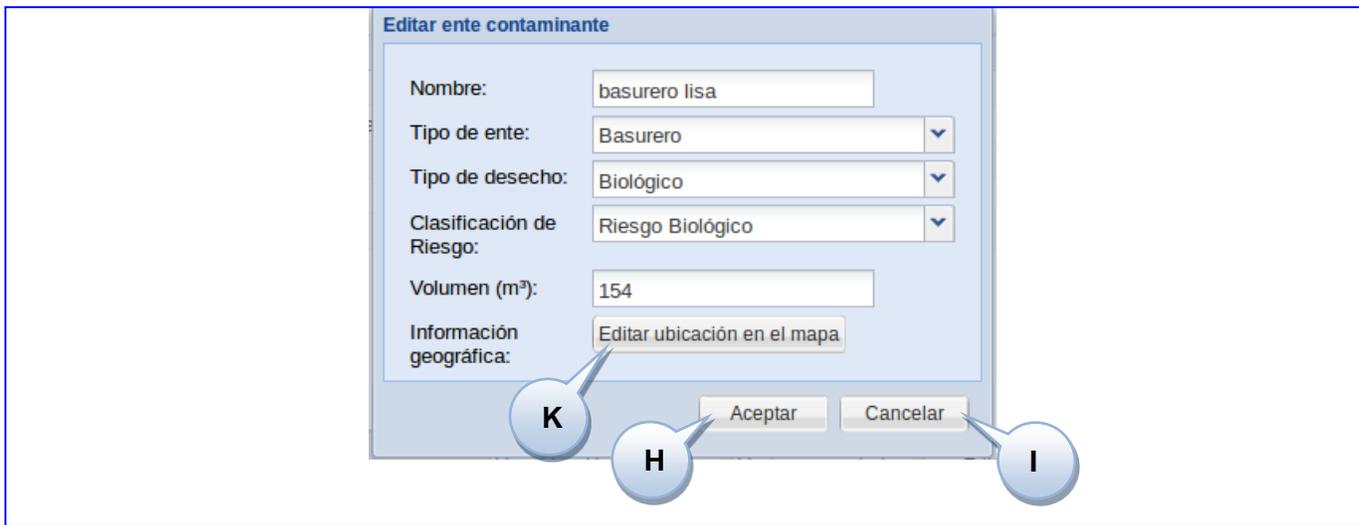
### Prototipo de interfaz

#### Interfaz 24



#### Interfaz 25

## *Capítulo 2. Presentación de la solución propuesta*



### Flujos Alternos

Acción del Actor	Respuesta del Sistema
3. El especialista selecciona el ente y presiona el botón "Editar" (L). Ver interfaz 24.	4. El sistema muestra la ventana correspondiente para que el usuario modifique los datos. Ver interfaz 25.
5. El especialista modifica los datos y presiona el botón "Editar ubicación en el mapa" (K).	6. El sistema oculta la ventana "Editar ente contaminante", dejando visible solamente la parte del mapa.
7. El especialista edita el ente contaminante seleccionado a partir de las herramientas de edición. Ver interfaz 26.	8. El sistema muestra nuevamente la ventana "Editar ente contaminante".
9. El especialista presiona el botón "Aceptar" (H). Ver interfaz 25.	10. El sistema muestra un mensaje indicando que el ente fue modificado.
3. El especialista selecciona el ente y presiona el botón "Ver ubicación en el mapa" (M). Ver interfaz 24.	4. El sistema muestra el ente seleccionado a menor escala en el mapa.
3. El especialista selecciona el ente y presiona el botón "Vectores asociados" (N). Ver interfaz 24.	4. El sistema muestra los vectores que están asociados al ente seleccionado.

### **Prototipo de interfaz**

### **Interfaz 26**

	
<b>Poscondiciones:</b>	El sistema actualiza el registro de entes contaminantes del sistema según las acciones del especialista.

### 2.4.4 Patrones de Casos de Uso

Los Patrones de Caso de Uso se presentan a modo de herramientas que permiten resolver los problemas que se les planteen a los desarrolladores de una forma ágil y sistemática describiendo como deberían ser estructurados y organizados los casos de uso, capturando las mejores prácticas para modelarlos. Estos patrones se enfocan hacia el diseño y las técnicas utilizadas en modelos de alta calidad, y no en cómo modelar usos específicos (PRESSMAN, 2005).

#### Patrones de casos de uso empleados

Patrón CRUD Completo: Este patrón consta de un caso de uso, llamado Información CRUD o Gestionar información modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.

Patrón Concordancia adición: En el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema. En la figura 8 correspondiente al Diagrama de casos de uso del sistema, se observa este patrón en la relación de los casos de uso “Gestionar Ente Contaminante” y “Ubicar Elemento”; “Gestionar Epidemias” y “Ubicar Elemento”.

Patrón Múltiples actores Rol común: Puede suceder que los dos actores jueguen el mismo rol sobre el CU. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad

externa interactuando con cada una de las instancias del caso de uso. En la figura 8 correspondiente al Diagrama de casos de uso del sistema, se observa este patrón en la relación de los actores “Consultor” y “Especialista”.

### **2.5 Arquitectura de Software**

La arquitectura de software de un sistema de programa o computación es la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente y las relaciones entre ellos (PRESSMAN, 2005). La misma está compuesta por los artefactos más significativos, permitiendo establecer un esquema de cómo va a quedar constituido el software. Sobre la arquitectura inicial se van agregando nuevos artefactos propiciando así una arquitectura más robusta, describiéndose en la misma los principales aspectos tomados en cuenta para la construcción del software de forma progresiva.

La Plataforma GeneSIG emplea la arquitectura Cliente-Servidor, Orientada a objeto y basada en componentes.

Arquitectura Cliente-Servidor: Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes. En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores estos últimos responden a la demanda del cliente que la produjo. Los clientes y los servidores pueden estar conectados a una red local o una red amplia. Bajo este modelo 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 como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura (Scribd, 2015).

Arquitectura Orientada a Objetos: Nombres alternativos para este estilo han sido Arquitecturas Basadas en Objetos, Abstracción de Datos y Organización Orientada a Objetos. Los componentes del estilo se basan en principios orientados a objetos: encapsulamiento, herencia y polimorfismo. Las interfaces están separadas de las implementaciones. Las representaciones de los datos y las operaciones están encapsuladas en un tipo abstracto de datos u objeto (Kicillof, y otros, 2004).

Arquitectura Basada en Componentes: se centra en el diseño y construcción de sistemas computacionales que utilizan componentes de software reutilizables. Este proceso de construcción de

una pieza de software con componentes ya existentes, da origen al principio de reutilización del software, mediante el cual se promueve que los componentes sean implementados de una forma que permita su utilización funcional sobre diferentes sistemas en el futuro. La reusabilidad es una de las características más importantes en el desarrollo de sistemas bajo una arquitectura basada en componentes. Un componente de software debe ser diseñado de tal manera que pueda ser reutilizado en otros sistemas (Larman, 2003).

