

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 6**



**Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

**Título: “Módulo de operaciones geométricas en capas
vectoriales para GeneSIG.”**

Autora: Liz Mavis Mestre Junco

Tutor(es): MSc. Romanuel Ramón Antúnez

Ing. Lidisy Hernández Montero

La Habana, junio del 2014
“Año 56 de la Revolución”

“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: La Voluntad.”

Albert Einstein.

Declaración de Autoría

Declaro ser la legítima autora del trabajo titulado: “Módulo de operaciones geométricas en capas vectoriales para GeneSIG”, y reconozco a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Firma del Autor

Liz Mavis Mestre Junco

Firma del Tutor

Ing. Lidisy Hernández Montero

Firma del Tutor

MSc. Romanuel Ramón Antúnez

Datos de contacto

Nombre y Apellidos: MSc. Romanuel Ramón Antúnez.

Correo electrónico: rromon@uci.cu

Categoría docente: Asistente.

Año de graduación: 2008.

Profesión: Ingeniero en Ciencias Informáticas.

Máster en Informática Aplicada: 2011.

Nombre y Apellidos: Ing. Lidisy Hernández Montero.

Correo electrónico: lhernandez@uci.cu

Categoría docente: Instructora.

Año de graduación: 2008.

Profesión: Ingeniero en Ciencias Informáticas.

A mi Mami querida por estar siempre a mi lado y brindarme todo su amor, dedicación y paciencia, por ser mi fuerza cuando la pensaba perdida, este resultado mami, es para ti.

A mi tía Margarita por su apoyo en todo momento y a mi primita Lisy.

A mis abuelos Herito por su constante preocupación y Rosalba que desde donde se encuentre siempre nos cuida.

A mi familia por su constante apoyo en las buenas y malas.

A mi mamá por estar a mi lado en todo momento, por permitirme tomar mis propias decisiones, por ser mi inspiración y por hacer de mi la persona que soy hoy.

A mi tía Margarita (Chi) por cuidarme, aconsejarme y ser mi amiga.

A mi segunda familia, mis amigos del alma Aymé (Pote), Arlettis, Ana, Alfre, Yadira (La mía), Enier y Julio por ser mi sustento en los momentos difíciles de mi vida, y por hacerme sonreír hasta en los malos tiempos; será difícil el camino sin ustedes.

A Alain que sin él no sabría cuál hubiese sido el resultado de esta investigación, por las incontables noches de dedicación, por estar ahí cuando lo necesitaba, muchas gracias por tu completa entrega.

A mis tutores Lidisy y Romanuel por confiar en mi.

A Rafael por dedicarme parte de su tiempo y ayuda.

A mis compañeros de aula que hicieron de estos cinco años memorables.

A mis profes que durante la carrera aportaron su granito de arena en mi formación.

Agradezco infinitamente a todos los que me brindaron su ayuda.

Gracias a todos.

Resumen

La presente investigación tiene como objetivo dar a conocer la relevancia del “Módulo de operaciones geométricas en capas vectoriales para GeneSIG”. Dicho módulo surge de la necesidad de contar con una herramienta que permita realizar cálculos geométricos como: operaciones de solape, determinación de contornos mínimos y determinación de zona de influencia. En su desarrollo se tuvo en cuenta la soberanía tecnológica por la que aboga hoy Cuba. Debido a ello se utilizaron herramientas y tecnologías libres teniendo como base la plataforma soberana GeneSIG creada en la Universidad de las Ciencias Informáticas. Este módulo brinda la posibilidad de determinar zonas de influencia mediante la inserción de condiciones y parámetros, garantizando obtener resultados precisos y confiables sobre los datos analizados; permitiendo de este modo, contribuir al proceso de toma de decisiones sobre cualquier ámbito socioeconómico hacia los estén dirigidos los Sistemas de Información Geográfica desarrollados sobre GeneSIG.

Palabras clave: Contornos mínimos, Operaciones de solape, Plataforma GeneSIG, Sistema de Información Geográfica, Zona de influencia.

Abstract

This research aims at making public the importance of the "Module of geometric stratified vectorial operations for GeneSIG". This module arises from of having a tool carry out geometric calculations such overlapping operations, determination of convex hull and determination of buffer. In its development the technological sovereignty advocated by Cuba has been considered. Due to these facts, free tools and technologies having like base the sovereign platform created at the University of Informatics Sciences. This module provides the ability to identify buffer by inserting conditions and parameters, ensuring to accurate and reliable results on the analyzed data; thus allowing to, contributing to the making decisions process on any socio-economic environment towards which the Geographical Information Systems developed on GeneSIG are addressed.

Keywords: Buffer, Convex hull, GeneSIG Platform, Geographic Information System, Overlay operation.

Índice

Introducción	1
Capítulo.1 Marco Teórico	5
1.1 Introducción	5
1.2 Principales conceptos	5
1.2.1 SIG.....	5
1.2.2 <i>Formato vectorial</i>	6
1.2.3 <i>Zonas de influencia</i>	6
1.2.4 <i>Operaciones de solape</i>	6
1.2.5 <i>Contornos mínimos</i>	7
1.2.6 <i>Algoritmo</i>	8
1.2.6.1 <i>Algoritmo para el cálculo de zonas de influencia</i>	9
1.2.6.2 <i>Algoritmos de cálculo de contornos mínimos</i>	14
1.2.6.3 <i>Algoritmos de cálculo de operaciones de solape</i>	16
1.2.7 <i>Plugin</i>	17
1.3 Análisis de soluciones existentes	17
1.3.1 <i>GDAL</i>	18
1.3.2 <i>Sextante</i>	19
1.3.3 <i>PostGIS</i>	20
1.3.4 <i>Observación de las soluciones existentes</i>	21
1.4 Metodología, Herramientas, Tecnología y Lenguajes de desarrollo	21
1.4.1 <i>Metodología de desarrollo</i>	21
1.4.2 <i>Lenguaje de Modelado Unificado</i>	21
1.4.3 <i>Herramienta Case</i>	22
1.4.4 <i>Lenguaje de desarrollo</i>	22
1.4.5 <i>Herramientas de desarrollo</i>	24
1.5 Conclusiones	26
Capítulo.2 Análisis y Diseño	27

2.1	Introducción	27
2.2	Propuesta de solución.....	27
2.3	Modelo del dominio.....	27
2.3.1	<i>Diagrama de clases del dominio</i>	28
2.3.2	<i>Breve descripción del modelo de dominio</i>	28
2.4	Técnicas de extracción de requisitos.....	28
2.5	Requisitos funcionales	29
2.6	Requisitos no funcionales.....	30
2.7	Diagrama de caso de uso.....	32
2.7.1	<i>Actores del sistema</i>	32
2.7.2	<i>Especificación de los casos de usos del sistema</i>	32
2.8	Patrón arquitectónico.....	39
2.8.1	<i>Patrones de diseño</i>	40
2.9	Diagrama de paquete.....	42
2.9.1	<i>Breve descripción de los paquetes del módulo</i>	42
2.10	Modelo de clases del diseño	43
2.11	Modelo de Datos	44
2.12	Conclusiones	45
Capítulo.3	Implementación y Pruebas.....	46
3.1	Introducción.....	46
3.2	Implementación	46
3.3	Diagrama de Componente.....	46
3.4	Diagrama de Despliegue	47
3.5	Prueba	48
3.6	Pruebas realizadas al módulo.....	48
3.6.1	<i>Diseño de casos de prueba</i>	49
3.7	Resultados de las pruebas	56
3.8	Conclusiones	56
Conclusiones	58

Recomendaciones	59
Referencias.....	60
Bibliografía	65
Anexos.....	70
1.1. Entrevista realizada al líder del proyecto Aplicativos SIG	70

Índice de Figuras

Fig. 1 Entrada de datos tipo punto (ESRI, 2012).....	9
Fig. 2 Operación Intersección (ESRI, 2012).....	9
Fig. 3 Operación de disolución (ESRI, 2012).....	10
Fig. 4 Entrada de datos tipo línea (Rivera, 2004).....	10
Fig. 5 Nodo inicial y final (Rivera, 2004).	10
Fig. 6 Cálculo de Δx y Δy (Rivera, 2004).	11
Fig. 7 Resultado de las ecuaciones expuestas anteriormente (Rivera, 2004).....	11
Fig. 8 Análisis completo de las líneas del buffer (Rivera, 2004).	12
Fig. 9 Método de truncamiento (Rivera, 2004).	13
Fig. 10 Método de adelgazamiento (Rivera, 2004).	13
Fig. 11 Método de envolvente (Rivera, 2004).	14
Fig. 12 Entrada de datos tipo polígono (Rivera, 2004).....	14
Fig. 13 Buffer externo e interno respectivamente (Rivera, 2004).	14
Fig. 14 Operaciones de solape en PostGIS (PostGIS, 2010).	20
Fig. 15 Diagrama de clases del dominio.	28
Fig. 16 Diagrama de casos de uso.	32
Fig. 17 Diagrama de paquetes.	42
Fig. 18 Diagrama de clases del diseño.	43
Fig. 19 Modelo Entidad-Relación.	44
Fig. 20 Diagrama de clases persistentes.....	45
Fig. 21 Diagrama de Componentes.....	47
Fig. 22 Diagrama de despliegue.	48
Fig. 23 Resultado de las pruebas.	56

Índice de Tablas

Tabla.1 Comparación entre los algoritmos de contornos mínimos.....	15
Tabla.2 Comparación entre los algoritmos de solape.....	16
Tabla.3 Actores del sistema.....	32
Tabla.4 Descripción del caso de uso Gestionar zona de influencia.	32
Tabla.5 Descripción de las variables.	49
Tabla.6 Descripción de las secciones.	49
Tabla.7 Descripción de la SC 1 “Insertar”.....	53
Tabla.8 Descripción de la SC 2 “Modificar”.....	54
Tabla.9 Descripción de la SC 3 “Mostrar”.....	55
Tabla.10 Resultados de las pruebas de caja negra.....	56

Introducción

En la década de los años 60 fueron lanzados varios satélites para efectuar estudios geodésicos completos de la superficie terrestre, por medio de equipos fotográficos de alta resolución colocados en esos satélites. El desarrollo de estos sensores permitió imágenes con resoluciones mucho más precisas, es decir, más claras y detalladas de la superficie terrestre. Propiciando que la cartografía¹ experimentara una serie de innovaciones técnicas en cuanto a su soporte, que dejó de ser el papel, para convertirse en digital, naciendo así los Sistemas de Información Geográfica (SIG). Convirtiéndose en una herramienta de suma importancia a nivel mundial, debido a su innumerable utilidad en el análisis geoespacial a través de cálculos (Mora, 2009).

Determinar la distancia entre puntos, la longitud de una línea o la superficie de un objeto son operaciones fundamentales que traen consigo los SIG. Para realizar un análisis geodésico más riguroso se emplean cálculos geométricos como: zonas de influencia, operaciones de solape y la determinación de contornos mínimos; que con el paso del tiempo se hizo indispensable la adopción de algoritmos que permitieran mostrar con mayor exactitud y rapidez los resultados de las operaciones geométricas aplicadas sobre el terreno (Felicísimo, 2000).

Con los avances de las Tecnologías de la Información y la Comunicación (TIC) se ha logrado un progresivo desarrollo de los SIG a nivel mundial, en cuanto a la calidad y robustez de la información que manejan, a través de la introducción de recursos tecnológicos que permiten adquirir, almacenar y realizar un análisis de la información obtenida desde diversas fuentes (Servicios Académicos Internacionales S.C. , 2013). Cuba no ha estado exenta del desarrollo progresivo de los SIG, siendo estos muy versátiles en muchas esferas, como por ejemplo: la transportación de pasajeros, los trabajos arqueológicos, la minería, los yacimientos de petróleo, entre otros (Batista Rodríguez, y otros, 2007).

La Universidad de las Ciencias Informáticas desempeña un papel protagónico en este ámbito con el desarrollo del proyecto GeneSIG del Centro de Geoinformática y Señales Digitales (GEySED) en conjunto con GeoCuba y las Fuerzas Armadas Revolucionaria (FAR). Este proyecto tiene como principal propósito desarrollar la plataforma soberana GeneSIG para el desarrollo de aplicaciones SIG en un ambiente web. La misma cuenta con un conjunto de *plugins* donde se agrupan la mayoría de las funcionalidades de los SIG convencionales.

En la plataforma se trabaja a partir de datos *raster* y vectoriales, para el caso de estos últimos las entidades asociadas a cada atributo tienen sus propias características espaciales y la geometría que sirve por sí sola para

¹ Cartografía: De *carta*: Arte de trazar mapas geográficos y *grafía*: Ciencia que los estudia.

Llevar a cabo numerosos análisis. Si se une a los atributos que esta geometría lleva asociados, se tiene la posibilidad de realizar un número mayor de estos análisis.

Actualmente la plataforma carece de una serie de operaciones que transforman los datos vectoriales actuando sobre sus geometrías, como es el caso del cálculo de zonas de influencia, operaciones de solape y contornos mínimos. Los resultados de estas operaciones son nuevas capas cuyas geometrías aportan información adicional a las geometrías originales, o bien las transforman para que su uso sea más adecuado en otros análisis u operaciones. Dichos análisis apoyan el proceso de toma de decisiones mediante la visualización de una zona de influencia.

El resultado de la zona visualizada en el mapa digital podrá en gran medida contrarrestar una futura propagación epidemiológica, es decir, permitirá obtener un pronóstico del área afectada y de este modo evacuar las zonas adyacentes a la infección con mayor rapidez. En otro caso donde hipotéticamente se desea colocar antenas de señales digitales, se determinaría la zona de influencia que contiene cada antena, cuyo radio será el de la onda emitida, comprobando que exista la menor cantidad de intercepciones posible, así la señal ocupará mayor área con menos emisores y el país ahorraría recursos materiales.

Por lo que se propone el siguiente problema a resolver: **¿Cómo obtener la información asociada a las geometrías presentes en capas vectoriales?**

Con el propósito de darle una solución efectiva al problema se plantea como **objetivo general** desarrollar un módulo para la plataforma GeneSIG que permita la realización de operaciones geométricas sobre capas vectoriales.

Partiendo del problema a resolver el **objeto de estudio** queda definido por los algoritmos que permiten la realización de operaciones geométricas sobre capas vectoriales.

Enmarcándose en el **campo de acción** los algoritmos geométricos de cálculo de zonas de influencia, operaciones de solape (recorte, diferencia, intersección y unión) y contornos mínimos (envoltura convexa, círculo mínimo y rectángulo mínimo).

Para facilitar la solución del problema a resolver planteado anteriormente se definen las siguientes **preguntas de la investigación**:

1. ¿Cuáles son los fundamentos teóricos de los algoritmos que permiten la realización de operaciones geométricas sobre capas vectoriales?

2. ¿Cuáles son las características que debe cumplir el Módulo de Operaciones Geométricas para la plataforma GeneSIG, que permita la realización de operaciones geométricas sobre capas vectoriales?

3. ¿El módulo desarrollado para la plataforma GeneSIG permite la realización de operaciones geométricas sobre capas vectoriales?

Para dar cumplimiento al objetivo trazado se plantean las siguientes **tareas de la investigación** que permiten una mejor guía de este proceso:

- Caracterización de los algoritmos y procedimientos para determinar zonas de influencia.
- Caracterización de los algoritmos y procedimientos para realizar operaciones de solape.
- Caracterización de los algoritmos y procedimientos para obtener contornos mínimos.
- Selección a partir del análisis de las bibliotecas existentes de algoritmos geométricos que se ajusten a la problemática a resolver.
- Diseño de la solución sobre la Plataforma GeneSIG de los algoritmos para determinar zonas de influencia.
- Implementación de los algoritmos para determinar zonas de influencia.
- Prueba y corrección de errores de la implementación.
- Integración del *plugin* a la Plataforma GeneSIG.
- Prueba y corrección de errores de la integración.

Al culminar la realización de las tareas de la investigación definidas se esperan los siguientes resultados:

1. Módulo para realizar operaciones geométricas en capas vectoriales para la Plataforma GeneSIG.
2. Artefactos ingenieriles correspondientes al módulo de operaciones geométricas.

Como punto de partida de la investigación se utilizaron durante su desarrollo los siguientes **Métodos Científicos**:

Métodos Teóricos:

Análisis-Síntesis: Se aplica en el análisis de las bibliografías seleccionadas para desarrollar el fundamento teórico de la presente investigación y sintetizar los principales conceptos asociados a las operaciones geométricas en capas vectoriales.

Histórico-lógico: Se utiliza para el estudio de trabajos e investigaciones anteriores realizadas durante el desarrollo de la plataforma GeneSIG.

Modelación: Se aplica en la modelación de los distintos procesos, artefactos y actividades que se realizan en la construcción del módulo propuesto.

Método Empíricos:

Observación: Se utiliza para efectuar una observación de la información de cada uno de los conceptos relacionados con los algoritmos geométricos, así como sus métodos para calcular las operaciones asociadas a la investigación con el objetivo de realizar una selección acertada.

Entrevista: Este método se utiliza en la realización de entrevistas a los desarrolladores del proyecto GeneSIG y Aplicativos SIG. Se aplica con el fin de obtener las principales deficiencias que dan origen al problema planteado y realizar el levantamiento de requisitos.

La investigación está estructurada de la siguiente manera:

Capítulo.1 Marco teórico

Contiene la fundamentación teórica del tema a desarrollar. Se describen los conceptos fundamentales sobre los SIG y los algoritmos geométricos. Se exponen las herramientas, metodologías, lenguajes y plataforma a utilizar para la implementación del módulo.

Capítulo.2 Análisis y Diseño

Se realiza la presentación de la solución propuesta del módulo y su funcionamiento, así como los requisitos tanto funcionales como no funcionales con los que debe cumplir el sistema y los artefactos generados por él.

Capítulo.3 Implementación y Pruebas

Se aborda todo el proceso de implementación de la solución propuesta, en función de los diagramas de componentes y despliegue y se valida la solución a partir de pruebas de caja negra realizadas al módulo de operaciones geométricas en capas vectoriales.

Capítulo.1 Marco Teórico

1.1 Introducción

En este capítulo se realiza un análisis teórico de los conceptos fundamentales sobre los SIG y los algoritmos geométricos. Se exponen las herramientas, metodologías, lenguajes y plataforma a utilizar para la implementación del módulo. Se detalla con claridad el objeto de estudio que dio origen a la investigación, además del análisis de las alternativas que pueden darle solución al problema.

1.2 Principales conceptos

En los siguientes epígrafes se definen los principales conceptos relacionados con el módulo de operaciones geométricas, para un mejor entendimiento de la investigación.

1.2.1 SIG

A continuación se expone una muestra de las diferentes definiciones que conceptualizan a un SIG:

“Es un sistema de hardware, software y procedimientos diseñado para realizar la captura, almacenamiento, manipulación, análisis, modelización y presentación de datos referenciados espacialmente para la resolución de problemas complejos de planificación y gestión” (NCGIA, 1990).

“Un SIG es una integración organizada de hardware, software, datos geográficos y personal, diseñado para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información” (LANGHI, 2002).

