



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 6

**DESARROLLO DEL MÓDULO PARA LA GENERACIÓN DE LAS
VISTAS DE LOS MAPAS PARA LA PLATAFORMA SOBERANA GENESIG.**



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

Autora:



Claudia Bravo Batista

Tutora:



Ing. Yordany Piñeiro Gómez

La Habana, Junio de 2012

“Año 54 de la Revolución”



Lo que sabemos es una gota de agua; lo que ignoramos es el océano.

Isaac Newton

DECLARACIÓN DE AUTORÍA

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

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

Claudia Bravo Batista
Firma del Autor

Yordanys Piñeiro Gómez
Firma del Tutor



DATOS DE CONTACTO

Tutor: Ing. Yordany Piñeiro Gómez

Graduado de Ingeniería en Ciencias Informáticas en Julio del 2007

Profesor de Ingeniería de Software I y II y de Gestión de Software

Dirección: Universidad de las Ciencias Informáticas (UCI), Edificio: 27, Apto: 201

Teléfono Oficina: +53 – 7 – 8372559. Teléfono Apto: +53 – 7 – 835 - 8867

E-mail: ypineirog@uci.cu

Síntesis del Tutor

Profesión Profesor de Ingeniería de Software I y II y de Gestión de Software de la Facultad 6 de la UCI.

Categoría docente: Asistente

Años de graduado: 5



DEDICATORIA

A Dios, que no le puedo ver, tocar, pero su sentir mueve las hojas de los árboles y mi ser.

A mi madre Magalys, mi mamita del alma, sin su amor, paciencia, dedicación y ejemplo, no pudiera caminar por las fronteras de la vida.

A mi padre Gustavo, que decir de papi, le debo lo que soy, porque siempre ha sido un ejemplo para mí, aunque él no lo sepa.

A mi hermano Gustavito, por amarnos y vernos crecer como personas de bien, te quiero mucho mi manito.

A mi tita Mirian, mi amor más lindo, solo quiero ser como voz.

A mis abuelos Mario, Manolo y Raúl, mis abuelitos queridos, por siempre estar en las tardes de lluvias, contándome las historias más lindas, por transmitirme su amor y ser mi fuerza para crecer.

A mis abuelas Teresa, Fela y Sonia, por ser divinas, mis abus, sin tener nada lo dan todo.

A mi amado vecino Ramón, que siempre fue y es el abuelo de mi vida.

A mis primas, Diane que desde pequeña sabemos lo que es la inocencia de un niño, Yanet que me hiciste ver la vida en colores, te amo, eres lo más sencillo de la vida, Jathe tu eres única, nadie es más fuerte que voz, todo mi amor es para ti.

A mi primo Michel, quien es grande, no por ser alto, sino porque llevas el niño que yo siempre voy adorar.

A mi madrina Karina, porque siempre me quiso como la hija que no tiene y es para mi una madre.

AGRADECIMIENTOS

A Dios, porque donde quiera que estoy, nunca me abandonas.

A mi pino del alma, porque cuando era niña, me enseñaste a conocer la bondad de la vida.

A mis profesores de la primaria, en especial Veranes y Ana Ivis, por ver la niña que siempre voy a llevar dentro.

A mis dos amores Luz María, eres lo más dulce que he conocido e Irene por ser una abuelita más.

A mi hermana Sonia porque te quiero.

A mis primas Yudith por ser la primera universitaria, eres ejemplo para todas nosotras, Jaine por ser tan especial, Yesica mi pequeña rebelde.

A mis tíos Ruly porque me enseñas ver la vida siempre con objetivos y Sergio que donde quieras que este te llevo en mi corazón.

A mi tutora Yordanys, por ser paciente, porque tu voluntad es incomparable, nunca voy a olvidarte.

A las personas que hicieron posible que yo pudiera terminar este trabajo de diploma, por ser tan especiales conmigo, Arniel por tu amor de siempre, Eduanys te adoro, Membri tu personalidad me encanta, Marcel mi pedacito de cielo, Raciél te admiro, Kenier en el poco tiempo que te conozco me has conquistado, Félix mi cosita linda, Olía por demostrar que la chicas sí podemos.

A mis amigas del alma Baby, Maidelyn, Yanet Roca y Mary con las que he compartido los mejores momentos de la universidad, gracias por estar en las buenas y en las malas, las quiero.

A mis amigos de siempre, Dainovy, Dorgis, Alejandro, Yoandri, Yanary, Rosalía, Yanet Hernández, Aliuska, José, Danilo, Wilfredo, donde quiera que estemos siempre nos vamos a querer, felicidades porque llegamos a cumplir el sueño de todo universitario.