### **2.6 Patrones**

Un patrón es una descripción de un problema bien conocido que suele incluir: descripción, escenario de uso, solución concreta, consecuencias de utilizar el patrón, ejemplos de implementación y lista de patrones relacionados (Craig Larman, 1999).

En el mundo del software los patrones implementan soluciones a problemas frecuentes, brindando un vocabulario de entendimiento común entre los desarrolladores (Patrones, 2001).

A continuación se mencionan varios patrones de diseño y arquitectónicos que han sido utilizados durante el desarrollo de la solución.

#### **2.6.1 Patrones de arquitectura**

Un patrón se define como una solución probada con éxito que aparece una y otra vez ante determinado tipo de problema en un contexto dado. Los patrones se definen por un nombre, un problema, una solución y las consecuencias de su aplicación. Éste define una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones (PRESSMAN, 2005).

#### **Patrones arquitectónicos empleados**

Patrón de arquitectura orientada a objetos: El patrón de arquitectura orientada a objetos define el sistema como un conjunto de objetos que cooperan entre sí en lugar de un conjunto de procedimientos. Los componentes del estilo se basan en principios orientados a objetos: encapsulamiento, herencia y polimorfismo. Son las unidades de modelado, diseño e implementación, y los objetos y sus interacciones son el centro de las incumbencias en el diseño de la arquitectura y en la estructura de la aplicación. Las interfaces están separadas de las implementaciones y en cuanto a las restricciones, puede admitirse o no que una interfaz pueda ser implementada por múltiples clases.

Patrón de arquitectura basado en componentes: Las características principales de este patrón son la modularidad, la reusabilidad y la compatibilidad. El mismo describe un acercamiento al diseño de sistemas como un conjunto de componentes que exponen interfaces bien definidas y que colaboran entre

sí para resolver problemas. Los componentes son diseñados de forma que puedan ser reutilizados en distintos escenarios en disímiles aplicaciones, aunque algunos componentes son realizados para una tarea específica.

Luego de haber analizado la arquitectura de la plataforma GeneSIG y haber decidido mantener la misma para el sistema a desarrollar, se hace necesario entonces definir los patrones de diseño que se emplean en la construcción del SIG.

### **2.6.2 Patrones de Diseño**

Un patrón de diseño es una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular (Prieto, 2008).

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interfaces, o sea, un patrón de diseño es una solución a un problema de diseño. Expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software. Éstos facilitan la reutilización de arquitecturas y diseños de software. Un patrón de diseño identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades. De forma general, se clasifican en dos grupos: Patrones de Principios Generales para Asignar Responsabilidad (GRASP: General ResponsibilityAssignment Software Patterns, por sus siglas en inglés) y Patrones de Diseño Pandilla de los Cuatro o Gound-of-Four (GoF) (Craig Larman, 1999). Para el desarrollo del producto, se hace uso de patrones pertenecientes a ambos grupos. Seguidamente, se argumenta al respecto.

#### **Patrones GoF empleados:**

Command (Acción): Este patrón se utiliza en el proceso de petición de una información cualquiera al sistema por un cliente, mediante la Interfaz Gráfica de Usuario (GUI); puesto que en el sistema propuesto el usuario interactúa constantemente con las GUI, por ende se hace uso de este patrón de diseño, convirtiéndose esta característica en una de las más primordiales. Permite encapsular las peticiones a través de un objeto, lo que permite realizar operaciones como gestionar las acciones de dicho objeto. Este patrón es utilizado en la clase AjaxHelper, la cual se encarga de manejar las peticiones desde la interfaz de usuario y el servidor.

Singlenton (Solitario): Este patrón garantiza el acceso único a una clase mediante una única instancia. De esta forma se controla el acceso a las clases. Esta propiedad se evidencia en el framework CartoWeb con el empleo del objeto “mapa” el cual es único en toda la ejecución de la aplicación, pues se evita que no se cree cada vez que se hace un envío de datos en la aplicación.

### Patrones GRASP empleados:

Experto: Se emplea para asignar una responsabilidad a la clase que posee la información necesaria para cumplir con la responsabilidad (Craig Larman, 1999). Dentro del sistema se utiliza este patrón en clases que deben ser las encargadas de proporcionar los datos cargados de una fuente de datos así como modificarlos mismos.

Creador: Es usado para asignar a la clase B la responsabilidad de crear una instancia de clase (Craig Larman, 1999). Dentro del sistema se evidencia su uso en los casos de la instanciación de las clases dentro de la interfaz, los controladores y otros que contienen objetos como Map.

Bajo acoplamiento: Se utiliza para asignar las responsabilidades de modo que se mantenga bajo acoplamiento (Craig Larman, 1999). Este patrón se utiliza en la asignación de responsabilidades a clases de manera que un cambio en una de estas genere poco cambio en otras.

Alta cohesión: Se emplea para asignar las responsabilidades de modo que se mantenga una alta cohesión (Craig Larman, 1999). Se da una alta cohesión funcional cuando los elementos de un componente colaboran para producir algún comportamiento bien definido.

### 2.7 Modelo de Diseño

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación. En el modelo de diseño, los casos de uso son realizados por las clases de diseño y sus objetos (Ivar Jacobson, 2000).

#### 2.7.1 Diagrama de clases del diseño

Una clase del diseño es una abstracción de una clase o construcción similar en la implementación del sistema. El lenguaje que se utiliza en dichas clases es el mismo que se emplea para la implementación del sistema; se especifican los atributos y las operaciones; se pueden realizar interfaces si tienen sentido para la programación y los métodos tienen correspondencia con las operaciones que fueron utilizadas en la programación.

A continuación se presenta en la Figura 9 el Diagrama de Clases del diseño correspondiente al Caso de Uso Gestionar Ente Contaminante, los demás pueden ser consultados en el expediente de proyecto.