“Un SIG se define como un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos. Los SIG son una nueva tecnología que permite gestionar y analizar la información espacial y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato” (Brenes, 2005).

Se concluye que un SIG no es más que un sistema informático que logra integrar tanto *hardware* como *software* con el fin de gestionar datos geográficos. Permite visualizar y analizar la información espacial para obtener una mejor toma de decisiones.

1.2.2 Formato vectorial

Los datos espaciales en un SIG pueden ser representados a través de dos formatos o sistemas espaciales: vectorial y *raster*. En el modelo vectorial, no existen unidades fundamentales que dividan la zona recogida, sino que se recoge la variabilidad y características de esta mediante entidades geométricas, para cada una de las cuales dichas características son constantes. La forma de estas entidades (su frontera), se codifica de modo explícito, a diferencia del modelo *raster*, donde venía implícita en la propia estructura de la malla. El formato vectorial representa el espacio geográfico mediante una serie de primitivas geométricas que contienen los elementos más destacados de dicho espacio. Estas primitivas son de tres tipos: puntos, líneas y polígonos (Olaya, 2011).

1.2.3 Zonas de influencia

Una de las transformaciones más importantes con capas vectoriales es la creación de zonas de influencia, también conocidas como *buffers*. Esta transformación puede llevarse a cabo con entidades de tipo punto, línea o polígono, y su resultado siempre es una nueva capa de polígonos. Las áreas cubiertas por estos polígonos reflejan las zonas de influencia de cada entidad, influencia que se considera la ejerce hasta una distancia dada.

Calcular un *buffer* de un punto es un procedimiento sumamente sencillo, ya que esta zona es simplemente un círculo centrado en el punto y como radio la distancia máxima de influencia, no sucediendo lo mismo con el caso de considerar líneas en lugar de puntos, la operación es conceptualmente similar, aunque el algoritmo es notablemente más complejo. Para una capa de polígonos, las zonas de influencia pueden establecerse no solo hacia el exterior, sino también hacia el interior de la geometría. En el caso de extender el área del polígono, la interpretación es la misma que en los ejemplos anteriores (Olaya, 2011).

1.2.4 Operaciones de solape

Las operaciones de combinación de capas equivalentes, expresadas en el álgebra de mapas mediante funciones locales son denominadas operaciones de solape. Estas operaciones permiten generar nuevas capas vectoriales a partir del cruce de dos de ellas, pudiendo dichas capas de origen contener distintos tipos de entidades, aunque principalmente regiones (polígonos). La naturaleza de estas operaciones es bien distinta a la de sus equivalentes

raster. Por una parte, se basan en cálculos geométricos que utilizan las coordenadas de cada entidad para obtener nuevas entidades resultantes.

Se encuentran entre dichas operaciones:

- Recorte o *clipping*.
- Diferencia.
- Intersección.
- Unión.

En el libro Sistemas de Información Geográfica, Víctor Olaya expone que en el recorte o *clipping*, se toma la capa que contiene los parámetros de interés, y otra capa de polígonos que contiene aquellas regiones que resultan de interés. El resultado mantiene la información, pero solo mantiene aquellas entidades que se incluyen total o parcialmente dentro de los polígonos de recorte.

Al contrario del recorte la diferencia, mantiene las capas que no entran dentro de la zona definida por el recorte, aunque el principio es el mismo del anterior.

La intersección es equivalente a una operación booleana (*AND*), ya que la capa resultante mantiene solo aquellas zonas para las que se dispone de información en ambas capas de entrada. Es decir, aquellas zonas donde hay entidades en ambas capas. El cruce de las capas, puede resultar imprecisa debido a la digitalización, esa falta de coincidencia da lugar a los llamados polígonos adicionales que no deberían estar allí, se denomina polígonos espúreos o *sliver polygons*.

La unión es semejante al operador lógico (*OR*). El resultado de la operación está dado por el conjunto de geometrías de la intersección y junto a ella las que corresponden a las zonas que aparecen únicamente en una de las capas de origen. Con el cruce de las capas, y al igual que en la intersección, las geometrías se dividen en trozos restantes que conforman la capa resultante.

1.2.5 Contornos mínimos

Una de las operaciones geométricas que pueden llevarse a cabo sobre un conjunto de puntos es la delimitación de un contorno mínimo que los englobe.

Existen diversas formas de contornos mínimos:

- Envoltente convexa mínima.
- Rectángulo mínimo.

- Círculo mínimo.

La envolvente convexa mínima o *convex hull*, define el polígono convexo de menor área dentro del cual se contienen todos los puntos del conjunto. Delimita el área dentro de la cual se puede inferir el comportamiento de una variable. El polígono que define a la envolvente mínima convexa puede ser empleado así mismo como dato de entrada para otras operaciones geométricas. Una serie de envolventes convexas sucesivas sobre un conjunto de puntos forma una teselación² en capas en forma de cebolla.

El círculo mínimo es aquel que contiene a todos los puntos de un conjunto dado con el menor radio posible los puntos situados en el borde del círculo mínimo siempre pertenecen a su vez a la envolvente mínima convexa. Por ello, el problema del cálculo del círculo mínimo para un conjunto de n puntos puede reducirse al cálculo de dicho círculo para el subconjunto de puntos que componen la envolvente mínima convexa.

El rectángulo mínimo es el rectángulo de menor área que cubre todos los puntos de un conjunto. Para el cálculo del rectángulo mínimo se aplica el hecho de que al menos un lado de este se sitúa sobre un lado de la envolvente mínima convexa. Por ello, basta comprobar los distintos rectángulos que pueden construirse sobre dicha envolvente, y tomar el de menor área. Generalmente, el rectángulo de menor área coincide con el de menor perímetro, pero no siempre es así; a pesar de esta característica, este último también cumple la condición citada con respecto a la envolvente convexa, por lo que su cálculo puede hacerse por un procedimiento idéntico (Olaya, 2011).

1.2.6 Algoritmo

“Algoritmo es un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema” (Wordreference, 2013).

“Algoritmo es un grupo finito de operaciones organizadas de manera lógica y ordenada que permite solucionar un determinado problema. Se trata de una serie de instrucciones o reglas establecidas que, por medio de una sucesión de pasos, permiten arribar a un resultado o solución” (Definición.de, 2013).

Se define como algoritmo al sistema de operaciones ordenadas lógicamente que brindan la posibilidad de arribar a la solución de un problema.

En los subepígrafes 1.2.6.1, 1.2.6.2 y 1.2.6.3 se definen los algoritmos de operaciones geométricas que sustentan el desarrollo del módulo propuesto.

²Teselación: Está compuesta de una o más formas idénticas que se repiten para formar un patrón.

1.2.6.1 Algoritmo para el cálculo de zonas de influencia

Para el cálculo de zonas de influencia no existen algoritmos definidos, sino procedimientos que dependen del tipo de entrada, sea puntos, líneas o polígonos.

Datos espaciales tipo punto (ver Fig 1):



Fig. 1 Entrada de datos tipo punto (ESRI, 2012)

El proceso solamente implica generar un polígono circular alrededor de cada punto con un radio igual al ancho del *buffer* (Rivera, 2004).

Existen dos métodos para asignar el ancho del *buffer* (ver Fig 2):

- El primer método utiliza una distancia fija de *buffer* para todos los puntos, en una capa esta distancia generalmente es proporcionada por el usuario.
- El segundo asigna a cada punto una distancia *buffer* individual basada en los atributos presentes en otra capa del sistema (pesos definidos). Los atributos de distancia se presentan en una tabla de atributos o tabla de búsqueda.
- Si hay múltiples puntos en la capa a la que se le aplicará el *buffer*, entonces se deben verificar las sobreposiciones que se produzcan. Este proceso implica dos operaciones adicionales intersección y disolución.



Fig. 2 Operación Intersección (ESRI, 2012)

Se realiza la operación de intersección para luego aplicarle la disolución, la cual genera un cuerpo sólido con las líneas que forman el *buffer* (ver Fig 3) (Rivera, 2004).



Fig. 3 Operación de disolución (ESRI, 2012)

Datos espaciales tipo línea (ver Fig 4):



Fig. 4 Entrada de datos tipo línea (Rivera, 2004)

Se asignan a cada segmento de línea el ancho de *buffer* apropiado (fijo o por pesos), este *buffer* se denominará distancia b y cada segmento de línea es procesado individualmente, obteniendo 3 segmentos de línea, en donde cada segmento de línea tiene un nodo inicial (E_1, N_1) y un nodo final (E_2, N_2) como se muestra en la Fig 5 (Rivera, 2004).

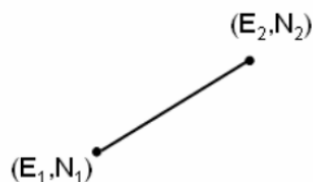


Fig. 5 Nodo inicial y final (Rivera, 2004)

A partir de estas coordenadas se calculan los valores Δx y Δy entre los dos extremos (ver Fig 6).

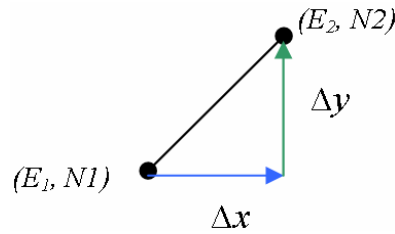


Fig. 6 Cálculo de Δx y Δy (Rivera, 2004)

Los puntos finales a una distancia perpendicular b se puede determinar a través de las siguientes ecuaciones:

$$(E1, N1) \begin{cases} E1 \pm b * \sin[\tan^{-1}(\Delta x \div \Delta y)] \\ N1 \pm b * \cos[\tan^{-1}(\Delta x \div \Delta y)] \end{cases}$$

$$(E2, N2) \begin{cases} E2 \pm b * \sin[\tan^{-1}(\Delta x \div \Delta y)] \\ N2 \pm b * \cos[\tan^{-1}(\Delta x \div \Delta y)] \end{cases}$$

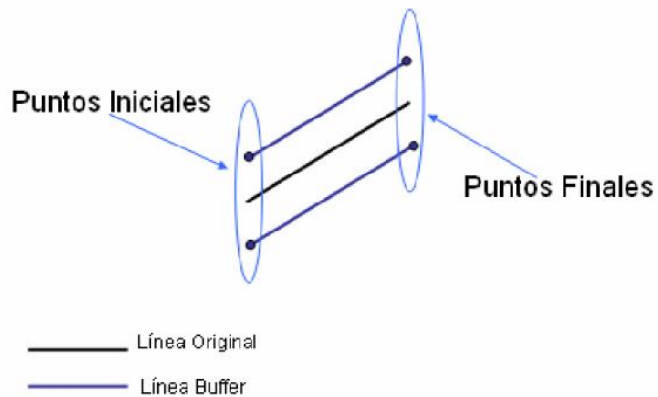


Fig. 7 Resultado de las ecuaciones expuestas anteriormente (Rivera, 2004)

En este punto el sistema examina el valor de la tangente al segmento de línea, para determinar si se producen incrementos crecientes o decrecientes al este o al norte dependiendo de la dirección de la línea. Una vez que las dos nuevas líneas paralelas de *buffer*, han sido identificadas, el siguiente segmento de línea puede ser procesado (ver Fig 7) (Rivera, 2004).

Posteriormente que se han identificado las líneas de *buffer* paralelas para el siguiente segmento, las intersecciones de las líneas paralelas de *buffer* de cada segmento son calculadas y las nuevas coordenadas

son asignadas a los vértices comunes. La prueba más sencilla para intersección de líneas puede ser utilizada en este punto. Estos pasos son repetidos hasta que el último segmento de línea haya sido procesado (ver Fig 8) (Rivera, 2004).

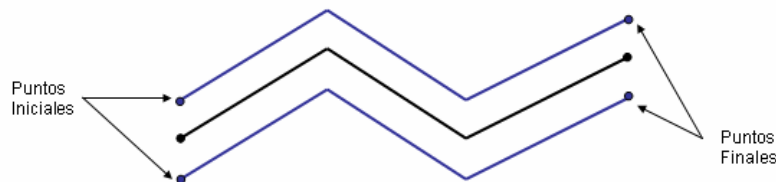


Fig. 8 Análisis completo de las líneas del *buffer* (Rivera, 2004)

El último paso consiste en la definición de la forma de los extremos del área de influencia, que consta de varios métodos son:

- Un sencillo truncamiento de los extremos finales de las líneas paralelas.
- Adelgazamiento en la finalización del *buffer* paralelo para encontrar los puntos finales.
- Envolverte consiste en encerrar el punto de inicio y el punto final de la línea con un semicírculo o un *buffer* de radio b .

El método de truncamiento consiste en trazar una línea recta desde el punto final de la línea *buffer* hacia el punto final de la línea original. No es el más utilizado para el análisis de *buffer* porque presenta como desventaja que en el punto inicial la distancia *buffer* no es óptima Fig 9 (Rivera, 2004).

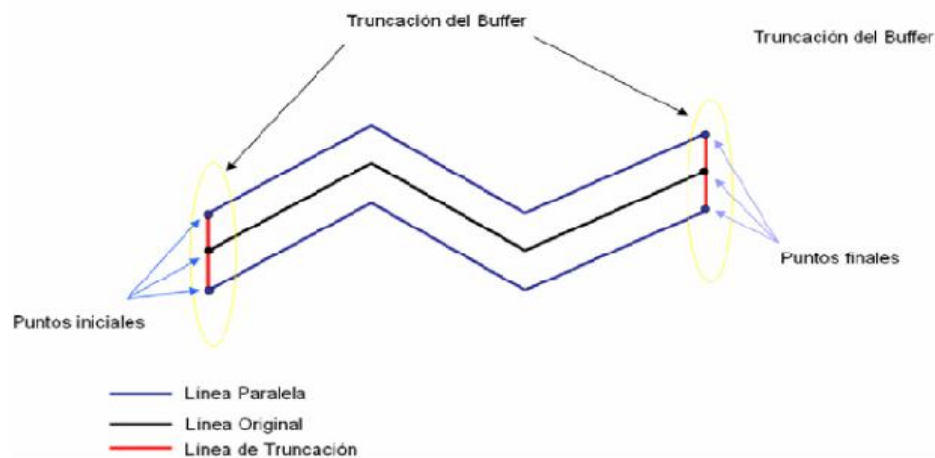


Fig. 9 Método de truncamiento (Rivera, 2004)

El método de adelgazamiento consiste en decrementar la distancia *buffer* que existe entre las líneas paralelas y la línea original, hasta que la distancia *buffer* se vuelve cero para ambas paralelas. Este método de terminación no es popular debido a que se debe establecer el grado de decremento para finalizarlo y al igual que el anterior la distancia *buffer* para el punto inicial y final no es óptima Fig 10 (Rivera, 2004).

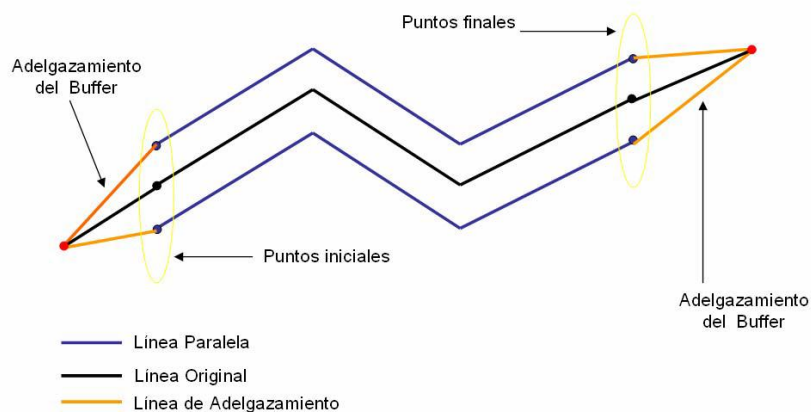


Fig. 10 Método de adelgazamiento (Rivera, 2004)

El método de envoltura consiste en encerrar el punto de inicio y el punto final de la línea original con un semicírculo de radio *buffer* b . Este método es el más óptimo ver Fig 11 (Rivera, 2004).

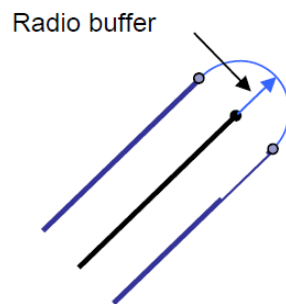


Fig. 11 Método de envolvente (Rivera, 2004)

Concluido la determinación del *buffer*, se reordena la nueva línea de segmentos generada para construir un polígono cerrado (Rivera, 2004).

Datos espaciales tipo polígono (ver Fig 12):



Fig. 12 Entrada de datos tipo polígono (Rivera, 2004)

El algoritmo para polígonos utiliza el mismo proceso que el *buffer* para datos de tipo línea, la diferencia consiste en que el *buffer* de polígono es generado sobre un solo lado de la línea que define a la misma entidad espacial (polígono).

El método por omisión es generar un *buffer*, el cual rodee los límites del polígono Fig 13.

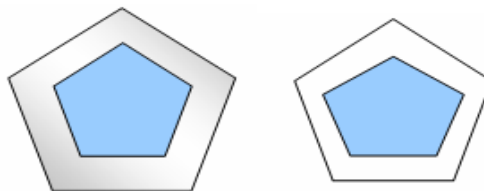


Fig. 13 *Buffer* externo e interno respectivamente (Rivera, 2004)

1.2.6.2 Algoritmos de cálculo de contornos mínimos

Graham Scan: Es relativamente sencillo y es una buena opción para puntos en el plano, pero tiene el inconveniente de que no puede ser generalizado a más de una dimensión. Este algoritmo utiliza una lista en la

que va almacenando y ordenando correctamente los puntos que constituyen los extremos del cierre convexo. El *Graham Scan* calcula la envolvente convexa en tiempo $O(n \log n)$ (Steven, 2011).

QuickHull: La idea es, ir descartando lo más pronto posible los puntos que no formarán parte de la frontera del cierre convexo, que suelen ser los más interiores de la nube de puntos. El algoritmo *Quick Hull* calcula la envolvente convexa en tiempo $O(n^2)$ (Steven, 2011).

Divide y vencerás: Consiste en ordenar los puntos según la coordenada x. Dividir los puntos en dos bloques, izquierda y derecha (L y R), de igual número de elementos, que cumplan que el punto más a la derecha de L está más a la izquierda que el punto más a la izquierda de R. Recursivamente, encontrar el *convex hull* de L y R. Para mezclar el cierre de L y el de R es necesario unirlos utilizando las tangentes comunes más altas y bajas. La tangente superior común puede descubrirse en tiempo lineal explorando alrededor del cierre de L en el sentido de las agujas del reloj y en el cierre de R en el sentido anti horario. Para la tangente inferior se gira en los sentidos inversos. Las líneas que queden dentro de la envolvente que se forma al unir los cierres de L y R serán borradas. Debido a que la mezcla no puede realizarse en tiempo lineal el cierre convexo puede ser encontrado en tiempo $O(n \log n)$. Este algoritmo también puede ser usado para calcular la intersección de los n semiplanos (Steven, 2011).