Al amor de mi vida Yasmani, mi Yas, por ser mi amigo incondicional, porque ni la distancia nos separa.

A mis amigos Viñolo, Jorge, Lisy, Maidy, Sucef, Suleidy, Yolaine con las que he convivido todo este tiempo, gracias por aguantarme.

A mis amigas de la infancia Indira, Mercy, Baby, las mimis Daily y Yanelys, las mellisas Ailen y Arlenis, Darlenis, porque con ellas he pasado los mejores momentos de mi vida.

A mi familia de la iglesia, Arturo, Lisandra, Viviana, Alinita, Roxy, porque me enseñaron a crecer en el amor y por siempre recordarme quien soy.

A mis vecinos Ritica, Magalys, Gloria y su tropa, Vigilia, por quererme desde siempre.

A los mejores amigos de papi, los pescadores, mi padrino Lencho, Silverio y el hijo de Anita, por enseñarme qué es la amistad.

A las mejores amigas de mami Dorgis, Marlenys, Mery, Naisa, por querer a mi mami.

Al tribunal, especialmente a mi oponente, por ser uno de los mejores profesores que he conocido, gracias por sus recomendaciones.

A mi príncipe Albert, por hacerme tan feliz, en tan poco tiempo.

A mis amigos de la radio, porque siempre han confiado en mí.

A mi tutor de la ayudantía Dieter, porque no solo es mi tutor, sino que además es mi amigo y confidente.

RESUMEN

La revolución tecnológica en la cual se encuentra inmersa la sociedad actual, va dirigida a cambiar los paradigmas y la forma de pensar del hombre. Un Sistema de Información Geográfica (SIG), es un nuevo modelo que ha revolucionado el campo de la cartografía, la geografía y la forma de visualizar el mundo. En el presente trabajo de diploma se realiza la propuesta de un módulo para la generación de las vistas de los mapas para la plataforma GeneSIG, que permite simplificar el proceso de búsqueda por temas asociados a una ubicación geográfica y compartir las vistas de los mapas entre los usuarios registrados en la aplicación.

Se aborda sobre los métodos científicos aplicados para el desarrollo de la investigación, así como las aplicaciones existentes en el mundo que responden al objeto de estudio. Se definen las tecnologías, metodologías y lenguajes de programación para la construcción del subsistema. Se identifican los requisitos que darán repuesta a las necesidades funcionales por parte del cliente, se construye el esqueleto arquitectónico haciendo uso del patrón basado en componentes y los patrones de diseño “Patrones de Asignación de Responsabilidades” (GRASP). Se realizan los modelos de diseño, de datos, de implementación y de prueba, que permiten tener una mayor visión del sistema a implementar. Como finalidad del mismo se hace la propuesta de un plugin que permite gestionar las vistas de los mapas para dar respuesta a la situación problemática planteada.

PALABRAS CLAVES

Autenticación, Búsqueda, Mapa, SIG, Ubicación, Vistas de los mapas.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA DEL PROCESO DE GESTIÓN DE LAS VISTAS DE LOS MAPAS.	5
INTRODUCCIÓN	5
1.1 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	5
1.2 OBJETO DE ESTUDIO	8
1.2.1 ESTRUCTURA Y FUNCIONAMIENTO DE LOS SIG	8
1.3 DESCRIPCIÓN DEL DOMINIO DEL PROBLEMA	11
1.4 ESTUDIO DE LAS APLICACIONES EXISTENTES	12
1.4.1 SIG DE ESCRITORIO	12
1.4.2 CLIENTES WEB SIG	14
1.5 ANÁLISIS DE LAS SOLUCIONES EXISTENTES	15
CONCLUSIONES PARCIALES	16
CAPÍTULO II. TENDENCIAS Y TECNOLOGÍAS ACTUALES.....	17
INTRODUCCIÓN	17
2.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	17
2.1.1 PROCESO UNIFICADO DE DESARROLLO.....	18
2.2 LENGUAJE DE MODELADO.....	19
2.2.1 LENGUAJE UNIFICADO DE MODELADO (UML)	19
2.3 HERRAMIENTAS CASE.....	20
2.3.1 VISUAL PARADIGM FOR UML	20
2.4 LENGUAJES DE PROGRAMACIÓN	21
2.4.1 HYPERTEXT PRE-PROCESSOR (PHP)	22
2.4.2 LENGUAJE DE MARCADO DE HIPERTEXTO (HTML)	23
2.4.3 HOJA DE ESTILO EN CASCADA (CSS)	23