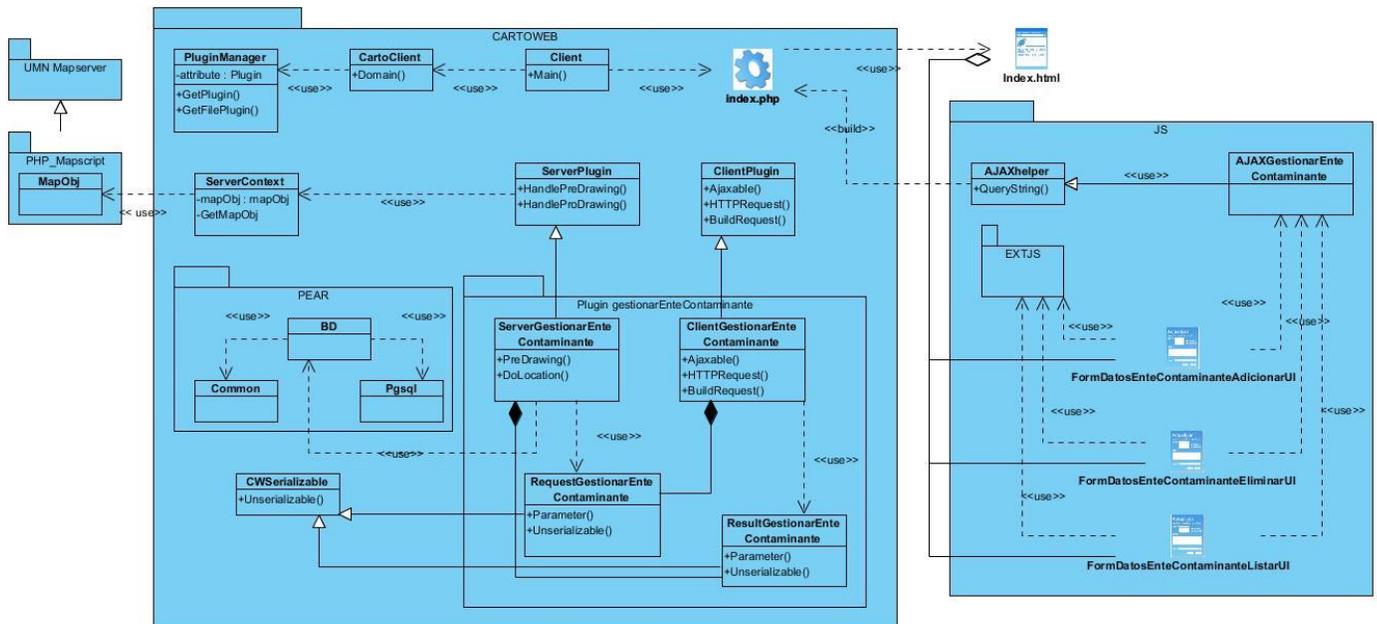


Fig. 9 Diagrama de Clases del Diseño del CU Gestionar Ente Contaminante.

## 2.7.2 Diseño de la Base de Datos

La base de datos necesita de una definición de su estructura, de manera que permita almacenar datos, reconocer el contenido y recuperar la información. Para diseñar una base de datos se necesita seguir un conjunto de pasos que comienzan con definir las clases persistentes, luego refinarlas y clasificarlas junto con sus atributos, para más tarde realizar el diagrama de entidad-relación (Falero, 2010).

Para la construcción del presente sistema se utilizó PostgreSQL como SGBD<sup>18</sup>. La base de datos fue diseñada utilizando la herramienta Visual Paradigm. A continuación se representa en la figura 10 el diagrama de clases persistentes y en la figura 11 el diagrama entidad relación de la base de datos.

<sup>18</sup> Sistema Gestor de Base de Datos.