Tabla. 1 Comparación entre los algoritmos de contornos mínimos

Algoritmos	Ventajas	Desventajas	Tiempo de respuesta
<i>Graham Scan</i>	<ul style="list-style-type: none"> ➤ Relativamente sencillo. ➤ Buena opción para puntos en el plano. 	<ul style="list-style-type: none"> ➤ No puede ser generalizado a dimensiones más altas. 	<ul style="list-style-type: none"> ➤ $O(n \log n)$.
<i>QuickHull</i>	<ul style="list-style-type: none"> ➤ <i>QuickHull</i> es generalmente el más utilizado, para tercera dimensión. ➤ Debido a su naturaleza recursiva permite una implementación rápida y sin embargo limpia. 	<ul style="list-style-type: none"> ➤ No puede realizarse en tiempo lineal el cierre convexo. 	<ul style="list-style-type: none"> ➤ $O(n^2)$.
Divide y vencerás	<ul style="list-style-type: none"> ➤ Con la recursividad el problema original se reduce a sub-problemas que por lo 	<ul style="list-style-type: none"> ➤ No puede realizarse en tiempo lineal el cierre convexo. 	<ul style="list-style-type: none"> ➤ $O(n \log n)$.

	general son bastante fácil de resolver (O'Rourke, 2000).		
--	--	--	--

1.2.6.3 Algoritmos de cálculo de operaciones de solape

Sutherland-Hodgman: Este algoritmo se basa en recortar el polígono parcialmente usando cada lado del rectángulo vertical de recorte hasta terminar con la intersección final, interior al rectángulo. El polígono a recortar se representa como una lista de vértices: $v_1, v_2, v_3, \dots, v_n$. Cada recorte del polígono generará una nueva lista de vértices que representará el nuevo polígono resultante del recorte.

Weiler-Atherton: Es el más general de los algoritmos de recorte, permite producir polígonos con aristas coincidentes a partir de polígonos no convexos. *Weiler-Atherton* produce polígonos separados para cada fragmento visible (Steven, 2011).

Nicholl-Lee-Nicholl: Es un algoritmo que elimina la necesidad de realizar todos los cálculos de intersecciones favoreciendo las comprobaciones para dar con la intersección correcta, si existe. Las nuevas divisiones se basan en crear vectores desde el punto conocido, P, a cada uno de los vértices del rectángulo de recorte. Al determinar si el punto extremo, Q, está a la izquierda de uno de estos vectores, de P a un vértice, se puede reducir las posibles regiones hasta encontrar la que contiene el punto, Q. Una vez que se haya encontrado el punto, Q, y la región en la que se encuentra, se puede aplicar la ecuación de la intersección (Steven, 2011).

Tabla.2 Comparación entre los algoritmos de solape

Algoritmos	Ventajas	Desventajas
<i>Sutherland-Hodgman</i>	<ul style="list-style-type: none"> ➤ Fácilmente generalizable a 3D (Gutiérrez, y otros, 2014). ➤ Sirve para cualquier ventana poligonal convexa, no sólo ventanas rectangulares (Gutiérrez, y otros, 2014). 	<ul style="list-style-type: none"> ➤ Recorta el polígono parcialmente.

Weiler-Atherton	<ul style="list-style-type: none"> ➤ Sirve para recortar polígonos convexos o cóncavos en cualquier polígono de recorte, acepta polígonos con agujeros. 	<ul style="list-style-type: none"> ➤ El algoritmo <i>Weiler-Atherton</i> es un algoritmo de recorte 2D (Bradley University, 2009).
Nicholl-Lee-Nicholl	<ul style="list-style-type: none"> ➤ Algoritmo que elimina la necesidad de realizar todos estos cálculos de intersecciones favoreciendo las comprobaciones para dar con la intersección correcta, si existe. 	<ul style="list-style-type: none"> ➤ Algoritmo para recortar líneas.

1.2.7 *Plugin*

“Es un programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal). No se trata de un parche ni de una actualización, es un módulo aparte que se incluye opcionalmente en una aplicación” (ALEGSA, 2014).

“Es un programa que interactúa con otro programa para aportarle una función o utilidad específica, generalmente muy específica. Este programa adicional es ejecutado por la aplicación principal” (Levis, 2012).

“Un plugin es aquella aplicación que en un programa informático, añade una funcionalidad adicional o una nueva característica al software. En nuestro idioma, por lo tanto, puede nombrarse al plugin como un complemento” (Definición.de, 2013).

Un sistema capaz de añadir nuevas características o permitir la agregación de nuevas funcionalidades, a un sistema mayor implementado previamente, se puede definir como un *plugin*.

1.3 **Análisis de soluciones existentes**

Con el avance de los SIG surgieron bibliotecas capaces de transformar los datos geoespaciales y realizar operaciones geométricas a través de múltiples algoritmos. En este epígrafe se plasma el análisis realizado a varias bibliotecas de este ámbito, indagando más sobre GDAL, Sextante y PostGIS.

1.3.1 GDAL

Geospatial Data Abstraction Library(GDAL) es una biblioteca de *software* para la lectura y escritura de Formatos de datos geoespaciales, publicada bajo la MIT³*License* por la Fundación Geoespacial de Código Abierto (*Open Source Geospatial Foundation*). Como biblioteca, presenta un único modelo abstracto de datos de uso para todos los formatos soportados y es decir, es multiplataforma. También viene con una variedad de utilidades en línea de comando para la traducción y el proceso de datos geoespaciales. Utiliza el *software MapServer* (Instituto de estadística y cartografía de Andalucía, 2011).

GDAL provee una serie de métodos y rutinas a los desarrolladores de *software*, para la manipulación y el procesamiento de datos geoespaciales, como por ejemplo conversión de datos en formato *raster*, reproyección de imágenes, redimensionamiento de imágenes, conversión de datos en formato vectorial, reproyección de datos vectoriales, entre otros (GDAL, 2013).

GDAL es un proyecto que contiene un subproyecto conocido como OGR (*OGR Simple Features Library*). OGR es una biblioteca de acceso y un conjunto de utilidades de línea de comandos para acceso de lectura (y en algunas ocasiones de escritura) a formatos de ficheros vectoriales.

Posee diversas funciones para el cálculo de operaciones geométricas que se enmarcan en la clase *OGRGeometry* como son (GDAL, 2014):

- *OGRGeometry::Buffer(doble dfDist, intrnQuadSegs=30)*: Construye una nueva geometría que contiene la región del *buffer* alrededor de la geometría que se invoca. Parámetros *dfDist*: Distancia del *buffer* que será aplicada, debe ser expresado con la misma unidad de las coordenadas de la geometría *nQuadSegs* número de segmentos utilizados para aproximarse a 90 grados (cuadrante) de curvatura. Devuelve una nueva geometría o *null* si ocurre un error.
- *OGRGeometry::ConvexHull()*: Crea un nuevo objeto geométrico y devuelve la envolvente convexa de la geometría que invoca el método o *null* en caso de error.

Además de contener funciones para la realización de operaciones geométricas, posee como desventaja que toda la información de las tablas asociadas a los *Shapes* (Ficheros DBF) se pierde ya que solo se transforma la geometría.

³MIT: El texto de la licencia no tiene copyright, lo que permite su modificación. La licencia MIT es muy parecida a la licencia BSD (*Berkeley Software Distribution*) en cuanto a efectos.

1.3.2 Sextante

Sextante es un marco de procesamiento de datos espaciales que ofrece capacidades avanzadas de análisis en los SIG. Permite ejecutar un gran número de algoritmos de análisis de la interfaz de un SIG, y se integra perfectamente con las capacidades básicas para los datos de entrada y salida y la representación. Además de los algoritmos nativos, lo apoyan aplicaciones externas como GRASS, Saga u Orfeo y Caja de herramientas. Sextante está programado en Java.

Contiene varias herramientas de productividad como un modelador gráfico, una interfaz de línea de comandos o una interfaz de procesamiento por lotes, lo que añade más poder a todos los algoritmos disponibles. Una simple pero potente API⁴ está disponible para los desarrolladores, por lo que es también fácil de implementar nuevos *geoalgorithms* o incluir soporte para nuevas aplicaciones externas. SEXTANTE es *software* libre y se distribuye bajo la licencia *General Public License* GPL⁵.

Algunas de sus utilidades son:

- Estadísticas de celda para múltiples capas *raster*.
- Herramientas de análisis para capas *raster*.
- Herramientas de cálculo para capas *raster*.
- Herramientas para capas de líneas.
- Herramientas para capas de puntos.
- Herramientas para capas vectoriales.
- Herramientas para crear nuevas capas *raster*.
- Tratamiento y análisis de imágenes.

Sextante permite a través de su consola a los usuarios más avanzados hacer un uso más ágil del programa y automatizar tareas mediante la creación de scripts (Equipo SEXTANTE, 2008).

En la actualidad, más de 300 algoritmos están disponibles en SEXTANTE que permiten el análisis de datos sobre diversos temas geográficos (Cartografía.cl, 2005). Esta biblioteca se enmarca en los SIG de escritorio (García, 2009).

4 API: Interfaz de programación de aplicaciones.

5 GPL: Licencia Pública General de GNU.

1.3.3 PostGIS

Es una extensión al sistema de base de datos objeto-relacional PostgreSQL, permite el uso de objetos SIG. PostGIS incluye soporte para índices GiST basados en *R-Tree*, y funciones básicas para el análisis de objetos SIG. Está creado por *Refractions Research Inc*, como un proyecto de investigación de tecnologías de bases de datos espaciales. Está publicado bajo licencia GNU. Con PostGIS se pueden usar todos los objetos espaciales como puntos, líneas, polígonos, multilíneas, multipuntos, y colecciones geométricas (PostGIS, 2012).

PostGIS implementa algoritmos de cálculo geométrico que brindan la posibilidad de realizar operaciones de solape (ver Fig 14). La operación unión de dos argumentos a través del siguiente método `ST_Union(A, B)`, devuelve un resultado en el cual no existen geometrías interiores que se superponen. La intersección superpone los dos puntos, los que coincidan serán el resultado de dicha intersección que se realiza con el método `ST_Intersection(A, B)`; la diferencia se desarrolla del mismo modo que la intersección `ST_Difference(A,B)` y `ST_Buffer(A, B, C)` que determina las zonas de influencia deseadas (PostGIS, 2010).

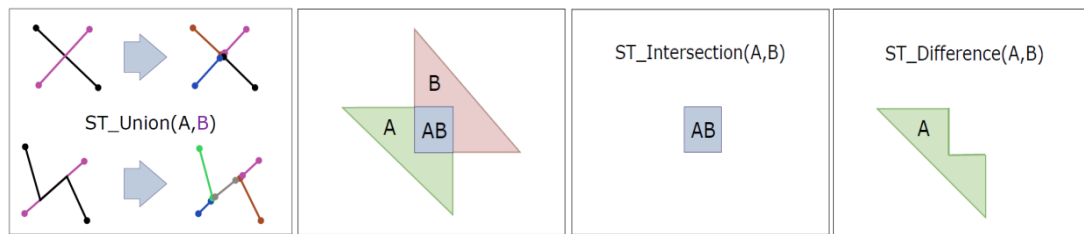


Fig. 14 Operaciones de solape en PostGIS (PostGIS, 2010)

PostGIS es una extensión que convierte el sistema de base de datos PostgreSQL en una base de datos espacial. La combinación de ambos es una solución perfecta para el almacenamiento, gestión y mantenimiento de datos espaciales (Mappigis.com, 2012).

Características:

- PostGIS es *software* libre, tiene licencia GPL.
- Es compatible con los estándares de OGC⁶.
- Soporta tipos de datos espaciales, índices espaciales y tiene cientos de funciones espaciales.

⁶Open Geospatial Consortium (OGC): Su fin es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica y de la World Wide Web.

- Permite importar y exportar datos a través de varias herramientas de conversión (**shp2pgsql**, **pgsql2shp**, **ogr2ogr**, **dxf2postgis**).

1.3.4 Observación de las soluciones existentes

Luego de analizar las ventajas y desventajas que presentan las bibliotecas GDAL/OGR, Sextante y PostGIS, se concluye que PostGIS es la biblioteca adecuada para la realización del módulo. Además de ser la biblioteca utilizada en el desarrollo de la plataforma GeneSIG, se le suma la excelente integración con PostgreSQL que no solo le permite almacenar datos geoespaciales, sino también realizar un análisis profundo sobre estos datos.

1.4 Metodología, Herramientas, Tecnología y Lenguajes de desarrollo

En el proceso de desarrollo del *software* es importante definir la metodología, tecnologías, herramientas y lenguajes de desarrollo, para obtener el producto final. Al tener la plataforma GeneSIG una arquitectura estable y definida la selección de las tecnologías y herramientas a utilizar para la construcción del módulo queda fuera del alcance de la presente investigación.

1.4.1 Metodología de desarrollo

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software y constan de (Romero, 2009):

- Una filosofía de desarrollo del software con una base de procesos de desarrollo de software.
- Múltiples herramientas, modelos y métodos para asistir al proceso de desarrollo del software.
- Suele estar documentada formalmente.

El Proceso Unificado de Rational (*RUP- Rational Unified Process*) es un ejemplo de un modelo de proceso moderno que proviene del trabajo en el UML y el asociado Proceso Unificado de Desarrollo de Software. Reúne elementos de todos los modelos de procesos genéricos, iteraciones de apoyo e ilustra buenas prácticas en la especificación y el diseño. (Sommerville, 2005).

1.4.2 Lenguaje de Modelado Unificado

El Lenguaje de Modelado Unificado (UML –*Unified Modeling Language*) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de *software*. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio, funciones de sistema y los datos de la

estructura, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes del *software* (Sitio Web Oficial de Lenguaje Unificado de Modelado, 2013).

UML ofrece diversos elementos de esquematización que representan las diferentes partes de un sistema, de los cuales se utilizan en el desarrollo del módulo.

1.4.3 Herramienta Case

Una herramienta Case (Ingeniería del Software Asistida por Computadoras) es un sistema que intenta proporcionar ayuda automatizada a las actividades del proceso de software como el análisis de requisitos, el modelado de sistemas, la depuración y las pruebas (Sommerville, 2005). Entre las herramientas utilizadas actualmente se pueden mencionar AllFusion ERWin, ArgoUML, Rational Rose, Visual Paradigm, entre otras, siendo el Visual Paradigm la seleccionada para crear los artefactos generados y ser el instrumento de modelado concebido para el ciclo completo (análisis y diseño, construcción, pruebas y despliegue) del proceso de desarrollo del módulo.

Visual Paradigm for UML v8.0

Es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para la construcción de sistemas a gran escala que necesiten confiabilidad y estabilidad en el desarrollo orientado a objetos (Targetware Informática S.A.C, 2013).

Visual Paradigm también ofrece:

- Navegación intuitiva entre la escritura del código y su visualización.
- Ingeniería inversa de bases de datos desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Distribución automática de diagramas – Reorganización de las figuras y conectores de los diagramas UML (Free Download Manager, 2007).
- Posee licencia gratuita.

1.4.4 Lenguaje de desarrollo

PHP 5

Es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la web. El código PHP es interpretado en el servidor web y genera código HTML y otro contenido que el visitante verá. Es un producto de código abierto, lo que quiere decir que permite acceder a su código, utilizarlo, modificarlo y redistribuirlo. Las

siglas PHP equivalían inicialmente a *Personal Home Page* (Página de inicio personal) pero se modificaron de acuerdo con la convención de designación de GNU y ahora es equivalente a PHP (*Hipertext PreProcessor*) (PHP, 2013).

JavaScript

Es un lenguaje que no requiere de compilación ya que trabaja del lado del cliente. Los navegadores son los encargados de interpretar el código, por lo que no se requiere tener instalado ningún *framework*. Principalmente es usado para la incorporación de comportamiento dinámico a documentos HTML (describir objetos, escribir funciones que respondan a movimientos del *mouse*, aperturas, utilización de teclas, cargas de páginas) (GOODMAN, 2001).

HTML

HTML es la abreviatura de *HyperText Markup Language*, y es el lenguaje que todos los programas navegadores usan para presentar información en la *World Wide Web* (WWW) (Grupo EIDOS, 2000).

Características de este lenguaje.

- No es necesario ningún programa especial para crear una página Web. Gracias a ello se ha conseguido que se puedan crear páginas con cualquier ordenador y sistema operativo.
- Es un lenguaje descriptivo.
- Describe hipertexto, texto de forma estructurada y agradable.
- Permite inserciones multimedia.

Hojas de Estilo en Cascada (CSS⁷)

CSS es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Permite controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a ella en la que aparezca ese elemento. CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne (W3C, 2014).

⁷ CSS: *Cascading Style Sheets*.

1.4.5 Herramientas de desarrollo

MapServer v6.0

Es una plataforma de código abierto (*Open Source*) para la publicación de datos espaciales y aplicaciones cartográficas interactivas en la web. Se distribuye bajo la licencia similar MIT *License*. Es un programa CGI⁸ escrito en C, que utiliza la información pasada en una petición de usuario en la URL y un archivo *Mapfile* para crear una imagen del mapa requerido, además la petición podría incluir imágenes para leyendas, barras de escala, mapas de referencia y valores pasados como variables CGI⁷.

PostgreSQL v9.1

Es un sistema de gestión de bases de datos objeto-relacional de código abierto, distribuido bajo la licencia BSD (*Berkeley Software Distribution*), y su código fuente está disponible. Utiliza un modelo cliente/servidor y usa multiprocesadores en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (UCI, 2013).

- **Atomicidad (Indivisible):** es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- **Consistencia:** es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- **Aislamiento:** es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.
- **Durabilidad:** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

PostGIS v1.5

El módulo PostGIS es una extensión para el servidor PostgreSQL. Como tal, PostGIS 1.5 requiere servidor PostgreSQL completo con el fin de compilar. Es compatible con versiones de PostgreSQL 8.3 en adelante. Las versiones anteriores de PostgreSQL no son soportadas (PostGIS, 2008). Permite usar todos los objetos

⁸**CGI:** Es una tecnología de *la World Wide Web* que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.

geométricos como puntos, líneas, polígonos, multilíneas, multipuntos, y colecciones geométricas (PostGIS, 2012). Posee la función *St_buffer* que complementará la realización del módulo.

Apache v2.2

Es un servidor web potente y flexible, con código de libre distribución, introducido en 1995 por el Grupo Apache como una extensión y mejora del HTTP del Centro Nacional de Aplicaciones para Supercomputadoras. El servidor Apache es popular en los sistemas basados en UNIX, incluyendo Linux, y también corre en Windows NT y otros sistemas operativos (Apache Tomcat, 2013).

Posee las siguientes particulares:

- Autenticación de acceso básico.
- Protocolo de transferencia de hipertexto seguro (HTTPS).
- CGI o interfaz de entrada común.
- Servlets de Java.
- *Server-Side Includes* (SSI).

Netbeans v7.2

Es un IDE⁹ de código abierto cuyo principal enfoque es el desarrollo de aplicaciones *desktop*, web, empresariales y para dispositivos móviles. Está escrito en el lenguaje de programación JAVA y es multiplataforma. NetBeans es libre tanto para el uso comercial como no comercial (Abela, 2010).