2.4.4 JAVASCRIPT	24
2.5 ENTORNO DE DESARROLLO INTEGRADO (IDE).....	24
2.5.1 IDE NETBEANS	24
2.6 FRAMEWORK O COMPONENTES.....	25
2.6.1 CARTOWEB	25
2.7 SERVIDOR WEB.....	26
2.7.1 APACHE.....	26
2.8 SISTEMA GESTOR DE BASES DE DATOS	27
2.8.1 POSTGRESQL COMO SERVIDOR (SGBD)	27
2.8.2 PGADMIN COMO CLIENTE PARA POSTGRESQL.....	28
2.9 SERVIDOR DE MAPAS	28
2.9.1 MAPSERVER.....	29
CONCLUSIONES PARCIALES	29
CAPÍTULO III. PRESENTACIÓN DEL MÓDULO VISTAS DE LOS MAPAS	30
INTRODUCCIÓN	30
3.1 MODELO DE DOMINIO	30
3.1.1 REPRESENTACIÓN DEL MODELO DE DOMINIO	30
3.2 REQUISITOS	33
3.2.1 ESPECIFICACIÓN DE LOS REQUISITOS FUNCIONALES.....	33
3.2.2 ESPECIFICACIÓN DE LOS REQUISITOS NO FUNCIONALES.....	35
3.3 DESCRIPCIÓN DEL MODELO DE CASOS DE USO DEL SISTEMA	37
3.3.1 DESCRIPCIÓN DE LOS ACTORES QUE INTERACTÚAN CON EL SISTEMA.....	37
3.3.2 IDENTIFICACIÓN DE LOS CASOS DE USO	37
3.3.3 DIAGRAMAS DE CASOS DE USO DEL SISTEMA (DCUS)	38
3.3.4 DESCRIPCIÓN EXTENDIDA DE LOS CASOS DE USO DEL SISTEMA	38

CONCLUSIONES	46
CAPÍTULO IV. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	47
INTRODUCCIÓN	47
4.1 ARQUITECTURA DE SOFTWARE	47
4.1.1 PATRONES ARQUITECTÓNICOS	47
4.1.2 PATRONES DEL DISEÑO	48
4.2 DIAGRAMA DE PAQUETES	49
4.3 MODELO DE DISEÑO.....	50
4.4 MODELO DE DATOS.....	53
4.5 MODELO DE DESPLIEGUE	54
4.6 MODELO DE IMPLEMENTACIÓN	55
4.7 PRUEBAS	56
4.7.1 DISEÑO DEL CASO DE PRUEBA PARA EL CASO DE USO: GESTIONAR VISTA DEL MAPA.....	58
CONCLUSIONES PARCIALES	65
CONCLUSIONES GENERALES.....	66
RECOMENDACIONES	67
REFERENCIAS BIBLIOGRÁFICAS.....	68
BIBLIOGRAFÍA	69



ÍNDICE DE TABLAS Y FIGURAS

Tabla 1 Descripción de los Actores del Sistema.....	37
Tabla 2 Casos de Uso.....	38
Tabla 3 Descripción del caso de uso "Gestionar Vista del Mapa"	45
Tabla 4 Diseño de Caso de Prueba del "CU Gestionar Vista del Mapa"	61
Tabla 5 Descripción de las variables	61
Tabla 6 Matriz de Datos "SC 1: Adicionar vista del mapa"	62
Tabla 7 Matriz de Datos "SC 2: Modificar vista del mapa"	63
Tabla 8 Matriz de datos "SC 3: Eliminar vista del mapa"	64
Figura 1 Componentes de un SIG.....	10
Figura 2 Modelo Cliente-Servidor	21
Figura 3 Modelo de Dominio	31
Figura 4 Diagrama de Casos de Uso.....	38
Figura 5 Diagrama de Paquetes	50
Figura 6 Diagrama de Clases del Diseño	52
Figura 7 Modelo Entidad-Relación.....	53
Figura 8 Modelo de Despliegue	55
Figura 9 Diagrama de Componentes.....	56



INTRODUCCIÓN

El hombre a través de la historia, ha sentido la necesidad de resolver las disímiles interrogantes del entorno en que se desarrolla. La creación del mapa como un medio propicio para representar la superficie terrestre sobre un plano, la búsqueda y visualización sobre éste, permitió dar respuesta a muchas de estas incógnitas, como por ejemplo: ¿qué es lo que se desea encontrar?, ¿cómo se puede encontrar? y ¿cuál es la finalidad de lo que se busca? Precisamente, la necesidad de obtener la información geográfica contenida en el mapa de una forma más rápida y precisa, así como resolver problemas para tomar decisiones sobre una situación dada, motivaron al hombre a buscar mecanismos dirigidos a mejorar la forma de visualización y búsqueda sobre el mapa.