## Capítulo 2. Presentación de la solución propuesta

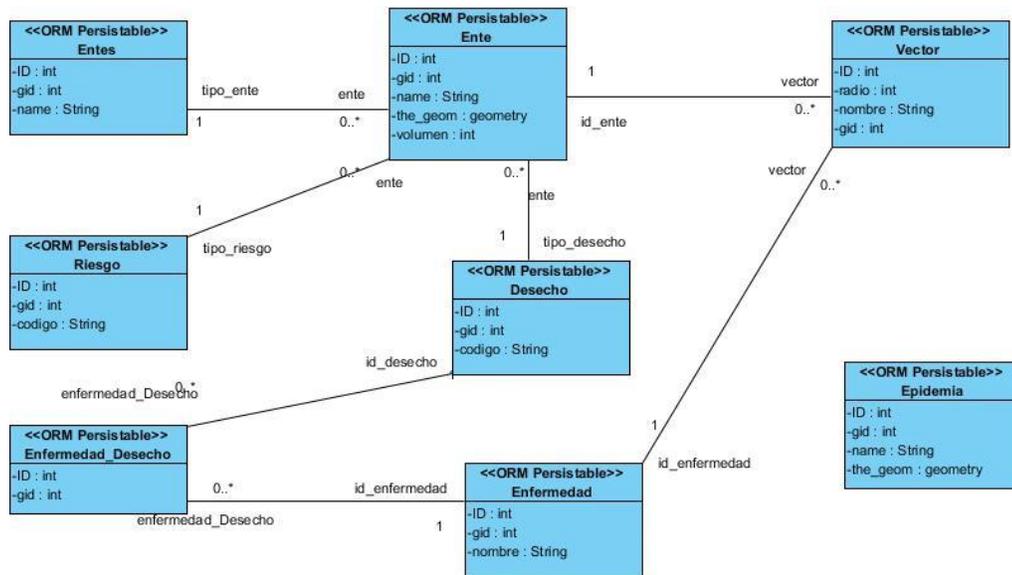


Fig. 10. Diagrama de Clases Persistentes

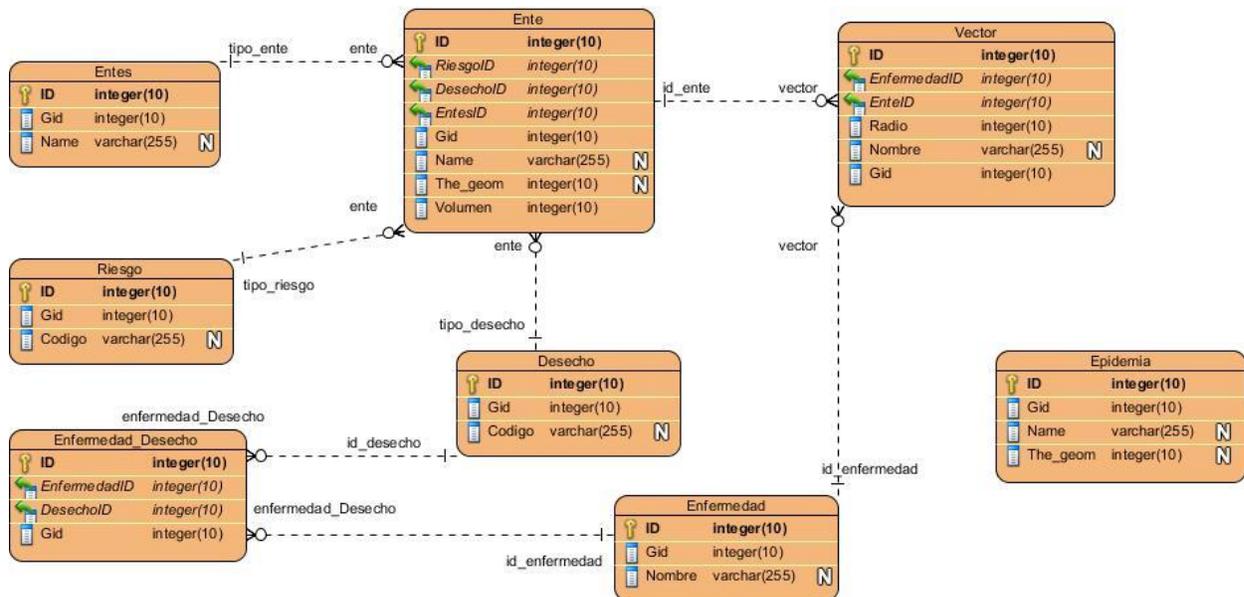


Fig. 11. Diagrama Entidad - Relación

### **2.8 Conclusiones Parciales**

La modelación de las clases del dominio proporcionó el punto de partida para la correcta elaboración del sistema, el levantamiento de los requisitos tanto funcionales como no funcionales proporciona las medidas necesarias con las que debe cumplir el sistema para su correcta implementación. A partir de los artefactos generados, acorde con la metodología de desarrollo AUP, y tomando en cuenta sus especificaciones, es posible realizar la implementación de la propuesta solución que cumpla con las necesidades del sistema que se desea desarrollar.

### **CAPÍTULO 3. CONSTRUCCION Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA**

#### **3.1 Introducción**

En este capítulo se abarcará todo lo referente al proceso de implementación y prueba del sistema. Se presentará el diagrama de componentes que muestra las dependencias entre los componentes de la solución a desarrollar. Además se realizarán las pruebas de software las que tienen como objetivo garantizar la calidad del software y encontrar errores que no fueron descubiertos con anterioridad.

#### **3.2 Modelo de implementación**

El modelo de implementación es una visión general de lo que tiene que ser implementado, este planifica las integraciones de sistemas necesarias en cada iteración, distribuye el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue, implementa las clases y subsistemas encontrados durante el diseño y describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros, probando así los componentes individualmente para después integrarlos (Ivar Jacobson, 2000).

##### **3.2.1 Diagrama de componentes**

Un componente es un elemento de funcionalidad del sistema reutilizable que proporciona y utiliza el comportamiento a través de las interfaces (Alarcón, 2008). Los diagramas de componentes proveen una vista arquitectónica de alto nivel del sistema; posibilitando visualizar el camino para la implementación. Los diagramas de componentes son utilizados para modelar la vista estática del sistema, mostrando la organización y las dependencias lógicas entre los componentes, además de que describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable y representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas.

A continuación se muestran el diagrama de componentes referentes al caso de uso Gestionar Ente Contaminante:

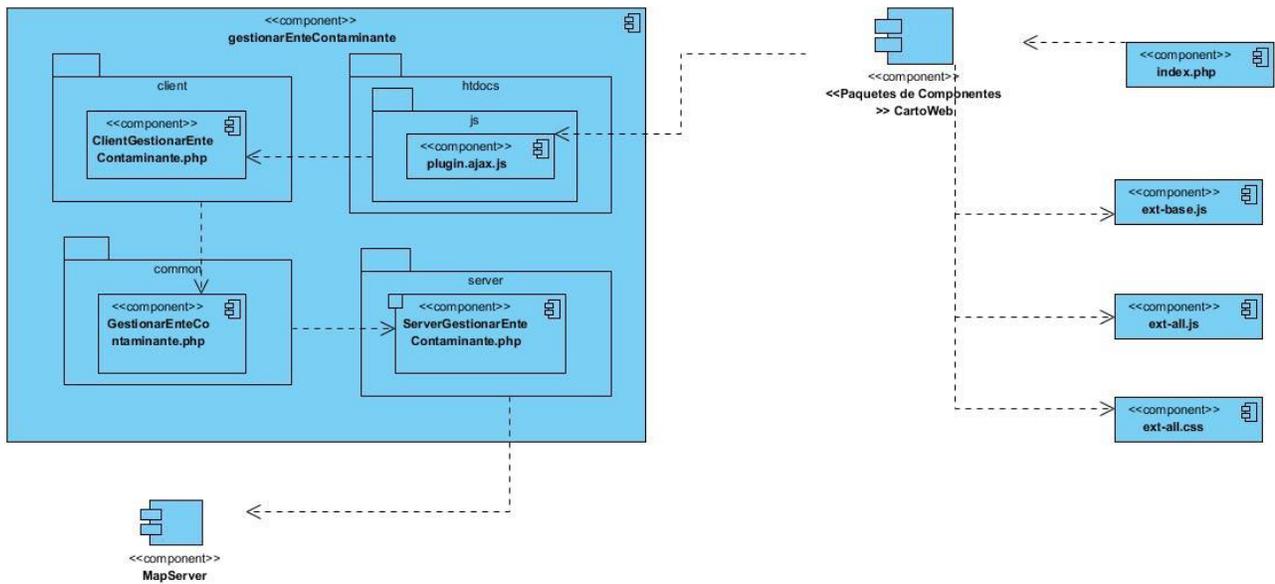


Fig. 12 Diagrama de componentes CU Gestionar Ente Contaminante

### 3.3 Modelo de despliegue

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene algo de memoria y, a menudo, capacidad de procesamiento (Santana Cardoso, 2011).

Los nodos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema. Representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar los componentes.

Los componentes son los elementos que participan en la ejecución de un sistema. Los nodos son los elementos donde se ejecutan los componentes. Los nodos representan el despliegue físico de los componentes. La relación entre un nodo y el componente que despliega puede mostrarse con una relación de dependencia, o listando los nodos desplegados en un compartimiento adicional dentro del nodo.

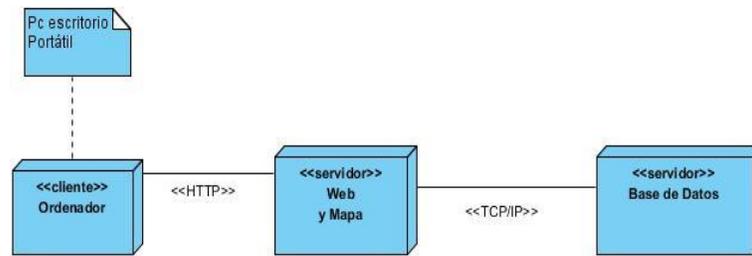


Fig. 13 Diagrama de Despliegue

### 3.4 Descripción de las pruebas.

La prueba es el proceso de ejecución de un programa con la intención de descubrir un error. Las pruebas del software son un elemento crítico para la garantía de la calidad del software (PRESSMAN, 2005).