El editor de código fuente que presenta en estos momentos es más ágil y a la vez robusto e integra conjuntamente lenguajes como HTML, JavaScript y CSS, características que lo convierten en una excelente herramienta para el desarrollo de soluciones informáticas. Se ejecuta en Windows, Linux, Mac OS X y Solaris. Permite la ingeniería inversa, así como la integración con diferentes marcos de trabajo.

Ext JS v3.3

ExtJS es una biblioteca de JavaScript que simplifica JavaScript + XML (Ajax). Ofrece un gran conjunto de widgets (componentes como ventanas de dialogo, grids) plenamente integrados y una API para conseguir interfaces web más dinámicas e interactivas con el usuario (Lozano, 2010). Permite realizar completas interfaces de usuario, fáciles de usar, muy parecidas a las conocidas aplicaciones de escritorio. Esto permite a los desarrolladores web

⁹Entorno de Desarrollo Integrado (IDE): Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

concentrarse en la funcionalidad. Debido a sus características para la realización de interfaces sencillas y amigables, la herramienta será utilizada durante el proceso de implementación del módulo.

1.5 Conclusiones

Partiendo del estudio y análisis de las diversas bibliografías consultadas se logró identificar algoritmos como el *Graham Scan*, *QuickHull* y *Divide y Vencerás* para determinar la envolvente mínima; *Sutherland-Hodgman*, *Weiler-Atherton* y *Nicholl-Lee-Nicholl* para las operaciones de solape. No sucediendo lo mismo para el cálculo de zonas de influencia que se determina según el objeto espacial sobre al cual se desea aplicar la operación.

A partir de la comparación de las características, potencialidades y desventajas que poseen las bibliotecas analizadas y el tratamiento que le dan a las diversas funciones relacionada con la investigación como son *buffer*, operaciones de solape y contornos mínimo se decide utilizar las soportadas por PostGIS; debido a su compatibilidad con PostgreSQL y su amplia gama de características descritas durante el desarrollo del presente capítulo.

La selección de la metodología, herramientas y lenguajes adecuados para el desarrollo del módulo destinado a complementar las funcionalidades de la plataforma GeneSIG, se fundamentan a partir de las herramientas que se utilizan en el desarrollo e implementación de la plataforma.

Capítulo.2 Análisis y Diseño

2.1 Introducción

En el presente capítulo se describe la solución propuesta para el desarrollo del módulo de operaciones geométricas, el cual proporciona nuevas funcionalidades a la plataforma GeneSIG; convirtiéndola en una herramienta capacitada para la realización de operaciones geométricas sobre capas vectoriales, contribuyendo al proceso de toma de decisiones. Para ello es necesario la clasificación y especificación de los requisitos tanto funcionales como no funcionales, además de la elaboración del diagrama de modelo de dominio y la clasificación de los casos de uso del sistema.

2.2 Propuesta de solución

El módulo de operaciones geométricas debe desarrollarse sobre la plataforma web GeneSIG, su desarrollo permite determinar zonas de influencia, además de almacenar la geometría obtenida luego de aplicarle las operaciones de *buffer* para futuros análisis.

El módulo le ofrece al usuario la posibilidad de seleccionar las capas sobre las cuales desea trabajar, así como el tamaño del radio del *buffer* para las zonas de influencia. También admite la visualización de las capas resultantes de determinación de las zonas de influencia.

2.3 Modelo del dominio

Para un mejor entendimiento del problema a resolver se muestra a continuación el modelo del dominio (Fig 15), su realización se basa en que gran parte de la información utilizada ha sido obtenida mediante el estudio de sistemas similares debido a que no se cuenta con expertos en el tema.

El modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de *software*.

Es un diagrama de clases en el que se muestran:

- Objetos del dominio o clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

2.3.1 Diagrama de clases del dominio

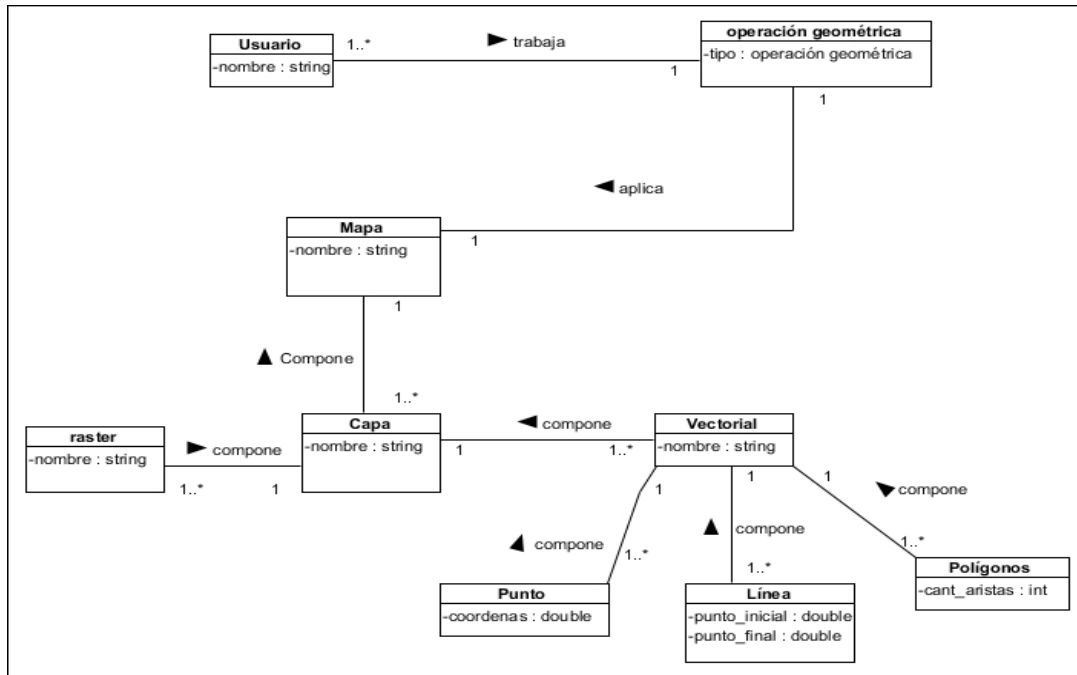


Fig. 15 Diagrama de clases del dominio

2.3.2 Breve descripción del modelo de dominio

A partir de la plataforma GeneSIG se desarrollan un conjunto de personalizaciones SIG en dependencia de las características del cliente. Uno de los módulos que contribuye a una mejor toma de decisiones es el de operaciones geométricas. Un usuario puede utilizar estas operaciones geométricas sobre una o varias capas del mapa, estas capas pueden ser *raster* o *vectorial*. Las vectoriales se conforman por varios objetos espaciales que pueden ser: puntos, líneas, polígonos o la combinación de ellos.

2.4 Técnicas de extracción de requisitos

La comprensión de los requisitos de un problema se encuentra entre las tareas más difíciles durante el proceso de desarrollo del software. Para contrarrestar los niveles de dificultad que presenta esta etapa existen un grupo de técnicas o métodos que posibilitan la captura de los requisitos. La entrevista, introspección, cuestionarios, listas de verificación, tormenta de ideas y análisis de la documentación son técnicas que proporcionan la obtención de los mismos.

Para realizar una buena construcción del software se obtuvieron los requisitos mediante las técnicas: entrevista y tormenta de ideas. La entrevista permitió conocer e identificar las necesidades del cliente, que en este caso corresponde al líder del proyecto Aplicativos SIG; mientras que la tormenta de ideas tuvo como principio reunir a un grupo constituido por los principales roles en esta primera etapa de desarrollo, en un ambiente libre de críticas o juicios, generando una variedad de puntos de vista diferentes y así poder englobar todos los criterios para encontrar una solución factible al problema.

2.5 Requisitos funcionales

Un requisito es una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado. Se aplica también a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación. Define los servicios que el sistema debe proporcionar (Laguna, 2009).

RF1. Insertar zona de influencia

Esta funcionalidad permite al usuario insertar una zona de influencia a través de las variables de entrada:

- Nombre de la zona (Formato: Alfanumérico, Obligatorio: Sí).
- Radio (Formato: Numérico, Obligatorio: Sí).
- Capa (Formato: Lista desplegable, Obligatorio: Sí).
- Color (Formato: Lista desplegable, Obligatorio: Sí).

RF2. Modificar zona de influencia

Esta funcionalidad permite al usuario modificar la zona anteriormente insertada cuyas variables de entrada serán iguales al requisito anterior.

- Nombre de la zona (Formato: Alfanumérico, Obligatorio: Sí).
- Radio (Formato: numérico, Obligatorio: Sí).
- Capa (Formato: Lista desplegable, Obligatorio: Sí).
- Color (Formato: Lista desplegable, Obligatorio: Sí).

RF3. Eliminar zona de influencia

Esta funcionalidad permite al usuario eliminar la o las zonas de influencia creadas.

RF4. Visualizar zona de influencia

Esta funcionalidad permite al usuario visualizar la zona de influencia seleccionada.

RF5. Insertar condición

Esta funcionalidad permite al usuario insertar las condiciones deseadas para el determinar la zona de influencia a través de los criterios:

- Capa (Formato: Lista desplegable, Obligatorio: Sí).
- Condición (Formato: Lista desplegable, Obligatorio: Sí).
- Cantidad (Formato: Numérico, Obligatorio: Sí).

RF6. Modificar condición

Esta funcionalidad permite al usuario modificar las condiciones creadas cuyas variables de entrada serán las iguales a las del requisito anterior.

- Capa (Formato: Lista desplegable, Obligatorio: Sí).
- Condición (Formato: Lista desplegable, Obligatorio: Sí).
- Cantidad (Formato: Numérico, Obligatorio: Sí).

RF7. Eliminar condición

Esta funcionalidad permite al usuario eliminar las condiciones anteriormente insertadas.

RF8. Listar zona de influencia

Esta funcionalidad permite al usuario obtener un listado con el nombre de la propiedad y la cantidad de condiciones correspondientes a una zona de influencia.

RF9. Listar condición

Esta funcionalidad permite al usuario obtener un listado de las condiciones insertadas.

2.6 Requisitos no funcionales

Los requisitos no funcionales definen propiedades emergentes del sistema, tales como el tiempo de respuesta, las necesidades de almacenamiento y la fiabilidad. Pueden ser más críticos que los funcionales, si un requisito funcional no se cumple, el sistema se degrada, si un requisito no funcional no se cumple, el sistema puede inutilizarse (Laguna, 2009).

Usabilidad

El sistema proporcionará una interfaz sencilla y atractiva, con facilidad de uso. Los grupos de botones y vínculos deben estar organizados por la funcionalidad, con el objetivo de facilitar al usuario la interacción con el sistema.

Confiabilidad

El módulo deberá ser estable y fiable. Deberá mostrar información confiable para evitar cualquier tipo de error para la toma de decisiones a través de los datos que proporciona el sistema para determinar la zona de influencia.

Restricciones de diseño.

El *software* deberá usar como lenguaje del lado del servidor, PHP y del lado del cliente, JavaScript.

Requerimientos de Hardware:

PC Cliente:

- Tarjeta de red
- Al menos 128 MB de memoria RAM.
- Procesador 512 MHz como mínimo.

Servidor:

- 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.

Requerimientos de Software:

PC Cliente:

- Navegador Web que cumpla con los estándares de la World Wide Web Consortium (W3C).
- Sistema operativo: GNU/Linux, Windows.

Servidor:

- Sistema operativo GNU/Linux Ubuntu Server 11.04 o superior.
- Servidor Web Apache 2.2 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL 9.1 o superior como Sistema Gestor de Base de Datos.
- PostGIS 1.5 o superior como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 6.0 o superior, con extensión PHP mapscript.

2.7 Diagrama de caso de uso

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de usos representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema (Jacobson, y otros, 2000)

A continuación se muestra el Modelo de casos de uso del sistema. Ver Fig 16

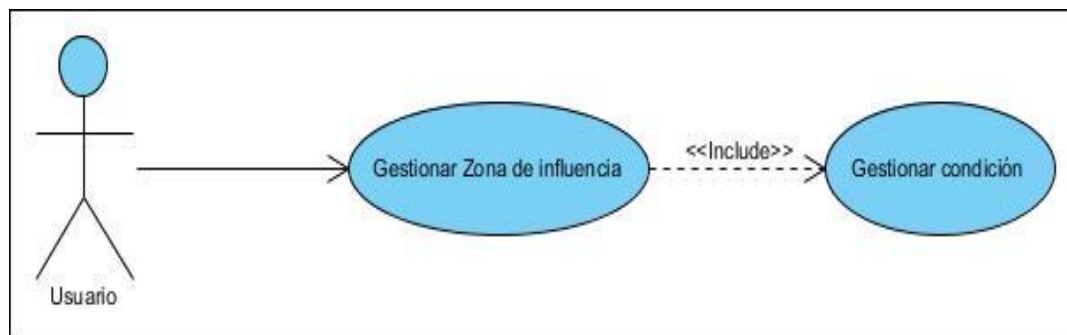


Fig. 16 Diagrama de casos de uso.

2.7.1 Actores del sistema

Tabla.3 Actores del sistema

Actor	Descripción
Especialista	Usuario con permisos para acceder a las funcionalidades del módulo.

2.7.2 Especificación de los casos de usos del sistema

Tabla.4 Descripción del caso de uso Gestionar zona de influencia

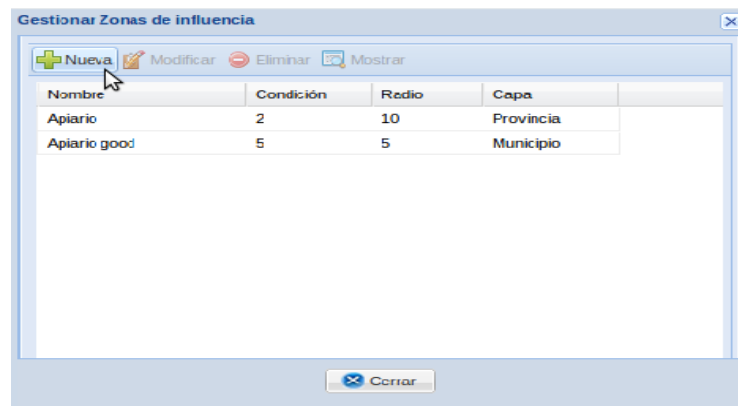
Caso de uso:	Gestionar zona de influencia.
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el usuario inserta las propiedades definidas para determinar la zona de influencia, una vez insertadas se obtendrá un listado con todos los datos de la propiedad la cual se podrá modificar y eliminar. Finalizando el caso de uso con la visualización de la zona de influencia.

Precondiciones:	-
Referencias:	RF1, RF2, RF3, RF4, RF8.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario accede al módulo <i>Buffer</i> . Ver Interfaz 1.	<p>2. El sistema muestra la ventana Gestionar Zona de influencia y las opciones Nueva, Eliminar, Modificar y Mostrar. Ver Interfaz 2.</p> <ul style="list-style-type: none"> - Si desea insertar un <i>buffer</i> ir a la sección "Insertar". - Si desea modificar los valores del <i>buffer</i> ir a la sección "Modificar". - Si desea visualizar un <i>buffer</i> ir a la sección "Mostrar". - Si desea eliminar un <i>buffer</i> ir a la sección "Eliminar".
Prototipo de Interfaz	

Interfaz 1



Interfaz 2



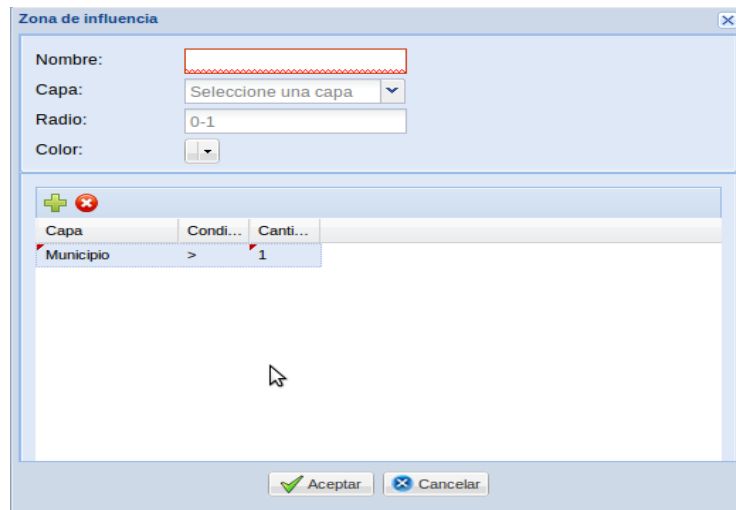
Sección "Insertar"

Acción del Actor	Respuesta del Sistema
3. El usuario da clic en el botón Nueva.	<p>4. El sistema muestra la ventana Zonas de influencia con los campos que el usuario debe llenar.</p> <p>Para las propiedades de la zona de influencia a crear:</p> <ul style="list-style-type: none"> - Nombre - Radio - Capa - Color <p>Para los criterios específicos de la zona de influencia a crear:</p> <ul style="list-style-type: none"> - Capa - Condición - Cantidad <p>Ver Interfaz 3</p>

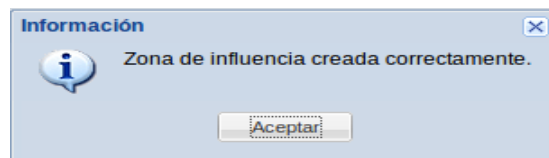
<p>5. El usuario inserta los datos para crear la zona de influencia y da clic en el botón Aceptar.</p>	<p>6. El sistema valida los datos, los almacena visualizándolos en la lista de <i>buffer</i> de la ventana Gestionar Zona de influencia y muestra el mensaje “Zona de influencia creada correctamente”. Ver Interfaz 4.</p>

Prototipo de Interfaz

Interfaz 3

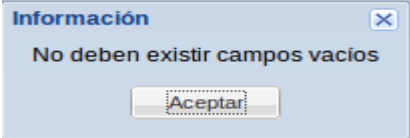


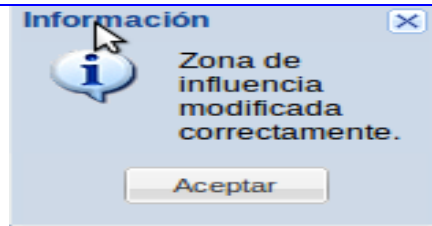
Interfaz 4



Flujos Alternos

Acción del Actor	Respuesta del Sistema
<p>5. El usuario deja campos vacíos para crear la zona de influencia y da clic en el botón Aceptar.</p>	<p>6. El sistema muestra el mensaje “No deben existir campos vacíos”. Ver Interfaz 5.</p>

5. El usuario inserta los datos para crear la zona de influencia y da clic en el botón Cancelar.	6. El sistema cierra la ventana.
Prototipo de interfaz	
<p>Interfaz 5</p> 	
Sección "Modificar"	
Acción del Actor	Respuesta del Sistema
3. El usuario selecciona de la lista de <i>buffer</i> el que desea modificar y da clic en el botón Modificar.	<p>4. El sistema muestra la ventana Zonas de influencia con los campos para que el usuario modifique los valores que desee.</p> <p>Para las propiedades de la zona de influencia:</p> <ul style="list-style-type: none"> - Nombre - Radio - Capa - Color <p>Para los criterios específicos de la zona de influencia:</p> <ul style="list-style-type: none"> - Capa - Condición - Cantidad <p>Ver Interfaz 3</p>
5. El usuario modifica los datos de la zona de influencia y da clic en el botón Modificar.	6. El sistema valida los datos, los actualiza mostrándolos en la lista de <i>buffer</i> y muestra el mensaje "Zona de influencia modificada correctamente". Ver Interfaz 6.
Prototipo de interfaz	
Interfaz 6	



Flujos Alternos

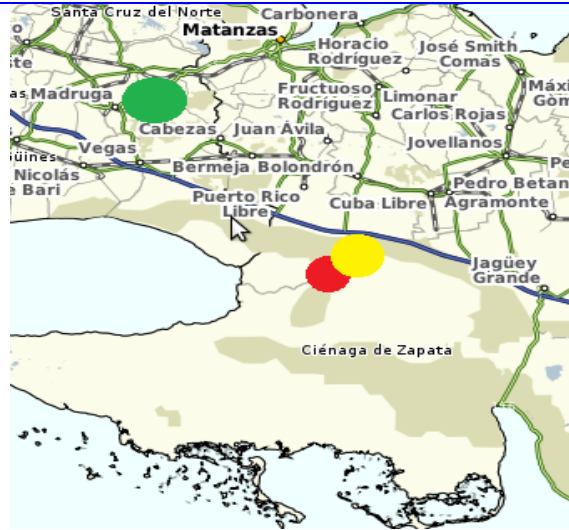
Acción del Actor	Respuesta del Sistema
5. El usuario deja campos vacíos y da clic en el botón Modificar.	6. El sistema muestra el mensaje “No deben existir campos vacíos”. Ver Interfaz 5.
5. El usuario selecciona de la lista de <i>buffer</i> el que desea modificar y da clic en el botón Cancelar.	6. El sistema cancela la selección realizada.
5. El usuario no selecciona de la lista de <i>buffer</i> el que desea modificar y da clic en el botón Modificar.	6. El sistema muestra el mensaje “Debe seleccionar un <i>buffer</i> ”.

Sección “Mostrar”

Acción del Actor	Acción del Actor
3. El usuario selecciona de la lista de <i>buffer</i> el que desea visualizar y da clic en el botón Mostrar.	4. El sistema cierra la ventana Gestionar Zona de influencia y muestra en el mapa el <i>buffer</i> seleccionado. Ver Interfaz 7.

Prototipo de interfaz

Interfaz 7



Flujos Alternos

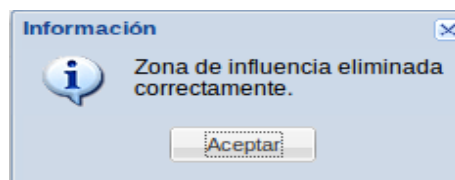
Acción del Actor	Respuesta del Sistema
3. El usuario no selecciona de la lista de <i>buffer</i> el que desea visualizar y da clic en el botón Mostrar.	4. El sistema muestra el mensaje “Debe seleccionar un <i>buffer</i> ”.
3. El usuario selecciona de la lista de <i>buffer</i> el que desea visualizar y da clic en el botón Cancelar.	4. El sistema cancela la selección realizada.

Sección “Eliminar”

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona de la lista de <i>buffer</i> el que desea eliminar y da clic en el botón Eliminar.	4. El sistema elimina el <i>buffer</i> seleccionado de la lista y muestra el mensaje “Zona de influencia eliminada correctamente”. Ver Interfaz 8

Prototipo de interfaz

Interfaz 8



Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El usuario no selecciona de la lista de <i>buffer</i> el que desea eliminar y da clic en el botón Eliminar.	4. El sistema muestra el mensaje “Debe seleccionar un <i>buffer</i> ”.
3. El usuario selecciona de la lista de <i>buffer</i> el que desea eliminar y da clic en el botón Cancelar.	4. El sistema cancela la selección realizada.
Poscondiciones	El sistema determina la zona de influencia para luego visualizarla

La descripción correspondiente al caso de uso Gestionar Condición se encuentra adjunto al documento “Modelo del sistema” perteneciente al expediente de proyecto.

Obtenidos los requisitos con los que contará el módulo se dará paso a determinar la estructura arquitectónica; que permitirá integrar los requisitos del sistema en aras de establecer un marco estructural básico que identificará los principales componentes y la relación entre ellos.

2.8 Patrón arquitectónico

Un patrón de arquitectura de software es un esquema genérico probado para solucionar un problema particular, el cual es recurrente dentro de un cierto contexto. Esta representación se especifica describiendo los componentes, con sus responsabilidades y relaciones.

El módulo se desarrollará siguiendo los principios de los patrones arquitectónicos basado en componentes y orientado a objetos como está establecido en la plataforma GeneSIG. La misma está creada sobre el framework CartoWeb que ha sido diseñado con una arquitectura bastante modular y escalable, permitiendo separar la lógica del servidor con respecto a la del cliente (CartoWeb, 2008).

Estilo Llamada y Retorno:

Arquitectura Basada en Componentes

“Define la composición de software como el proceso de construir aplicaciones mediante la interconexión de componentes de software a través de sus interfaces (de composición), abogaba por la utilización de componentes prefabricados sin tener que desarrollarlos de nuevo”(Robaina, 2008)

Tal es el caso del módulo que la arquitectura basada en componente le permite reutilizar código pre-elaborado para la realización de diversas tareas, además de que concede la posibilidad de adicionarle nuevas funcionalidades al módulo sin dañar las que ya posee.

Arquitectura Orientada a Objetos

Esta arquitectura permite modificar la implementación de un objeto sin afectar a sus clientes. De esta manera es posible descomponer problemas en colecciones de agentes en interacción. Además, un objeto es ante todo una entidad reutilizable en el entorno de desarrollo (Kicillof, 2004).

Se utiliza en la implementación del módulo, puesto que se basa que en principios Orientado a Objetos: encapsulamiento, herencia y polimorfismo, lo proporciona la facilidad de realizar llamadas a funcionalidades del objeto creado desde cualquier clase.

2.8.1 Patrones de diseño

Patrones GRASP

Los patrones de diseños GRASP son patrones generales de software para asignación de responsabilidades a objetos, es el acrónimo de "*General Responsibility Assignment Software Patterns*"; entre los que se encuentran: Experto, Creador, Bajo Acoplamiento, Alta Cohesión, Controlador.

Experto: Se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la tarea. En este caso, se tiene la clase *ClientBuffer.php*, que es experta en procesar los datos que son enviados a través del navegador web y la clase *ServerBuffer.php*, que es la encargada de la interacción con el servidor de mapas y el servidor de bases de datos. Su uso permite que se conserve el encapsulamiento, ya que los objetos se valen de su propia información para realizar las tareas asignadas. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas de fácil mantenimiento.

Bajo Acoplamiento: El acoplamiento es una medida de la fuerza en que una clase está conectada a otras clases, que las conoce y recurre a ellas. En este caso, se refleja el bajo acoplamiento, en cada una de las clases del módulo de operaciones geométricas, con el objetivo de que una clase no dependa de muchas clases, de esta forma, no se afectan las clases al ser sometidas a cambios de componentes, son fáciles de entender por separado y fáciles de reutilizar. No se puede considerar de manera aislada a otros patrones como el Experto o el de Alta Cohesión.

Controlador: Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. En este caso, se encuentra reflejado en la clase *plugin.ajax*, encargada del comportamiento del *plugin* y gestiona todos los eventos que ocurren en el mismo.

Creador: Se aplica para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello. Viéndose reflejadas en las clases *ClientBuffer.php* y *ServerBuffer.php*.

Alta Cohesión: La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Además, permite mejorar la claridad y la facilidad con que se entiende el diseño y genera a menudo un bajo acoplamiento.

Patrones GOF (Gang of Four)

Los patrones GOF resuelven problemas de diseño específicos y hacen el diseño flexible y reusable. Según su propósito se clasifican en (Gamma, y otros, 1995):

- **De creación:** conciernen al proceso de creación de objetos.
- **De estructura:** tratan la composición de clases y/o objetos.
- **De comportamiento:** caracterizan las formas en las que interactúan y reparten responsabilidades las distintas clases u objetos.

Singleton (solitario): Su propósito es garantizar que una clase sólo tenga una única instancia, proporcionando un punto de acceso global a la misma. Es fácilmente modificable para permitir más de una instancia y, en general, para controlar el número de las mismas, incluso si es variable; el acceso a la “Única Instancia” está más controlado (Facultad de informática - Universidad Politécnica de Madrid, 2010). Se refleja en la clase *ServerContext.php* donde se crea un única instancia de tipo *MapObj* para su utilización durante todo el período de ejecución del software.

Command (acción): Encapsula las peticiones en forma de objetos. Permite parametrizar a los clientes con distintas peticiones, hacer colas de peticiones, llevar un registro de las peticiones realizadas, y deshacer el efecto de las peticiones (Universidad de Arantxa, 2008). Este patrón se pone de manifiesto en la clase *Ajaxhelper.js* a la hora de comunicarse con el servidor web.

2.9 Diagrama de paquete

Los diagramas de paquetes muestran la organización de los paquetes lo que facilita el manejo de los modelos de un sistema complejo y los espacios de nombres, es decir, dos elementos pueden tener los mismos nombres, pero pertenecer a diferentes paquetes. Permiten dividir un modelo para agrupar y encapsular sus elementos en unidades lógicas individuales y pueden estar anidados unos dentro de otros.

A continuación se muestra el diagrama de paquetes correspondiente al módulo (ver Fig 17).

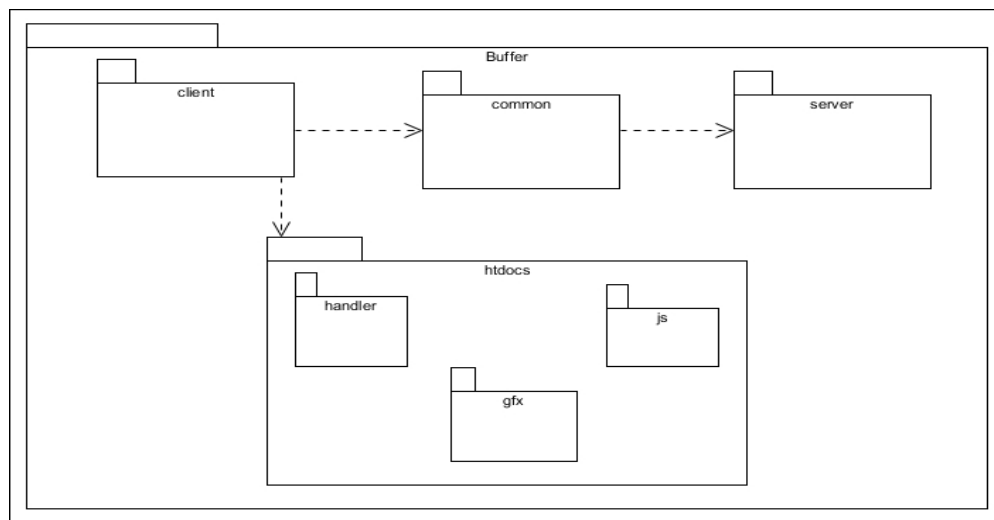


Fig. 17 Diagrama de paquetes

2.9.1 Breve descripción de los paquetes del módulo

client/ contiene todos los componentes de extensión php (*ClientBuffer.php*) del *plugin* que implementan el módulo CartoClient del CartoWeb.

server/ contiene todos los componente de extensión php (*ServerBuffer.php*) del *plugin* que implementan el módulo CartoServer del CartoWeb.

common/ contiene todos los componentes que implementan aquellas clases que servirán de puente para la comunicación entre *ClientBuffer.php* y *ServerBuffer.php*.

htdocs/gfx/ contiene las imágenes que serán utilizadas en el *plugin*.

htdocs/css/ contiene las hojas de estilo que serán utilizadas en el *plugin*.

htdocs/js/ contiene todos los componentes (de extensión js) del *plugin* que implementan el módulo cliente en javascript del CartoWeb.

2.10 Modelo de clases del diseño

El modelo de clases del diseño describe gráficamente las especificaciones y relaciones entre clases.

Contiene la siguiente información (Valencia, 2013):

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de los atributos.
- Navegabilidad.
- Dependencias.

La Fig 18 muestra el diagrama de clases del diseño correspondiente al módulo.

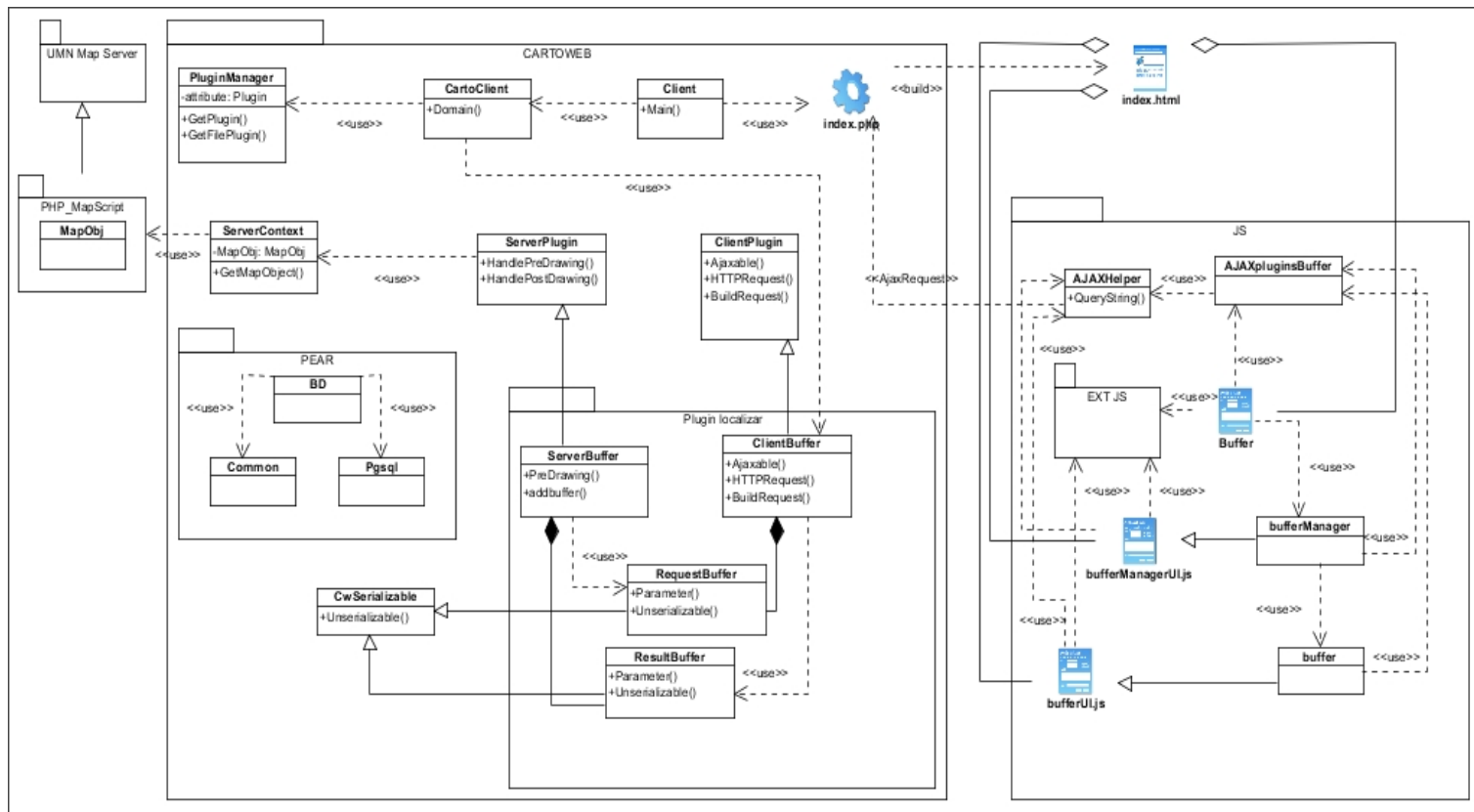


Fig. 18 Diagrama de clases del diseño

2.11 Modelo de Datos

El Modelo Entidad-Relación permite representar las entidades existentes y las relaciones entre ellas. Tiene como propósito garantizar que los datos persistentes sean almacenados y que sea definido el comportamiento que debe ser implementado en las bases de datos.

El diagrama Entidad-Relación correspondiente al módulo está constituido por tres tablas las cuales se relacionan entre sí; la tabla capa se relaciona con la tabla propiedad y criterio a través de la relación de uno a muchos al igual que las dos últimas mencionadas (ver Fig 19). Ya constituido el modelo de datos se desprende del mismo el modelo de clases persistente, cuyo objetivo es mostrar las clases que permanecerán aun después de terminada la ejecución del software (ver Fig 20).