#### Estrategia de Pruebas

Las pruebas de software constituyen un conjunto de tareas y actividades, que pueden ser planificadas y organizadas antes de ejecutarse, mediante modelos y plantillas asegurando la calidad de este proceso de pruebas. Varios autores como Pressman proponen diferentes estrategias de prueba, las cuales presentan las características siguientes (PRESSMAN, 2005):

- ✓ Comienzan a nivel de módulo.
- ✓ Son apropiadas diferentes técnicas de pruebas.
- ✓ La prueba la lleva a cabo el responsable del desarrollo del software y en los casos en que el proyecto es grande un grupo independiente de pruebas.

Una estrategia de prueba de software debe incluir pruebas de alto nivel, las cuales se encargan de validar que los requisitos implantados por el cliente se correspondan con las funcionalidades del sistema.

#### Niveles de Pruebas (Ivar Jacobson, 2000)

- ✓ *Pruebas Unitarias*: Se basan en detectar errores en los datos, lógica y algoritmos.
- ✓ *Pruebas de Integración*: Se llevan a cabo para detectar errores de interfaces y relaciones entre componentes.
- ✓ *Pruebas Funcionales*: Se llevan a cabo para detectar errores en la implementación de los requerimientos.
- ✓ *Pruebas de Sistema*: Se realizan para detectar fallas en el cubrimiento de los requerimientos.
- ✓ *Pruebas de Aceptación*: Se realizan para detectar fallas en la implementación del sistema.

#### Tipos de Pruebas

## Capítulo 3. Construcción y validación de la solución propuesta

---

- ✓ *Pruebas de Función:* Este tipo de pruebas se realiza con el propósito de verificar el cumplimiento de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.
- ✓ *Pruebas de Regresión:* Se realizan para asegurar que cuando una no conformidad encontrada en el sistema ha sido corregida ninguna de las funcionalidades liberadas previamente falla como resultado de las correcciones o que las características nuevamente agregadas no han creado conflicto con las versiones anteriores del software.
- ✓ *Pruebas de Integración:* Las pruebas de integración se realizan durante la construcción del sistema, con el objetivo de crear la estructura del programa. Así se comprobaría la unión de todos los componentes mediante sus interfaces, además de asegurar que cumplen con las funcionalidades requeridas.
- ✓ *Pruebas de Rendimiento:* Se realizan generalmente para observar el rendimiento del sistema, para observar el tiempo de respuesta que sea el adecuado; también se pueden utilizar para el trabajo con grandes volúmenes de datos así como para el funcionamiento de la aplicación bajo una cantidad de peticiones esperadas (Pressman, 2000).

### Métodos de pruebas

- ✓ *Método de caja blanca:* Se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento.
- ✓ *Método de caja negra:* Se centran fundamentalmente en los requisitos funcionales del software. Estas se realizan sobre la interfaz de la aplicación, con el objetivo de demostrar que las funciones son correctas, que la entrada se acepta de forma adecuada y que se produce una salida correcta (PRESSMAN, 2005).

Para desarrollar la prueba de Caja Negra existen varias técnicas entre ellas:

- ✓ *Técnica de la Partición de Equivalencia:* Divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- ✓ *Técnica del Análisis de Valores Límites:* Prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

Estas pruebas permiten encontrar: (PRESSMAN, 2005)

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o en accesos a las Bases de Datos externas.

## Capítulo 3. Construcción y validación de la solución propuesta

---

- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y terminación.

“El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada, una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica” (Pressman, 2000).

### 3.4.1 Pruebas de software aplicadas

Las pruebas aplicadas al sistema, según los niveles de pruebas, fueron las *pruebas funcionales*, las mismas están dirigidas a que el sistema cumpla con las expectativas del cliente y para detectar errores en la implementación de los requisitos.

Según el tipo de prueba se seleccionó las *pruebas de función* con el propósito de verificar el cumplimiento de los requisitos funcionales y como método de prueba para probar los requerimientos, las funciones y las respuestas de los sistemas se decidió utilizar las *pruebas de caja negra*; además la plataforma GeneSIG es un producto informático liberado, con varios años de explotación, la cual ha transitado por varios procesos de intensiva revisión. Se determinaron además los casos de pruebas por cada caso de uso.

Se selecciona la *técnica de partición equivalente* con el objetivo de dividir el campo de entrada en clases de datos de los que se pueden derivar casos de prueba. Estas divisiones son denominadas diseños de caso de prueba. Permite examinar los valores válidos e inválidos de las entradas existentes en el sistema, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico.

### 3.4.2 Diseño de las pruebas para validar la solución propuesta

El diseño de las pruebas se basa en la creación de casos de prueba cuya ejecución permitirá observar posibles síntomas de defectos. Se puede definir un caso de prueba como “*el conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular como, por ejemplo, ejercitar un camino concreto de un programa o verificar el cumplimiento de un determinado requisito*” (Lara, 2001).

Estos son desarrollados para verificar el cumplimiento total o parcial de un requisito. Las entradas representan las variables que se pueden especificar y las mismas contienen: V, I, o N/A. V indica válido, I

## *Capítulo 3. Construcción y validación de la solución propuesta*

indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante. A continuación se muestra el diseño caso prueba del caso de uso “Gestionar Ente Contaminante” utilizando la técnica de Partición de Equivalencia, los demás pueden ser consultados en el expediente de proyecto.

**Caso de Uso:** Gestionar Ente Contaminante.

**Descripción general del caso de uso:** El caso de uso inicia cuando el especialista necesita gestionar un ente contaminante, insertando, modificando o eliminando determinado ente y termina cuando se visualiza en el mapa los cambios realizados.

**Condiciones de Ejecución:** Solo puede realizar esta acción el especialista y debe para ello estar autenticado en la aplicación.

**Tabla 3. Diseño Caso Prueba CU: “Gestionar Ente Contaminante”.**

Nombre de la sección	Escenario	Descripción	Variables			Respuesta de sistema	Flujo central
			Nom bre	Tipo Des echo	Información Geográfica		
SC 1: “Insertar ente contaminante”.	EC 1.1: “Insertar ente contaminante” con éxito.	El especialista inserta un ente contaminante a partir de sus datos, dicho ente se añade al sistema y se muestra en el mapa una región que lo representa.	V Bas urer o La Hab ana	V Bas urer o	V “Área válida”	Se inserta correctamente un ente contaminante.	Seleccionar en la Barra de herramientas/ Botón “Insertar ente contaminante”/Llenar campos /Botón “Aceptar”/.
	EC 1.2: “Insertar ente contaminante” con campos vacíos.	Si el especialista no llena los campos correspondientes, el sistema muestra un mensaje de error.	I “”	V “”	V “Área válida”	Emite un mensaje de error: “Debe rellenar todos los campos”.	