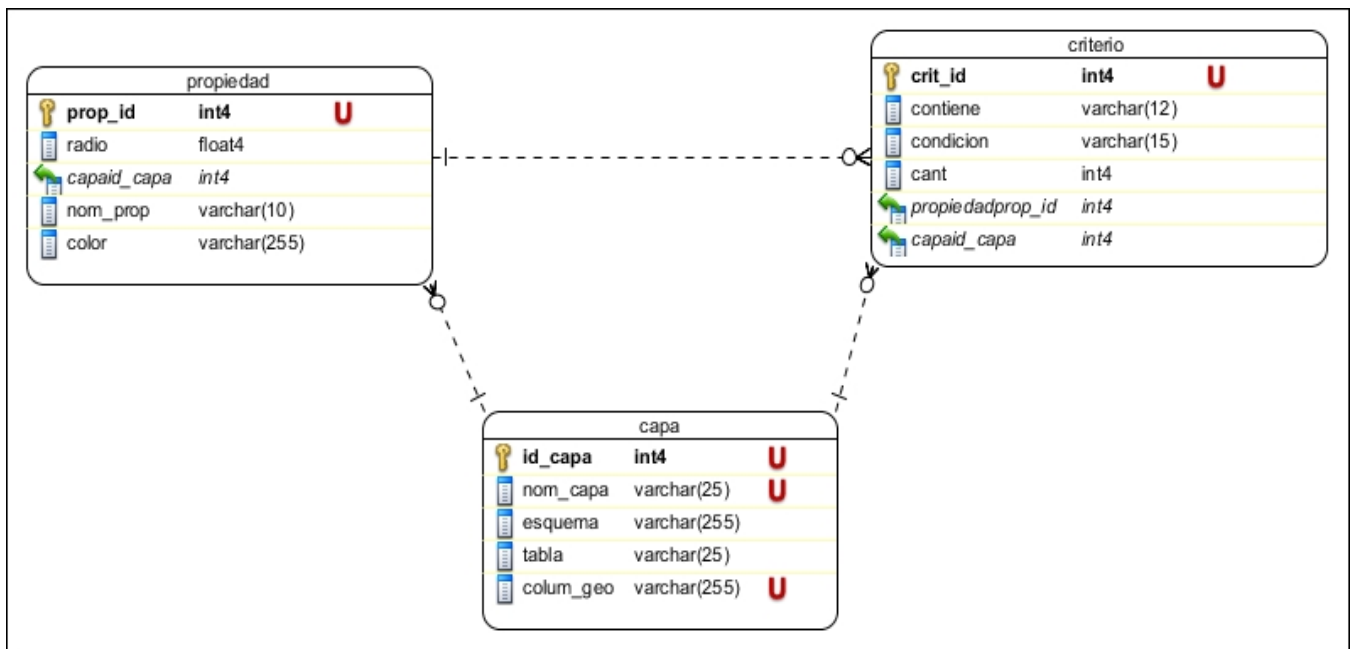


Fig. 19 Modelo Entidad-Relación

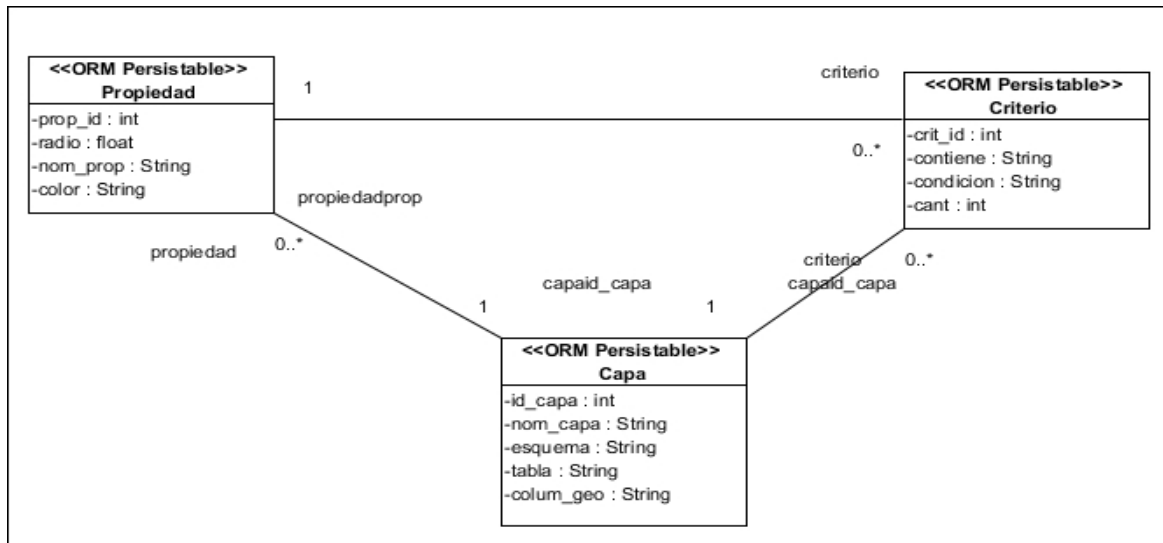


Fig. 20 Diagrama de clases persistentes

2.12 Conclusiones

Luego de generar los artefactos fundamentales propuestos por la metodología RUP en sus primeras fases, se sientan las bases para lograr que durante el proceso de implementación se cumplan los requisitos definidos; cuya identificación permite que se obtenga un producto que satisfaga las necesidades del cliente. Una adecuada selección de los requisitos no funcionales proporcionará un ambiente estable para que el software cuente con los recursos necesarios para realizar sus funcionalidades.

Se expone la arquitectura sobre la cual se desarrollará el módulo, basándose en la estructura que presenta la plataforma GeneSIG. Además se muestran los diagramas de paquetes y clases del diseño que representan la estructura del módulo para una mayor comprensión de las relaciones entre los paquetes y sus clases, sumándosele a ello el diagrama modelo Entidad–Relación permite observar la organización física y lógica de la base de datos con la cual cuenta el sistema.

Capítulo.3 Implementación y Pruebas

3.1 Introducción

Conociendo las necesidades básicas del usuario, las tecnologías y recursos necesarios para lograr un ambiente de trabajo adecuado y definida la arquitectura base sobre la cual será sustentado el módulo a desarrollar; se dará paso a la fase de implementación. Como parte de la fase de construcción del módulo de operaciones geométricas se muestran durante el desarrollo del capítulo los artefactos ingenieriles, entre los que se encuentran los diagramas de componentes y de despliegue. Posteriormente, se describen las pruebas a las que se sometió el sistema para verificar el cumplimiento de cada uno de los requisitos especificados y detectar posibles errores.

3.2 Implementación

Durante el desarrollo del flujo de trabajo de implementación definido por la metodología RUP, los resultados del diseño se transforman en componentes de ficheros que contienen código fuente donde se evidencia la arquitectura planteada y se describe la organización y la relación existente entre los componentes del sistema.

3.3 Diagrama de Componente

El diagrama de componente describe los elementos físicos del sistema y sus relaciones, muestra las opciones de realización incluyendo código fuente, binario y ejecutable. En este diagrama se pueden manejar paquetes que son contenedores de clases utilizados para mantener el espacio de nombres de clases dividido en compartimentos, de manera que se utilizan para representar subsistemas del sistema en el mundo físico. Cada paquete se liga con otros a través de dependencias, que se representan con flechas de líneas discontinuas que van del componente dependiente al componente del cual depende (Zamitiz, 2013).

En la Fig 21 se muestra el diagrama de componentes correspondiente al módulo, donde se representan los componentes del producto:

JS: Incluye todos los ficheros con extensión .js, es decir las vistas del módulo con su lógica correspondiente y el componente encargado de enviar las peticiones al servidor web.

CartoWeb: Representa la forma en que se organizan los componentes y su relación en el momento de ejecución del módulo, así como su acceso al servidor de Base datos como al servidor de mapas.

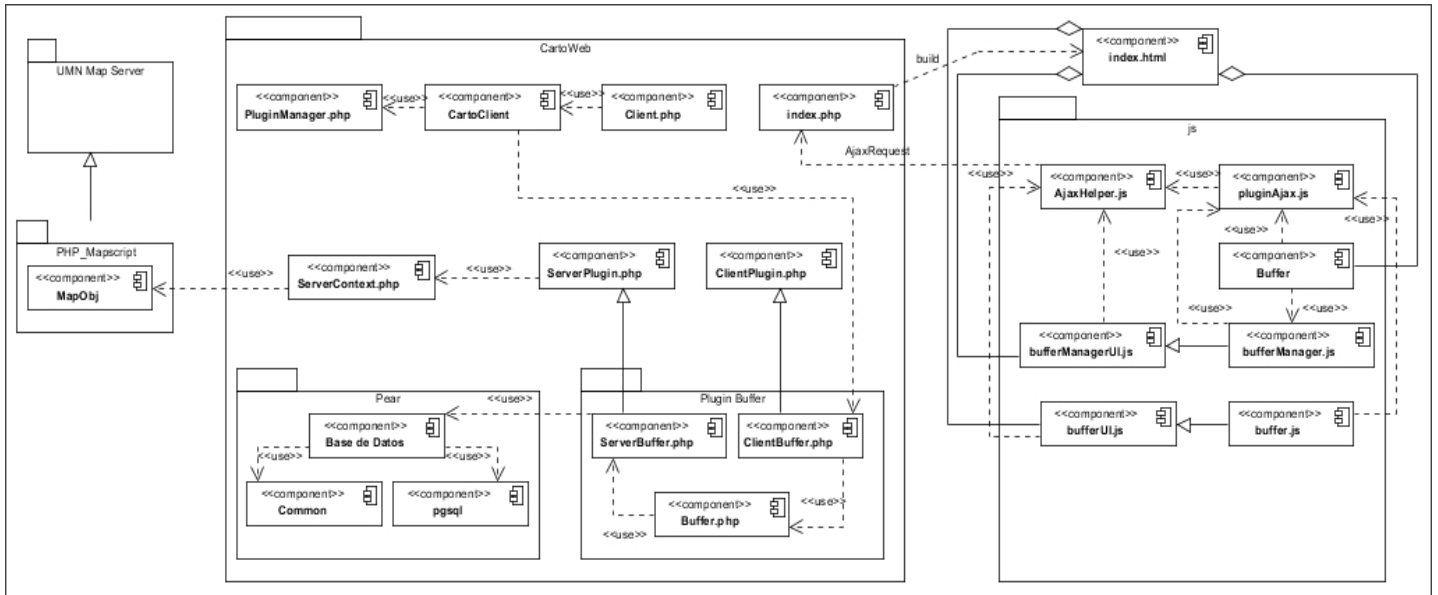


Fig. 21 Diagrama de Componentes

3.4 Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador o un dispositivo.

En la Fig 22 se muestra el diagrama de despliegue correspondiente al módulo de operaciones geométricas, el cual cuenta con un servidor web Apache que se encarga de manejar todas las peticiones realizadas desde el ordenador cliente a través del protocolo HTTP, además de construir las interfaces de usuarios y procesar los datos antes de enviar la respuesta. Se requiere del uso de un servidor de mapas MapServer que se comunica vía TCP/IP con el servidor de base de datos, siendo PostgreSQL el gestor de base de datos a utilizar con extensión PostGIS para trabajo con datos espaciales.

Protocolos

HTTP (Protocolo de Transferencia de Hipertexto): Desde un navegador, permite a una computadora cliente leer y ejecutar páginas web de un servidor web que se encuentra dentro de la Intranet o en Internet. Estas páginas pueden incluir texto, imágenes, sonido, video y vínculos a otros archivos HTML (Poratti, 2004). En el sistema es utilizado para la conexión de los clientes con los servidores web y de Mapas.

TCP/IP (*Transfer Control Protocol/Internet Protocol*, lo que en español significa Protocolo de Control de Transmisión/Protocolo Internet): Permite llevar a cabo la comunicación entre diferentes plataformas, sistemas operativos, topología y arquitecturas por el mejor camino disponible puede ser usado tanto en una red LAN (red de área local) de dos máquinas, como en una red WAN (red de área extensa) compuesta por millones de máquinas, por ejemplo, Internet (Poratti, 2004). En el caso del módulo se emplea en la comunicación de los servidores Web y de Mapas con el servidor de Base de Datos.

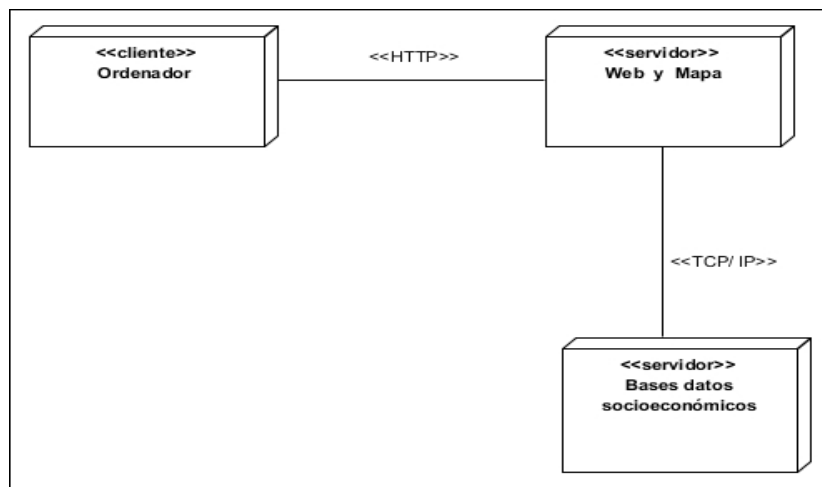


Fig. 22 Diagrama de despliegue

3.5 Prueba

La fase de pruebas es una de las más importantes durante el ciclo de desarrollo de un software, en esta se verifican y evitan posibles errores en el período de ejecución del sistema, antes de ser entregado al usuario final. Su objetivo principal es proporcionar la información necesaria sobre la calidad del producto y asegurar el correcto funcionamiento de todos los requisitos especificados por el cliente.

3.6 Pruebas realizadas al módulo

Las pruebas realizadas al módulo fueron las de caja negra de manera que se pudieron comprobar los requisitos funcionales del software, validándose por completo los valores de entrada del sistema. Estas pruebas son aplicadas sobre la interfaz del producto para tratar de encontrar errores de interfaz, funciones incorrectas o faltantes, errores de estructura de datos, errores de comportamiento, errores de inicialización y culminación y de validación.

El método aplicado de caja negra es la partición equivalente que divide el dominio de entrada del sistema en clases de datos, a partir de las cuales se derivan dos casos de pruebas correspondientes al módulo.

3.6.1 Diseño de casos de prueba

Descripción general: El caso de uso se inicia cuando el usuario inserta las propiedades definidas para determinar la zona de influencia, una vez insertadas se obtendrá un listado con todos los datos de la propiedad la cual se podrá modificar y eliminar. Finalizando el caso de uso con la visualización de la zona de influencia.

Tabla.5 Descripción de las variables

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Nombre de la propiedad	Campo de texto	No	Nombre que desee al usuario para identificar la propiedad de la zona de influencia.
2	Radio	Campo de texto	No	Radio de la zona de influencia
3	Capa	Lista desplegable	No	Capa del mapa con la que se trabajará
4	Color	Lista desplegable	No	Color de la zona de influencia
5	<i>Buffer</i>	Lista	No	Lista de <i>buffer</i> guardados en el sistema.

Tabla.6 Descripción de las secciones

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: "Insertar"	EC 1.1: "Insertar valores correctamente"	El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario inserte los valores necesarios para calcular una zona de influencia. El usuario inserta los datos para crear la zona de influencia y da clic en el botón Aceptar. El sistema valida los datos, los almacena visualizándolos en la lista de <i>buffer</i> de la ventana Gestionar Zona de influencia	GeneSIG. Clic en la opción <i>Buffer</i> . Clic en la opción Nueva. Llena los campos de la ventana Zonas de influencia. Clic en el botón Aceptar.

		y muestra el mensaje “Zona de influencia correctamente creada”.	
	EC 1.2: “Insertar valores incorrectamente”	El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario inserte los valores necesarios para calcular una zona de influencia. El usuario deja campos vacíos para crear la zona de influencia y da clic en el botón Aceptar. El sistema muestra el mensaje “No deben existir campos vacíos”.	GeneSIG. Clic en la opción <i>Buffer</i> . Clic en la opción Nueva. Llena los campos de la ventana Zonas de influencia. Clic en el botón Aceptar.
	EC 1.3: “Cancelar opción de insertar”	El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario inserte los valores necesarios para calcular una zona de influencia. El usuario inserta los datos para crear la zona de influencia y da clic en el botón Cancelar. El sistema cierra la ventana.	GeneSIG. Clic en la opción <i>Buffer</i> . Clic en la opción Nueva. Llena los campos de la ventana Zonas de influencia. Clic en el botón Cancelar.
SC 2: “Modificar”	EC 2.1: “Modificar valores correctamente”	El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario modifique los valores necesarios para calcular una zona de influencia. El usuario modifica los datos para crear la zona de influencia y da clic en el botón Modificar. El sistema valida los datos, los almacena visualizándolos en la lista de <i>buffer</i> de la ventana Gestionar Zona de influencia	GeneSIG. Clic en la opción <i>Buffer</i> . Clic en la lista de <i>buffer</i> para seleccionar uno. Clic en la opción Modificar. Llena los campos de la ventana Zonas de influencia. Clic en el botón Modificar.

		y muestra el mensaje “Zona de influencia correctamente modificados”.	
	EC 2.2: “Modificar valores incorrectamente”	El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario modifique los valores necesarios para calcular una zona de influencia. El usuario deja campos vacíos y da clic en el botón Modificar. El sistema muestra el mensaje “No deben existir campos vacíos”.	GeneSIG. Clic en la opción <i>Buffer</i> . Clic en la lista de <i>buffer</i> para seleccionar uno. Clic en la opción Modificar. Llena los campos de la ventana Zonas de influencia. Clic en el botón Modificar.
	EC 2.3: “Cancelar opción de modificar”	El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario modifique los valores necesarios para calcular una zona de influencia. El usuario selecciona de la lista de <i>buffer</i> el que desea modificar y da clic en el botón Cancelar. El sistema cancela la selección realizada.	GeneSIG. Clic en la opción <i>Buffer</i> . Clic en la lista de <i>buffer</i> para seleccionar uno. Clic en la opción Modificar. Llena los campos de la ventana Zonas de influencia. Clic en el botón Cancelar.
	EC 2.4: “Seleccionar Modificar sin elegir un <i>buffer</i> ”	El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario modifique los valores necesarios para calcular una zona de influencia. El usuario no selecciona de la lista de <i>buffer</i> el que desea modificar y da clic en el botón Modificar. El sistema muestra el mensaje “Debe seleccionar un <i>buffer</i> ”.	GeneSIG. Clic en la opción <i>Buffer</i> . Clic en la opción Modificar.

<p>SC 3: "Mostrar"</p>	<p>EC 3.1: "Mostrar <i>buffer</i> correctamente"</p>	<p>El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario visualice una zona de influencia. El usuario selecciona de la lista de <i>buffer</i> el que desea visualizar y da clic en el botón Mostrar. El sistema cierra la ventana Gestionar Zona de influencia y muestra en el mapa el <i>buffer</i> seleccionado.</p>	<p>GeneSIG. Clic en la opción <i>Buffer</i>.</p> <p>Clic en la lista de <i>buffer</i> para seleccionar uno.</p> <p>Clic en la opción Mostrar.</p>
	<p>EC 3.2: "Seleccionar mostrar sin elegir un <i>buffer</i>"</p>	<p>El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario visualice una zona de influencia. El usuario no selecciona de la lista de <i>buffer</i> el que desea visualizar y da clic en el botón Mostrar. El sistema muestra el mensaje "Debe seleccionar un <i>buffer</i>".</p>	<p>GeneSIG. Clic en la opción <i>Buffer</i>.</p> <p>Clic en la opción Mostrar.</p>
	<p>EC 3.3: "Cancelar opción mostrar"</p>	<p>El usuario elige la opción <i>Buffer</i> y el sistema muestra la ventana Gestionar Zona de influencia para que el usuario visualice una zona de influencia. El usuario selecciona de la lista de <i>buffer</i> el que desea visualizar y da clic en el botón Cancelar. El sistema cancela la selección realizada.</p>	<p>GeneSIG. Clic en la opción <i>Buffer</i>.</p> <p>Clic en la lista de <i>buffer</i> para seleccionar uno.</p> <p>Clic en la opción Cancelar.</p>

Tabla.7 Descripción de la SC 1 “Insertar”

ID del escenario	Escenario	Var 1	Var 2	Var 3	Var 4	Respuesta del sistema	Resultado de la prueba.
EC 1.1	Insertar valores correctamente.	V/Prop 1	V/3	V/Provincia	V/azul	El sistema valida los datos, los almacena visualizándolos en la lista de <i>buffer</i> de la ventana Gestionar Zona de influencia y muestra el mensaje “Zona de influencia creada correctamente”.	Satisfactorio.
EC 1.2	Insertar valores incorrectamente.	I/Campo Vacío	V/3	V/Provincia	V/azul	El sistema muestra el mensaje “No deben existir campos vacíos”.	Mensaje de error.
		V/Prop 1	I/Campo Vacío	V/Provincia	V/azul		
		V/Prop 1	V/3	I/Campo Vacío	V/azul		Satisfactorio.
		V/Prop 1	V/3	V/Provincia	I/Campo Vacío		
EC 1.3	Cancelar opción de insertar.	V/Prop 1	V/3	V/Provincia	V/azul	El sistema cierra la ventana.	Satisfactorio.