## *Capítulo 3. Construcción y validación de la solución propuesta*

	EC 1.3: Cerrar la ventana "Insertar ente contaminante".	Al seleccionar el botón "Cancelar" el sistema cierra la ventana "Insertar ente contaminante" sin guardar cambios y muestra el mapa.	NA	NA	NA	El sistema cierra la ventana "Insertar ente contaminante" sin guardar cambios.	
	EC 1.4: "Insertar ente contaminante" con área inválida.	Si el especialista no selecciona un área válida, se muestra un mensaje de error.	V Bas urer o La Hab ana	V Bas urer o	I "Área inválida"	Emite un mensaje de error: "Debe seleccionar un área válida."	
SC 2: "Eliminar ente contaminante".	EC 2.1 "Eliminar ente contaminante" con éxito.	El especialista elimina un ente contaminante a partir de su selección en el mapa, dicho ente se elimina del sistema y se elimina del mapa la región que lo representaba.	NA	NA	V "Área válida"	Queda eliminado el ente contaminante del sistema.	Seleccionar en la Barra de herramientas/ Botón "Eliminar ente contaminante"/ Seleccionar un ente en el mapa / Mensaje de confirmación.
	EC 2.2 "Eliminar ente contaminante" con un área inválida.	Si el especialista no selecciona un área válida para eliminar, el sistema muestra un mensaje de error.	NA	NA	I "Área inválida"	Emite un mensaje de error: "Debe seleccionar un área válida".	Seleccionar en la Barra de herramientas/ Botón "Eliminar ente contaminante"/
SC 3: "Listar	EC 3.1: "Editar ente	El especialista puede editar un ente	V Bas	V Bas	V "Área	El sistema modifica los	Seleccionar en la Barra de

## *Capítulo 3. Construcción y validación de la solución propuesta*

entes contaminantes”.	contaminante” con éxito.	contaminante a partir de sus datos, dicho ente se modifica en el sistema.	urer o La Lisa	urer o	válida”	datos del ente seleccionado y lo guarda en el sistema.	herramientas/ Botón “Listar entes contaminantes”/
	EC 3.2: “Editar ente contaminante” con campos vacíos.	Si el especialista no llena los campos correspondientes, el sistema muestra un mensaje de error.	I “	V Bas urer o	V “Área válida”	Emite un mensaje de error “Debe rellenar todos los campos”.	Seleccionar ente / Botón “Editar”/ Llenar campos /Botón “Aceptar”/.
	EC 3.3: Cerrar la opción “Listar ente contaminante”.	Al seleccionar el botón “Cancelar” el sistema cierra la ventana “Listar ente contaminante” sin guardar cambios y muestra el mapa.	NA	NA	NA	El sistema cierra la ventana “Listar entes contaminantes” sin guardar cambios.	Seleccionar en la Barra de herramientas/ Botón “Listar entes contaminantes”.
	EC 3.4: Eliminar ente contaminante con éxito”.	El especialista elimina un ente contaminante a partir de su selección en el listado, dicho ente se elimina del sistema.	NA	NA	NA	Queda eliminado el ente del sistema.	Seleccionar en la Barra de herramientas/ Botón “Listar entes contaminantes”/ Seleccionar ente / Botón “Eliminar”.

## Capítulo 3. Construcción y validación de la solución propuesta

Tabla 4. Variables del Caso de Prueba: “Gestionar Ente Contaminante”.

No	Nombre del campo	Clasificación	Requerido	Descripción
1	Nombre	Campo de texto	Si	Especifica el nombre del ente contaminante.
2	Tipo de ente	Lista desplegable	Si	Especifica el tipo de ente que puede seleccionar el especialista.
3	Tipo de desecho	Lista desplegable	Si	Especifica el tipo de desecho que está compuesto el ente contaminante.
4	Clasificación de riesgo	Lista desplegable	Si	Especifica la clasificación de riesgo que puede tener este ente contaminante.
5	Volumen	Campo de texto (numérico)	Si	Especifica el volumen estimado que posee el ente contaminante.
6	Ubicar en el mapa	Región	Si	Especifica la región geográfica del ente contaminante.

### 3.4.3 Resultados de las pruebas realizadas

Teniendo en cuenta las condiciones de la solución informática propuesta para la representación de entes contaminantes sobre la plataforma GeneSIG, se realizaron pruebas de caja negra, de las cuales se exponen sus resultados:

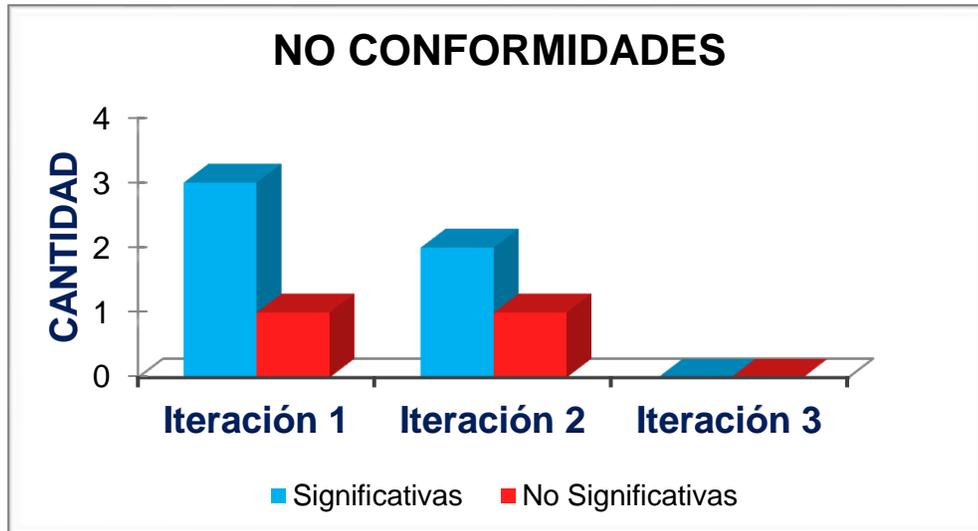
Las pruebas de caja negra se centran en verificar el cumplimiento de los requisitos funcionales del software (PRESSMAN, 2005), con el fin de encontrar la mayor cantidad de no conformidades existentes en el producto. Dentro de las técnicas empleadas por las pruebas de caja negra se utilizó, partición de equivalencia, que según define Pressman: “se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar” (PRESSMAN, 2005).