Tabla.8 Descripción de la SC 2 “Modificar”

ID del escenario	Escenario	Var 1	Var 2	Var 3	Var 4	Var 5	Respuesta del sistema	Resultado de la prueba.
EC 2.1	Modificar valores correctamente.	V/Prop 1	V/3	V/Provincia	V/azul	V/Seleccionado un <i>buffer</i>	El sistema valida los datos, los almacena visualizándolos en la lista de <i>buffer</i> de la ventana Gestionar Zona de influencia y muestra el mensaje “Zona de influencia correctamente modificada”.	Satisfactorio.
EC 2.2	Modificar valores incorrectamente.	I/Campo Vacío	V/3	V/Provincia	V/azul	V/Seleccionado un <i>buffer</i>	El sistema muestra el mensaje “No deben existir campos vacíos”.	Mensaje de error.
		V/Prop 1	I/Campo Vacío	V/Provincia	V/azul	V/Seleccionado un <i>buffer</i>		
		V/Prop 1	V/3	I/Campo Vacío	V/azul	V/Seleccionado un <i>buffer</i>		
		V/Prop 1	V/3	V/Provincia	I/Campo Vacío	V/Seleccionado un <i>buffer</i>		
EC 2.3	Cancelar opción de modificar.	V/Prop 1	V/3	V/Provincia	V/azul	V/Seleccionado	El sistema cancela la selección realizada.	Satisfactorio.

						do un <i>buffer</i>		
EC 2.4	Seleccionar Modificar sin elegir un <i>buffer</i> .	NA	NA	NA	NA	I/No es selecci onado un <i>buffer</i>	El sistema muestra el mensaje “Debe seleccionar un <i>buffer</i> ”.	Mensaje de error.

Tabla.9 Descripción de la SC 3 “Mostrar”

ID del escenario	Escenario	Var 1	Var 2	Var 3	Var 4	Var 5	Respuesta del sistema	Resultado de la prueba.
EC 3.1	Mostrar <i>buffer</i> correctamente.	NA	NA	NA	NA	V/Sele cciona do un <i>buffer</i>	El sistema cierra la ventana Gestionar Zona de influencia y muestra en el mapa el <i>buffer</i> seleccionado.	Satisfactorio.
EC 3.2	Seleccionar mostrar sin elegir un <i>buffer</i> .	NA	NA	NA	NA	I/No es selecci onado un <i>buffer</i>	El sistema muestra el mensaje “Debe seleccionar un <i>buffer</i> ”.	Mensaje de error.
EC 3.3	Cancelar opción mostrar.	NA	NA	NA	NA	V/Sele cciona do un <i>buffer</i>	El sistema cancela la selección realizada.	Mensaje de error.

3.7 Resultados de las pruebas

La tabla 10 y la Fig 23 muestra los resultados correspondientes a las pruebas realizadas al módulo de operaciones geométricas, que contiene el número de iteraciones que se le aplicaron a los casos de prueba, el total de no conformidades¹⁰ encontradas y las que fueron resueltas.

Tabla.10 Resultados de las pruebas de caja negra

No. de Iteraciones	Total de no conformidades	No procede	Resueltas
1era Iteración	5	-	5
2da Iteración	1	-	1

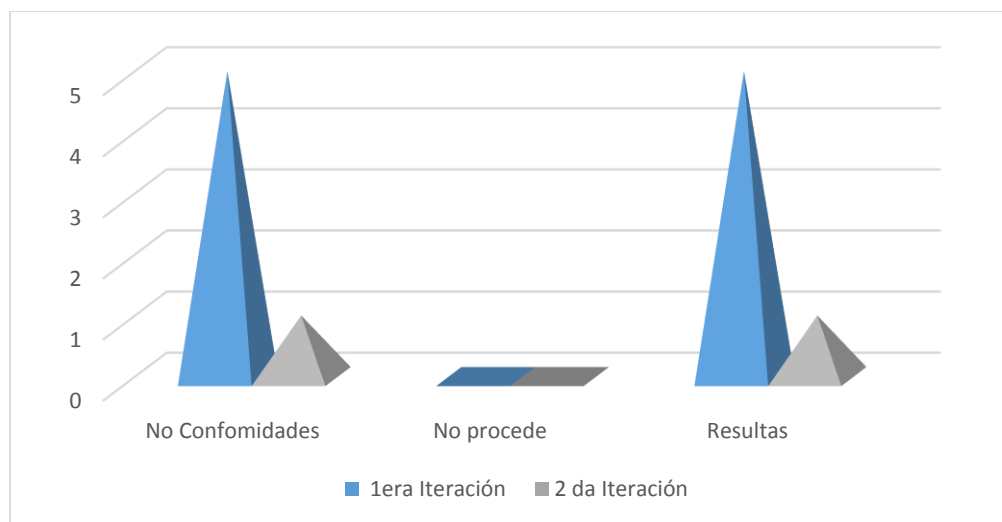


Fig. 23 Resultado de las pruebas

3.8 Conclusiones

En el presente capítulo se muestra como se desarrolló el módulo, reflejado a través de los artefactos correspondientes al flujo de trabajo Implementación y Pruebas definidos por RUP, como son: el diagrama de componentes que representa los elementos físicos y las relaciones existentes entre ellos y el diagrama de despliegue que permite mostrar de forma gráfica la distribución física del módulo en tiempo de ejecución. La

¹⁰ no conformidades: problemas detectados al probar la aplicación.

aplicación de pruebas de caja negra al sistema, permitió comprobar hasta qué punto este cumple con las funcionalidades requeridas por el cliente. A través de los resultados arrojados de las dos iteraciones a las que fue sometido el módulo se corrigieron las deficiencias encontradas.

Conclusiones

La existencia de algoritmos geométricos contribuye a gran escala a la simplificación de complejos cálculos geométricos. La combinación de estos algoritmos con los SIG; hacen de este último una herramienta capacitada para favorecer y facilitar el manejo de la información geoespacial. Al finalizar la presente investigación se arriban a las siguientes conclusiones:

El estudio y análisis de los algoritmos geométricos expuesto en la investigación, permitió comprender el proceso de cálculo de operaciones de solape, determinación de contornos mínimos y zonas de influencia. Profundizando en el procedimiento para determinar zonas de influencia, empleándose diferentes entidades espaciales.

A partir de la observación del funcionamiento de varias bibliotecas que permiten realizar cálculos geométricos, se seleccionó PostGIS luego de comparar sus funciones correspondientes a la investigación, sus ventajas y desventajas.

La creación del módulo de operaciones geométricas para la plataforma GeneSIG brinda la posibilidad de:

- Determinar zonas de influencia a través de la inserción de parámetros definidos para que el usuario logre resultados válidos en dependencia del ámbito en que se esté trabajando y su posterior visualización en el mapa.
- Almacenar la capa resultante de la operación para futuros análisis.
- Apoyar al proceso de toma de decisiones.

Los artefactos ingenieriles generados en el proceso de desarrollo proveen el entendimiento de la estructura del módulo, permitiendo así la modificación e incorporación de nuevas funcionalidades o la reutilización de algún que otro componente sin dañar la composición del módulo.

Recomendaciones

Concluido el módulo de operaciones geométricas y desarrolladas todas las tareas trazadas, con el propósito de mejorar el resultado obtenido durante la investigación se proponen las siguientes recomendaciones:

Aplicar el módulo en proyectos productivos que desarrollen SIG en la UCI y en Cuba que hagan uso de la plataforma GeneSIG.

Continuar con el desarrollo y ampliación del módulo de forma tal que se le añadan las operaciones geométricas de solape y contornos mínimos.

Referencias

- Abela, Charlie.** 2010. *Imperative and Object Oriented*. [En línea]. [Citado el: 12 de noviembre de 2013.] http://staff.um.edu.mt/cabe2/lectures/.../netbeans_tutorial.pdf.
- Agu, Emmanuel.** CS 543: *Computer Graphics*. Lecture 8: 3D Clipping and Viewport Transformation. [En línea] [Citado el: 23 de enero de 2014.] <http://web.cs.wpi.edu/~emmanuel/courses/cs543/slides/lecture8.pdf>.
- ALEGSA.** 2014. *¿Qué es un plugin?*. ALEGSA.com.ar. [En línea]. [Citado el: 20 de mayo de 2014.] <http://www.alegsa.com.ar/Dic/plugin.php>.
- Apache Tomcat.** 2013. Sitio Web Oficial de Apache Tomcat. [En línea] Apache Software Foundation. [Citado el: 6 de diciembre de 2013.] <http://www.apachetomcat.org>.
- Balmaseda, C y Romero, Madelin .** 2012. *Revista de Protección Vegetal*. [En línea]. [Citado el: 24 de octubre de 2013.] http://scielo.sld.cu/scielo.php?pid=S1010-27522012000300012&script=sci_arttext.1010-2752.
- Batista Rodríguez, José A. , y otros.** 2007. *Diseño y desarrollo de un sistema de información geográfica para la gestión de la información geológica y geofísica de Cuba nororiental*. [En línea]. [Citado el: 24 de octubre de 2013.] <http://revista.ismm.edu.cu/index.php/revistamg/article/viewFile/86/91.1993.8012>.
- Bradley University.** 2009. WEILER-ATHERTON. [En línea]. [Citado el: 21 de enero de 2014.] <http://cs1.bradley.edu/public/jcm/weileratherton.html>.
- Brenes, Carlos.** 2005. *SISTEMAS DE INFORMACIÓN GEOGRÁFICA*. [En línea]. [Citado el: 29 de noviembre de 2013.] <http://www.oirsa.org/aplicaciones/subidoarchivos/BibliotecaVirtual/sistemasinfgeog.pdf>.
- Cartografía.cl.** 2005. Cartografía.cl. [En línea]. [Citado el: 12 de diciembre de 2013.] <http://www.cartografia.cl/beta/index.php/home/sig/713-una-biblioteca-java-libre-para-analisis-geoespacial>.
- CartoWeb.** 2008. *CartoWeb*. [En línea]. [Citado el: 20 de marzo de 2014.] <http://cartoweb.org/>.
- Confederación de Empresarios de Andalucía(CEA).** 2010. *Sistema de Información Geográfica, tipos y aplicaciones empresariales*. [En línea]. [Citado el: 4 de diciembre de 2013.] <http://sig.cea.es/gvsig--sig.tipos.y.aplicaciones.empresariales>.
- Definición.de.** 2013. Definición [En línea]. [Citado el: 12 de diciembre de 2013.] <http://definicion.de/algoritmo/#ixzz2nL2MjUDE>.
- . 2013. Definición [En línea]. [Citado el: 11 de diciembre de 2013.] <http://definicion.de/plugin/>.

- 1998-2013.** Diseño web y desarrollo web. *Programación en Castellano*. [En línea]. [Citado el: 7 de diciembre de 2013.] http://www.programacion.com/articulo/eclipse_--_i_-_historia_y_toma_de_contacto_288.
- Domech, Armando González.** *LA CARTOGRAFÍA DE ATLAS EN CUBA: SITUACIÓN ACTUAL Y PERSPECTIVAS EN EL SIGLO XXI*. [En línea] [Citado el: 12 de noviembre de 2013.] <http://observatoriogeograficoamericalatina.org.mx/egal9/Nuevastechnologias/Sig/01.pdf>.
- Equipo SEXTANTE.** 2008. *Conceptos generales de SEXTANTE*. [En línea]. foro.gabrielortiz.com/comparte/.../ManualElementosSEXTANTE.pdf. Edición 1.0.
- ESRI.** 2012. *ArcGIS Resource Center*. [En línea]. [Citado el: 19 de diciembre de 2013.] <http://help.arcgis.com/es/arcgisdesktop/10.0/help/index.html#/000800000019000000>.
- . **ESRI.** *ESRI*. [En línea] [Citado el: 4 de diciembre de 2013.] <http://www.esri.es/es/productos/arcgis>.
- Facultad de informática - Universidad Politécnica de Madrid.** 2010. Universidad Politécnica de Madrid. [En línea]. [Citado el: 24 de marzo de 2014.] http://is.ls.fi.upm.es/docencia/proyecto/docs/patrones_gof.pdf.
- Felicísimo, Angel Manuel.** 2000. *Modelos digitales*. s.l. : Pentalfa, 2000. ISBN 84-7848-475-2.
- Free Download Manager.** 2007. *Sitio de descarga de software*. [En línea] 5 de marzo de 2007. [Citado el: 6 de diciembre de 2013.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
- Gamma, E, y otros.** 1995. *Desing Patterns*. Autoedición 1995. ISBN 0-201- 63361-2.
- García, Andrés Reynaldo Milord y Yesnier Bravo.** 2012. *Compumat*. [En línea] 2012. [Citado el: 5 de diciembre de 2013.] http://compumat.uci.cu/sites/default/files/public/p983-ponencia-3644_0.pdf.
- García, Ing. Nelly.** 2009. *Diseño e implementación de un Servidor de Mapas Web para una Red Bioclimática en Montaña*. Mérida, 2009.
- GDAL.** 2014. *GDAL - Geospatial Data Abstraction Library*. [En línea] 2014. [Citado el: 24 de febrero de 2014.] <http://www.gdal.org/ogr/classOGRGeometry.html#ac906abd9b1cd9dd2b208a10c6e9d1bf0>.
- . 2013. *Geospatial Data Abstraction Library (GDAL)*. [En línea] 2013. [Citado el: 12 de diciembre de 2013.] <http://www.gdal.org/>.
- GOODMAN, Danny.** 2001. *JavaScript Bible*. New York: Hungry Minds, Inc.,. ISBN:0764547186.
- Grupo EIDOS.** 2012. *Lenguaje HTML*. [En línea]. [Citado el: 21 de diciembre de 2013.] <http://ebookbrowse.net/lenguaje-html-jesus-sanchez-grupo-eidos-pdf-d299315684>. Versión 1.0.0.

- Gutiérrez, Antonio y Pascual Broncano, Pedro J.** 2014. *Apuntes de Gráficos*. [En línea]. [Citado el: 22 de enero de 2014.] <http://arantxa.ii.uam.es/~pedro/graficos/teoria/Recorte2D/Recorte2D.htm#conceptos>.
- Instituto de estadística y cartografía de Andalucía.** 2011. *Manual. Manual para la transformación de formato GIS a CAD con cambio de sistema de referencia mediante librerías GDAL*. 2011.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** 2000. *El Proceso Unificado del Desarrollo del Software*. Madrid, España: Pearson Educación, S.A., 2000. ISBN 8478290362.
- Kicillof, Carlos Reynoso y Nicolás.** 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. [En línea]. [Citado el: 10 de marzo de 2014.] <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
- Laguna, Miguel A.** 2009. *Ingeniería del Software I 3º I.T.I.Gestión*. [En línea] 2009. [Citado el: 3 de marzo de 2014.] www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf.
- LANGHI, J.C.D YAGÜEZ y R.** 2002. *Sistema de Información Geográfica (S.I.G)*. [En línea] 2002. [Citado el: 29 de noviembre de 2013.] http://www.inta.gov.ar/barrow/info/documentos/SIG/que_es_sig.htm.
- Levis, Diego.** 2012. *Glosario*. [En línea]. [Citado el: 11 de diciembre de 2013.] <http://diegolevis.com.ar/secciones/Articulos/glosario1.html#P>.
- Lozano, Pablo Ramiro Correa.** 2010. *Análisis comparativo de los Frameworks Adobe Flex, JAVA RICH FACES y EXT JS para el desarrollo de aplicaciones enriquecidas en Internet (RIA)*. Quito., 2010.
- Mappiggis.com.** 2012. *Mappiggis.com*. [En línea]. [Citado el: 12 de diciembre de 2013.] <http://mappiggis.com/2012/09/>.
- Microsoft.** Developer Network. [En línea] [Citado el: 10 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/bb972268.aspx#ref01back>.
- Mora, Felipe.** 2009. *Espacio Geográfico*. [En línea]. [Citado el: 24 de octubre de 2013.] <http://felipemorageo.blogspot.com/2009/03/la-cartografia-y-su-historia.html>.
- National Centre of Geographic Information and Analysis (NCGIA)** 1990. *Introducción a los SIG*. [En línea] [Citado el: 11 de diciembre de 2013.] <http://www.geogra.uah.es/gisweb/1modulosespanyol/IntroduccionSIG/GISModule/GISTheory.htm>.
- NCGIA.** 1990. *NCGIA National Center for Geographic Information and Analysis 1990*. [En línea]. [Citado el: 29 de noviembre de 2013.] <http://www.ncgia.ucsb.edu>.
- O'Rourke, Joseph.** 2000. *Computational Geometry in C*. Cambridge, Inglaterra : Cambridge University Press, 2000. ISBN 0521649765.
- Olaya, Victor.** 2011. *Sistemas de Información Geográfica*. 2011.

- OSGEO.** *QGIS 0.9.1 Guía de usuario e instalación.* [En línea] [Citado el: 3 de diciembre de 2013.] http://download.osgeo.org/qgis/doc/manual/qgis-0.9.1_user_guide_es.pdf.
- PHP.** 2013. PHP. [En línea]. [Citado el: 6 de diciembre de 2013.] <http://php.net>.
- Poratti, Gustavo Gabriel.** 2004. *Redes. La guía de referencia total y definitiva.* Buenos Aires, Argentina.
- PostGIS.** 2012. Manual. *Manual de PostGIS.* 2012.
- . 2008. OSGEO. *Manual PostGIS 1.5.8.* [En línea]. [Citado el: 25 de febrero de 2014.] <http://download.osgeo.org/postgis/docs/postgis-1.5.8.pdf>.
- . 2010. *PostGIS power.* [En línea]. [Citado el: 21 de diciembre de 2013.] <http://s3.opengeo.org/>.
- Pressman, Roger S.** 2005. *Ingeniería del software:Un enfoque práctico.*: Mc Gram Hill, 2005. ISBN 0-07-285318-2.
- Proenza, Yuniel.** 2011. *Revista GeoNews.* [En línea]. [Citado el: 24 de octubre de 2013.] <http://revistageonews.wordpress.com/category/el-sig-del-mes/>.
- QGIS.** 2013. QGIS. [En línea]. [Citado el: 4 de diciembre de 2013.] <http://qgis.org/es/site/>.
- Ramsey, Paul.** *Manual PostGIS.* versión 1.5
- Rivera, Miguel Félix Mata.** 2004. *Tesis: Recuperando y Analizando Datos.* México D.F.
- Robaina, Irilys Ledón.** 2008. *Propuesta del Diseño Arquitectónico del Simulador de Sistemas Biológicos.* La Habana, Cuba.
- Romero, Hermenegildo.** 2009. *Metodologías de desarrollo.* Slideshare.net. [En línea]. [Citado el: 6 de diciembre de 2013.] <http://www.slideshare.net/MeneRomero/metodologias-de-desarrollo>.
- Sánchez, María A. Mendoza.** 2004. *Metodologías De Desarrollo De Software.* [En línea]. [Citado el: 6 de diciembre de 2013.] <http://www.informatizate.net>.
- Servicios Académicos Internacionales S.C.** 2013. *BIBLIOTECA VIRTUAL de Derecho, Economía y Ciencias Sociales.* [En línea]. [Citado el: 25 de octubre de 2013.] <http://www.eumed.net/libros-gratis/2009c/590/EL%20DESARROLLO%20DE%20LOS%20SIG.htm>.
- Sherman, Gary.** *Qgis.* [En línea] [Citado el: 4 de diciembre de 2013.] es.scribd.com/doc/36183389/Qgis-0-9-1-User-Guide-Es.
- Global Biodiversity Information Facility in Spain(GBIF)** *Sistema de Información Geográfica (SIG). Técnicas básicas para estudios de biodiversidad .* [En línea] [Citado el: 15 de noviembre de 2013.] [http://www.gbif.es/ficheros/Guion_SIG.pdf\(14-11\)](http://www.gbif.es/ficheros/Guion_SIG.pdf(14-11)).

Sitio Web Oficial de Lenguaje Unificado de Modelado. 2013. *Lenguaje Unificado de Modelado (UML)*. [En línea]. [Citado el: 6 de diciembre de 2013.] <http://www.uml.org>.

Sommerville, Ian. 2005. *Ingeniería del software*. Madrid, 2005. ISBN 84-7829-074-5.

—. 2005. *INGENIERÍA DEL SOFTWARE*. [trad.] United Kingdom Pearson Education Limited. Séptima Edición. Madrid: Addison-Wesley, 2005. pág. 712. ISBN: 84-7829-074-5.

Steven, R. Davidson. 2011. *Curso de Gráficos*. [En línea] abril de 2011. [Citado el: 12 de diciembre de 2013.] <http://graficos.conclase.net/curso/?cap=007>.

Targetware Informática S.A.C. 2013. [software.com.ar](http://www.software.com.ar). [En línea]. [Citado el: 6 de diciembre de 2013.] <http://www.software.com.ar/visual-paradigm-para-uml.html>.

UCI. 2013. postgreSQL. [En línea] 2013. [Citado el: 6 de diciembre de 2013.] <http://postgreSQL.uci.cu>.

Universidad de Arantxa. 2008. *Patrones de Diseño*. [En línea]. [Citado el: 30 de marzo de 2014.] <http://arantxa.ii.uam.es/~eguerria/docencia/0708/09%20Command.pdf>.

Valencia, María Eugenia. 2013. Universidad del Valle. [En línea]. [Citado el: 26 de marzo de 2014.] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/DISCLASES_A12.pdf.

W3C. 2014. *W3C.Hojas de estilo en cascada*. [En línea]. [Citado el: 5 de marzo de 2014.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.

WordPress, Free Premium. 2011. *Tu código fuente*. [En línea]. [Citado el: 3 de marzo de 2014.] <http://www.tucodigofuente.com/2011/10/29/modelado-de-sistemas-con-casos-de-uso-que-es/>.

Wordreference. 2013. [Wordreference.com](http://www.wordreference.com). [En línea]. [Citado el: 12 de diciembre de 2013.] <http://www.wordreference.com/definicion/algoritmo>.

Zamitz, Ing. Carlos Alberto Román. 2013. *TEMAS ESPECIALES DE COMPUTACIÓN*. [En línea]. [Citado el: 21 de abril de 2014.] <http://profesores.fi-b.unam.mx/carlos/aydoo/compos.html>.

Bibliografía

- Abela, Charlie.** 2010. *Imperative and Object Oriented*. [En línea]. [Citado el: 12 de noviembre de 2013.] http://staff.um.edu.mt/cabe2/lectures/.../netbeans_tutorial.pdf.
- Agu, Emmanuel.** CS 543: *Computer Graphics.Lecture 8: 3D Clipping and Viewport Transformation*. [En línea] [Citado el: 23 de enero de 2014.] <http://web.cs.wpi.edu/~emmanuel/courses/cs543/slides/lecture8.pdf>.
- ALEGSA.** 2014. ALEGSA.com.ar. [En línea]. [Citado el: 20 de mayo de 2014.] <http://www.alegsa.com.ar/Dic/plugin.php>.
- Apache Tomcat.** 2013. Sitio Web Oficial de Apache Tomcat. [En línea] Apache Software Foundation. [Citado el: 6 de diciembre de 2013.] <http://www.apachetomcat.org>.
- Balmaseda, C y Romero, Madelin .** 2012. *Revista de Protección Vegetal*. [En línea]. [Citado el: 24 de octubre de 2013.] http://scielo.sld.cu/scielo.php?pid=S1010-27522012000300012&script=sci_arttext.1010-2752.
- Batista Rodríguez, José A. , y otros.** 2007. *Diseño y desarrollo de un sistema de información geográfica para la gestión de la información geológica y geofísica de Cuba nororiental*. [En línea]. [Citado el: 24 de octubre de 2013.] <http://revista.ismm.edu.cu/index.php/revistamg/article/viewFile/86/91.19938012>.
- Bradley University.** 2009. WEILER-ATHERTON. [En línea]. [Citado el: 21 de enero de 2014.] <http://cs1.bradley.edu/public/jcm/weileratherton.html>.
- Brenes, Carlos.** 2005. *SISTEMAS DE INFORMACIÓN GEOGRÁFICA*. [En línea]. [Citado el: 29 de noviembre de 2013.] <http://www.oirsa.org/aplicaciones/subidoarchivos/BibliotecaVirtual/sistemasinfgeog.pdf>.
- Cartografía.cl.** 2005. Cartografía.cl. [En línea]. [Citado el: 12 de diciembre de 2013.] <http://www.cartografia.cl/beta/index.php/home/sig/713-una-biblioteca-java-libre-para-analisis-geoespacial>.
- CartoWeb.** 2008. CartoWeb. [En línea]. [Citado el: 20 de marzo de 2014.] <http://cartoweb.org/>.
- Confederación de Empresarios de Andalucía(CEA).** 2010. *Sistema de Información Geográfica, tipos y aplicaciones empresariales*. [En línea]. [Citado el: 4 de diciembre de 2013.] http://sig.cea.es/gvsig--sig.tipos_y_aplicaciones_empresariales.
- Definición.de.** 2013. Definicion .de. [En línea]. [Citado el: 12 de diciembre de 2013.] <http://definicion.de/algoritmo/#ixzz2nL2MjUDE>.
- . 2013. Definición.de. [En línea]. [Citado el: 11 de diciembre de 2013.] <http://definicion.de/plugin/>.

- 1998-2013.** Diseño web y desarrollo web. *Programación en Castellano*. [En línea]. [Citado el: 7 de diciembre de 2013.] http://www.programacion.com/articulo/eclipse_--_i_-_historia_y_toma_de_contacto_288.
- Domech, Armando González.** *LA CARTOGRAFÍA DE ATLAS EN CUBA: SITUACIÓN ACTUAL Y PERSPECTIVAS EN EL SIGLO XXI*. [En línea] [Citado el: 12 de noviembre de 2013.] <http://observatoriogeograficoamericalatina.org.mx/egal9/Nuevastechnologias/Sig/01.pdf>.
- Equipo SEXTANTE.** 2008. *Conceptos generales de SEXTANTE*. [En línea]. foro.gabrielortiz.com/comparte/.../ManualElementosSEXTANTE.pdf. Edición 1.0.
- ESRI.** 2012. *ArcGIS Resource Center*. [En línea]. [Citado el: 19 de diciembre de 2013.] <http://help.arcgis.com/es/arcgisdesktop/10.0/help/index.html#/000800000019000000>.
- . **ESRI.** *ESRI*. [En línea] [Citado el: 4 de diciembre de 2013.] <http://www.esri.es/es/productos/arcgis>.
- Facultad de informática - Universidad Politécnica de Madrid.** 2010. Universidad Politécnica de Madrid. [En línea]. [Citado el: 24 de marzo de 2014.] http://is.ls.fi.upm.es/docencia/proyecto/docs/patrones_gof.pdf.
- Felicísimo, Angel Manuel.** 2000. *Modelos digitales*. s.l. : Pentalfa, 2000. ISBN 84-7848-475-2.
- Free Download Manager.** 2007. Sitio de descarga de software. [En línea] 5 de marzo de 2007. [Citado el: 6 de diciembre de 2013.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
- Gamma, E, y otros.** 1995. *Desing Patterns*. Autoedición. 1995. ISBN 0-201- 63361-2.
- García, Andrés Reynaldo Milord y Yesnier Bravo.** 2012. *Compumat*. [En línea] 2012. [Citado el: 5 de diciembre de 2013.] http://compumat.uci.cu/sites/default/files/public/p983-ponencia-3644_0.pdf.
- García, Ing. Nelly.** 2009. *Diseño e implementación de un Servidor de Mapas Web para una Red Bioclimática en Montaña*. Mérida, 2009.
- GDAL.** 2014. *GDAL - Geospatial Data Abstraction Library*. [En línea] 2014. [Citado el: 24 de febrero de 2014.] <http://www.gdal.org/ogr/classOGRGeometry.html#ac906abd9b1cd9dd2b208a10c6e9d1bf0>.
- . 2013. *Geospatial Data Abstraction Library (GDAL)*. [En línea] 2013. [Citado el: 12 de diciembre de 2013.] <http://www.gdal.org/>.
- GOODMAN, Danny.** 2001. *JavaScript Bible*. New York: Hungry Minds, Inc.,. ISBN:0764547186.
- Grupo EIDOS.** 2012. *Lenguaje HTML*. [En línea]. [Citado el: 21 de diciembre de 2013.] <http://ebookbrowse.net/lenguaje-html-jesus-sanchez-grupo-eidos-pdf-d299315684>. Versión 1.0.0.

- Gutiérrez, Antonio y Pascual Broncano, Pedro J.** 2014. Apuntes de Gráficos. [En línea]. [Citado el: 22 de enero de 2014.] <http://arantxa.ii.uam.es/~pedro/graficos/teoria/Recorte2D/Recorte2D.htm#conceptos>.
- Instituto de estadística y cartografía de Andalucía.** 2011. Manual. *Manual para la transformación de formato GIS a CAD con cambio de sistema de referencia mediante librerías GDAL*.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** 2000. *El Proceso Unificado del Desarrollo del Software*. Madrid, España: Pearson Educación, S.A., 2000. ISBN 8478290362.
- Kicillof, Carlos Reynoso y Nicolás.** 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. [En línea]. [Citado el: 10 de marzo de 2014.] <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
- Laguna, Miguel A.** 2009. *Ingeniería del Software I 3º I.T.I.Gestión*. [En línea] 2009. [Citado el: 3 de marzo de 2014.] www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf.
- LANGHI, J.C.D YAGÜEZ y R.** 2002. *Sistema de Información Geográfica (S.I.G)*. [En línea] 2002. [Citado el: 29 de noviembre de 2013.] http://www.inta.gov.ar/barrow/info/documentos/SIG/que_es_sig.htm.
- Levis, Diego.** 2012. Glosario. [En línea]. [Citado el: 11 de diciembre de 2013.] <http://diegolevis.com.ar/secciones/Articulos/glosario1.html#P>.
- Lozano, Pablo Ramiro Correa.** 2010. *Análisis comparativo de los Frameworks Adobe Flex, JAVA RICH FACES y EXT JS para el desarrollo de aplicaciones enriquecidas en Internet (RIA)*. Quito., 2010.
- Mappiggis.com.** 2012. Mappiggis.com. [En línea]. [Citado el: 12 de diciembre de 2013.] <http://mappiggis.com/2012/09/>.
- Microsoft.** Developer Network. [En línea] [Citado el: 10 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/bb972268.aspx#ref01back>.
- Mora, Felipe.** 2009. *Espacio Geográfico*. [En línea]. [Citado el: 24 de octubre de 2013.] <http://felipemorageo.blogspot.com/2009/03/la-cartografia-y-su-historia.html>.
- National Centre of Geographic Information and Analysis (NCGIA)** 1990. *Introducción a los SIG*. [En línea] [Citado el: 11 de diciembre de 2013.] <http://www.geogra.uah.es/gisweb/1modulosespanyol/IntroduccionSIG/GISModule/GISTheory.htm>.
- NCGIA.** 1990. *NCGIA National Center for Geographic Information and Analysis*. [En línea]. [Citado el: 29 de noviembre de 2013.] <http://www.ncgia.ucsb.edu>.
- O'Rourke, Joseph.** 2000. *Computational Geometry in C*. Cambridge, Inglaterra : Cambridge University Press, 2000. ISBN 0521649765.
- Olaya, Victor.** 2011. *Sistemas de Información Geográfica*. 2011.

- OSGEO.** *QGIS 0.9.1 Guía de usuario e instalación.* [En línea] [Citado el: 3 de diciembre de 2013.] http://download.osgeo.org/qgis/doc/manual/qgis-0.9.1_user_guide_es.pdf.
- PHP.** 2013. PHP. [En línea]. [Citado el: 6 de diciembre de 2013.] <http://php.net>.
- Poratti, Gustavo Gabriel.** 2004. *Redes. La guía de referencia total y definitiva.* Buenos Aires, Argentina.
- PostGIS.** 2012. Manual. *Manual de PostGIS.* 2012.
- . 2008. OSGEO. *Manual PostGIS 1.5.8.* [En línea]. [Citado el: 25 de febrero de 2014.] <http://download.osgeo.org/postgis/docs/postgis-1.5.8.pdf>.
- . 2010. *PostGIS power.* [En línea]. [Citado el: 21 de diciembre de 2013.] <http://s3.opengeo.org/>.
- Pressman, Roger S.** 2005. *Ingeniería del software:Un enfoque práctico.*: Mc Gram Hill, 2005. ISBN 0-07-285318-2.
- Proenza, Yuniel.** 2011. *Revista GeoNews.* [En línea]. [Citado el: 24 de octubre de 2013.] <http://revistageonews.wordpress.com/category/el-sig-del-mes/>.
- QGIS.** 2013. QGIS. [En línea]. [Citado el: 4 de diciembre de 2013.] <http://qgis.org/es/site/>.
- Ramsey, Paul.** *Manual PostGIS*. versión 1.5
- Rivera, Miguel Félix Mata.** 2004. *Tesis: Recuperando y Analizando Datos.* México D.F.
- Robaina, Irilys Ledón.** 2008. *Propuesta del Diseño Arquitectónico del Simulador de Sistemas Biológicos.* La Habana, Cuba.
- Romero, Hermenegildo.** 2009. *Metodologías de desarrollo* .Slideshare.net. [En línea]. [Citado el: 6 de diciembre de 2013.] <http://www.slideshare.net/MeneRomero/metodologias-de-desarrollo>.
- Sánchez, María A. Mendoza.** 2004. *Metodologías De Desarrollo De Software.* [En línea]. [Citado el: 6 de diciembre de 2013.] <http://www.informatizate.net>.
- Servicios Académicos Internacionales S.C.** 2013. *BIBLIOTECA VIRTUAL de Derecho, Economía y Ciencias Sociales.* [En línea]. [Citado el: 25 de octubre de 2013.] <http://www.eumed.net/libros-gratis/2009c/590/EL%20DESARROLLO%20DE%20LOS%20SIG.htm>.
- Sherman, Gary.** *Qgis.* [En línea] [Citado el: 4 de diciembre de 2013.] es.scribd.com/doc/36183389/Qgis-0-9-1-User-Guide-Es.
- Global Biodiversity Information Facility in Spain(GBIF).** *Sistema de Información Geográfica (SIG). Técnicas básicas para estudios de biodiversidad* . [En línea] [Citado el: 15 de noviembre de 2013.] [http://www.gbif.es/ficheros/Guion_SIG.pdf\(14-11\)](http://www.gbif.es/ficheros/Guion_SIG.pdf(14-11)).

Sitio Web Oficial de Lenguaje Unificado de Modelado. 2013. *Lenguaje Unificado de Modelado (UML)*. [En línea]. [Citado el: 6 de diciembre de 2013.] <http://www.uml.org>.

Sommerville, Ian. 2005. *Ingeniería del software*. Madrid, 2005. ISBN 84-7829-074-5.

—. 2005. *INGENIERÍA DEL SOFTWARE*. [trad.] United Kingdom Pearson Education Limited. Séptima Edición. Madrid: Addison-Wesley, 2005. pág. 712. ISBN: 84-7829-074-5.

Steven, R. Davidson. 2011. *Curso de Gráficos*. [En línea] abril de 2011. [Citado el: 12 de diciembre de 2013.] <http://graficos.conclase.net/curso/?cap=007>.

Targetware Informática S.A.C. 2013. [software.com.ar](http://www.software.com.ar). [En línea]. [Citado el: 6 de diciembre de 2013.] <http://www.software.com.ar/visual-paradigm-para-uml.html>.

UCI. 2013. postgreSQL. [En línea] 2013. [Citado el: 6 de diciembre de 2013.] <http://postgreSQL.uci.cu>.

Universidad de Arantxa. 2008. *Patrones de Diseño*. [En línea]. [Citado el: 30 de marzo de 2014.] <http://arantxa.ii.uam.es/~eguerra/docencia/0708/09%20Command.pdf>.

Valencia, María Eugenia. 2013. Universidad del Valle. [En línea]. [Citado el: 26 de marzo de 2014.] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/DISCLASES_A12.pdf.

W3C. 2014. W3C. Hojas de estilo en cascada. [En línea]. [Citado el: 5 de marzo de 2014.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.

WordPress, Free Premium. 2011. *Tu código fuente*. [En línea]. [Citado el: 3 de marzo de 2014.] <http://www.tucodigofuente.com/2011/10/29/modelado-de-sistemas-con-casos-de-uso-que-es/>.

Wordreference. 2013. [Wordreference.com](http://www.wordreference.com). [En línea]. [Citado el: 12 de diciembre de 2013.] <http://www.wordreference.com/definicion/algoritmo>.

Zamitz, Ing. Carlos Alberto Román. 2013. *TEMAS ESPECIALES DE COMPUTACIÓN*. [En línea]. [Citado el: 21 de abril de 2014.] <http://profesores.fi-b.unam.mx/carlos/aydoo/compos.html>.

Anexos

1.1. Entrevista realizada al líder del proyecto Aplicativos SIG

1. ¿Qué es GeneSIG?
2. ¿Con qué objetivo se utilizaría el módulo de determinación de zonas de influencia?
3. ¿Qué tipo de datos espaciales se analizarían?
4. ¿El tamaño del *buffer* será un valor predeterminado o el usuario podrá insertarlo?
5. ¿La capa resultante de la operación geométrica es necesario almacenarla?
6. ¿Qué aportaría almacenarla?