A continuación se exponen las no conformidades encontradas en las iteraciones de las pruebas realizadas después. Las pruebas funcionales al sistema se realizaron en tres iteraciones para verificar que las funcionalidades implementadas fueron las acordadas con el cliente y que respondían correctamente a sus necesidades. Ver anexo 2.

La siguiente tabla muestra el resumen de los defectos encontrados como resultado de estas pruebas.

## Capítulo 3. Construcción y validación de la solución propuesta

---



En la primera iteración de pruebas al sistema se encontraron 4 No Conformidades de las cuales 3 son Significativas siendo corregidas todas ellas, luego en la segunda iteración se encontraron 3 No Conformidades de las cuales 2 son Significativas siendo todas corregidas, se procedió a realizar una tercera iteración, la cual arrojó resultados satisfactorios, por lo que no se hizo necesaria la realización de más iteraciones.

### 3.5 Conclusiones parciales

En el presente capítulo quedaron plasmados todos los artefactos generados en la fase de construcción según la metodología AUP. Se realizó el diagrama de componentes, representando los elementos más importantes que forman parte del sistema. Se le realizaron pruebas de caja negra al sistema para validar su correcto funcionamiento, con el objetivo de obtener no conformidades y errores que puedan influir en su funcionamiento.

### **CONCLUSIONES GENERALES**

Una vez concluido el proceso investigativo y desarrollado el Sistema de Información Geográfica para el MINSAP, se resaltan una serie de conclusiones las cuales son expuestas a continuación:

1. Con la utilización del sistema desarrollado, la representación y consulta de la información asociada a los entes de contaminación manifestados en el país y que es manejada en el MINSAP, específicamente en las instituciones de sanidad, brinda mayor satisfacción a los trabajadores y clientes de esta entidad y se torna menos engorrosa, requisito indispensable para poder realizar un conjunto de servicios encaminados a brindarle respuesta a las quejas de la población siendo de vital importancia para la toma de decisiones en el sector de la salud.
2. Los artefactos generados en el proceso de desarrollo del SIG servirán para que otros desarrolladores obtengan un mejor entendimiento de su estructura, facilitando la realización de posibles modificaciones o la agregación de nuevas funcionalidades al mismo.
3. Las herramientas y tecnologías utilizadas en el desarrollo de la aplicación obedecen a criterios de selección de tecnologías libres y multiplataforma, adecuándose a las políticas que impulsa la universidad y el país.
4. Todos los requisitos funcionales previstos fueron implementados satisfactoriamente teniendo en cuenta las restricciones no funcionales especificadas a lo largo de la investigación.
5. El diseño de las pruebas de Caja Negra permitió validar el cumplimiento de los requisitos funcionales y comprobar que el sistema efectúa las acciones que debe realizar en el momento esperado.

## **RECOMENDACIONES**

En correspondencia con los resultados obtenidos y la experiencia acumulada a lo largo de todo el proceso investigativo se recomienda lo siguiente:

1. Establecer un período de capacitación para los usuarios del Ministerio de Salud Pública que trabajarán con el Sistema de Información Geográfica, con el objetivo de prepararlos para un buen desarrollo en sus actividades y funcionamiento con la aplicación.
2. Continuar el ciclo de desarrollo de la herramienta, con la identificación de nuevos requisitos funcionales, en aras de ampliar las posibilidades de trabajo del sistema desarrollado.
3. Socializar el resultado de esta investigación en algún espacio de la universidad para que sirva de consulta a investigaciones afines o similares.

## REFERENCIAS BIBLIOGRÁFICAS

- Alarcón, Raúl. 2008.** Diseño Orientado a Objetos con UML. 2008. 2008.
- AUP. 2015.** AUP. [En línea] 2015. [Citado el: 25 de febrero de 2015.]  
[http://ingenieriadesoftware.mex.tl/63758\\_AUP.html](http://ingenieriadesoftware.mex.tl/63758_AUP.html).
- Boletín Epidemiológico. Salud, Organización Panamericana de la. 2004.** 4, 2004, Vol. 25.
- Camptocamp SA. 2008.** CartoWeb: Advanced Geographical Information System for the Web. [En línea] 2008. [Citado el: 25 de abril de 2015.]
- Carlos, Castillo-Salgado. 2003.** *Sistemas de Información Geográfica en Salud:Conceptos Básicos*. Washington, DC. : PAHO, 2003.
- CDC. EpiInfo. 2014.** [En línea] 2014. [Citado el: 13 de noviembre de 2014.] <http://www.cdc.gov/epiinfo/>.
- Craig Larman. 1999.** Patrones Grasp. [En línea] 1999. [Citado el: 2015 de enero de 18.]  
<http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/>.
- Diccionario. 2015.** DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA. [En línea] 2015. [Citado el: 16 de junio de 2015.]  
<http://www.alegsa.com.ar/Dic/aplicacion.php>.
- Epidemiológico, Boletín. 2001.** 3, 2001, Vol. 22.
- Espinosa, A. M. 2000.** *Fundamentos del Mapserver, Mapscript, PostGIS y su integración con el Cartoweb*. Habana : s.n., 2000.
- Espinosa, Antonio Membrides, Ramos, Eduanys Puerta y Antunez, Romanuel Ramon. 2012.** *Genesisig. Desarrollo de Sistemas de Información Geográficos sobre una plataforma soberana*. 2012.
- Falero, Yadira Hernández Inza & Héctor René Sánchez. 2010.** *Gestión de Nomencladores para el Programa Nacional de Informatización del Conocimiento Geológico*. 2010.
- Investigación, Revista de. 2015.** Revista de Investigación. [En línea] 2015. [Citado el: 25 de enero de 2015.]  
<http://issuu.com/revistadeinvestigacionupel-ipc/docs/r80-6>.
- Ivar Jacobson, Grady Booch,James Rumbaugh. 2000.** *Proceso unificado de desarrollo de software*. . Madrid : ISBN, 2000. s.n.
- Json. 2011.** [En línea] 2011. [Citado el: 2 de mayo de 2015.] <http://json.org/json-es.html>.
- Kicillof, Nicolás y Reynoso, Carlos. 2004.** Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. 2004.
- Lanzillotta, Analía. 2005.** Master Magazine. [En línea] 2005. [Citado el: 2 de mayo de 2015.]  
<http://www.mastermagazine.info/termino/5560.php>.
- Lara, P. and Fernández, L. 2001.** *Generación de casos de prueba a partir de especificaciones UML*. 2001.
- Larman, Craig. 2003.** *UML y Patrones*. 2003. 2da, Edición..
- MappingGis. 2015.** MappingGis. [En línea] 2015. [Citado el: 2 de mayo de 2015.]  
<http://mappinggis.com/2012/09/por-que-utilizar-postgis/>.
- MapServer. 2015.** MapServer. [En línea] 2015. [Citado el: 19 de junio de 2015.] <http://mapserver.org/es/>.
- Martínez Piedra, R. 2001.** Boletín Epidemiológico.SIGEpi. s.l. : OPS, 2001, Vol. 22.
- Mateus Pulido, Pablo Andrés. 2009.** *Registro y gestión patrimonial de yacimientos arqueológicos bajo*. 2009.

- Moreta, Orlando Ramiro. 2009.** *Diseño e implementación de Mapa Interactivo utilizando Web Mapping y Base de Datos Espacial: Ciudad de Quevedo.* 2009.
- NetBeans IDE. 2015.** [En línea] 2015. [Citado el: 2 de mayo de 2015.] <https://netbeans.org/features/index.html>.
- Núñez, Héctor Manuel Fernández. 2006.** Revista Cubana Higiene Epidemiología. [En línea] diciembre de 2006. [Citado el: 13 de junio de 2015.] [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1561-30032006000300003](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1561-30032006000300003).
- Núñez, M. H. 2006.** Revista Cubana Higiene Epidemiología. [En línea] 2006. [http://bvs.sld.cu/revistas/hie/vol44\\_3\\_06/hie03306.htm](http://bvs.sld.cu/revistas/hie/vol44_3_06/hie03306.htm).
- Palau, Pedro Enrique y Rodríguez, Luis José Tristá. 2012.** *Sistema de Información Geográfica para el Ministerio de Educación Superior.* 2012.
- Patrones. 2001.** Patrones de diseño. [En línea] 2001. [Citado el: 11 de febrero de 2015.] <http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>.
- PostgreSQL. 2013.** PostgreSQL. [En línea] 2013. [Citado el: 2 de mayo de 2015.] <http://www.postgresql.org/about>.
- Potencier, Fabien. 2008.** *Symfony 1.1, la guía definitiva.* [En línea] 2008.
- PRESSMAN. 2005.** *Ingeniería del Software: Un enfoque práctico.* 2005. [En línea] 2005.
- Pressman, Roger. 2000.** *Ingeniería de Software un Enfoque Práctico.* 2000. Vol. 6ta Edición.
- Prieto, F. 2008.** Patrones de diseño. [En línea] 2008. [Citado el: 11 de febrero de 2015.] [http://www.infor.uva.es/~felix/datos/priiii/tr\\_patrones-2x4.pdf](http://www.infor.uva.es/~felix/datos/priiii/tr_patrones-2x4.pdf).
- Rae. 2015.** Real Academia Española. [En línea] 2015. [Citado el: 12 de junio de 2015.] <http://www.rae.es/>.
- Santana Cardoso, Yesenia. 2011.** *Tecnologías de la Información y la Comunicación.* 2011. 2011.
- Scribd. 2015.** *Arquitectura Basada en Componentes.* [En línea] 2015. [Citado el: 12 de marzo de 2015.] <http://es.scribd.com/doc/14704374/Arquitectura-Basada-en-Componentes#scribd>.
- SIG. 2015.** SIG. [En línea] 2015. <https://langleruben.wordpress.com/%C2%BFque-es-un-sig/>.
- Teja Pérez, Julio. 1987.** *La Habana : INSPECCION SANITARIA ESTATAL. RESOLUCIÓN MINISTERIAL, 1987, Gaceta Oficial de la República de Cuba. Salud Pública., Vol. 215.*
- The Apache Software Apache. 2012.** Apache. [En línea] Apache License, 2012. <http://www.apache.org/licenses/LICENSE-2.0.html>.
- Torchetti. 2015.** [En línea] 2015. [Citado el: 26 de enero de 2015.] <http://www.gsdoc.org/gsdiconf/GSDI-5/papers/Ricardo>.
- Valle, Dianelys del. 2009.** *Desarrollo del Portal WAP para la plataforma de gestión de contenidos Gina.* Habana : s.n., 2009. s.n.

## **ANEXOS**

### **Anexo 1. Preguntas de la guía de entrevista:**

**Entrevistada:** Dra María Graciela González Vázquez, Epidemióloga Municipio Bauta.

**Objetivos:** Conocer los procedimientos que se realizan para identificar riesgos de enfermedad en el Municipio.

#### **Preguntas:**

1. ¿Tienen identificados de alguna manera que lugares en Bauta son propensos a determinadas enfermedades?
2. ¿Se tiene ubicación geográfica de estos lugares? ¿Es posible que los especialistas se ubiquen como llegar sin dificultades?
3. ¿Se tiene algún estudio demográfico de estas zonas?
4. ¿Se identifican zonas de contaminación en las áreas de riesgo?

**Entrevistada:** Osmaida López Salvador, Técnica Lucha Antivectorial.

**Objetivos:** Identificar problemas para la localización y eliminación de basureros urbanos

1. ¿Conocen en detalle la ubicación de los basureros urbanos en el municipio?
2. ¿Se conoce la cantidad exacta de basureros, zanjas o vertederos de basura existentes?
3. ¿Cómo se planifica la eliminación de un basurero urbano por parte de los entes de Salud Pública?
4. ¿Cómo se asignan los recursos para la eliminación de estos entes contaminantes?

## Anexo 2. No conformidades detectadas al aplicar los casos de prueba.

**Tabla 5. No Conformidades detectadas al aplicar los casos de prueba al sistema.**

Iteración	No. NC	Descripción	Significativas	No Significativas	Ubicación	Estado
1.	1	Falta de ortografía en la ventana Insertar ente contaminante. Faltan tildes.		x	Barra de herramientas/ opción "Insertar ente contaminante".	Resuelta.
	2	No se muestran correctamente todos los datos de los entes, aún faltan datos por mostrar.	x		Barra de herramientas/ opción "Listar entes contaminante"/ Botón "Mostrar Datos".	Resuelta.
	3	El sistema no cancela el listado de epidemias.	x		Barra de herramientas/ opción "Listar Epidemias"/ Botón "Cerrar ventana".	Resuelta.
		La interfaz correspondiente a esta sección no se muestra en la aplicación, por lo que no se puede probar ningún escenario de esta sección.	x		Barra de herramientas/ opción "Tematizar"	Resuelta.
2.	1	El sistema no responde cuando se le da clic al botón "Editar".	x		Barra de herramientas/ opción "Listar Epidemias"/ Seleccionar Epidemia/ Botón "Editar".	Resuelta.
	2	El sistema no responde cuando se le da clic al botón "Ver ubicación en el mapa".	x		Barra de herramientas/ opción "Listar entes por área / Botón "Ver ubicación en el mapa".	Resuelta.
	3	Algunos errores que se muestran para especificar los valores que el usuario debe entrar son escritos en inglés.		x	Barra de herramientas/ opción "Insertar vector"	Resuelta.