En la actualidad, con los revolucionarios cambios en las Tecnologías de la Información y las Comunicaciones (TIC), se han creado herramientas informáticas, que permiten responder a las necesidades de gestión de la información geográfica, de búsqueda y visualización sobre el mapa. Un ejemplo de estas son “los Sistemas de Información Geográfica (SIG), que son aplicaciones que integran tecnología informática, personas e información geográfica con la función de capturar, analizar, almacenar, editar y representar datos georreferenciados” (Oyala, 2010).

Los SIG están integrados por los datos que identifican la información geográfica que se visualiza, los métodos que proporcionan formulaciones y metodologías que se aplican sobre los datos, el software que es la aplicación informática que permite que puedan ser analizados, interpretados y visualizados los datos por medio de los métodos, el hardware que permite ejecutar el software y las personas que son la clave para el trabajo con la aplicación. De esta forma, se puede concluir que un SIG es la herramienta idónea para buscar y visualizar determinada ubicación geográfica en el mapa.

El desarrollo de los SIG se ha convertido en materia de estudio y aplicación en muchos de los países del mundo, en los cuales se han creado instituciones que proveen la construcción de estos sistemas como Environmental Systems Research Institute (ESRI), Intergraph, Mapinfo, Bentley Systems, Autodesk o Smallworld.

En Cuba, existen diversas instituciones que tienen entre sus proyectos la construcción de Sistemas de Información Geográfica; muestra de lo expresado es la Universidad de las Ciencias Informáticas (UCI), centro de estudios universitarios que contribuye al progreso de la economía del país a través del desarrollo de productos informáticos, tarea en la que se encuentran inmersas cada una de sus facultades. En la facultad 6 se encuentra el centro de desarrollo de software Geoinformática y Señales Digitales (GEYSED), que abarca entre sus proyectos la creación de la plataforma GeneSIG, aplicación



informática que es utilizada como base cartográfica para la construcción de SIG con perfiles más específicos.

La plataforma GeneSIG proporciona servicios de acceso a la información geográfica para su consulta, análisis y visualización y brinda altos niveles de escalabilidad. Dentro de los múltiples servicios que brinda la aplicación, le permite al usuario realizar una búsqueda de determinada ubicación geográfica, utilizando las herramientas que provee el sistema. Un ejemplo es el zoom ampliado, a través del cual se obtiene una vista del mapa, que es el contorno de visualización de la ubicación seleccionada. Sin embargo, la plataforma no posee mecanismos para guardar la información de la búsqueda realizada. Es por ello, que cuando el usuario desee en otro momento, obtener la misma ubicación geográfica, debe ejecutar los mismos procedimientos de búsqueda; incurriendo en tener que ocupar un tiempo para volver obtener la misma vista del mapa y en ocasiones, la vista no se obtiene con la misma exactitud que la anterior.

A partir de la situación descrita con anterioridad, se puede identificar el siguiente **problema a resolver**: ¿Cómo contribuir a la simplificación del proceso de búsqueda por temas asociados a una ubicación geográfica en la Plataforma Soberana GeneSIG?

Para darle solución al problema a resolver se precisa el siguiente **objetivo general**: Desarrollar un subsistema que permita gestionar las vistas de los mapas en la Plataforma Soberana GeneSIG.

El problema descrito genera como **objeto de estudio** de esta investigación: el proceso de gestión de las vistas de los mapas en los Sistemas de Información Geográfica, enmarcando el proceso de gestión de las vistas de los mapas en la Plataforma Soberana GeneSIG como campo de acción.

Con el propósito de dar cumplimiento al objetivo general se proponen las siguientes **tareas de la investigación**:

1. Identificación de los diferentes patrones de solución existentes en las aplicaciones SIG.
2. Análisis y selección de las tecnologías que permiten desarrollar el plugin vistas de los mapas.
3. Realización de la documentación técnica correspondiente al Modelo de Dominio para lograr una familiarización con el entorno de trabajo que actualmente existe.
4. Descripción de los requisitos que debe cumplir el plugin vistas de los mapas.
5. Descripción del Modelo de Casos de Uso del plugin vistas de los mapas.
6. Descripción del Modelo de diseño del plugin vistas de los mapas.
7. Descripción del Modelo de Implementación del plugin vistas de los mapas.



8. Descripción del Modelo de Despliegue del plugin vistas de los mapas.
9. Implementación del plugin vistas de los mapas.
10. Realización de las pruebas de caja negra correspondientes al plugin vistas de los mapas.

A partir de la correcta ejecución de las tareas de la investigación se propone la siguiente **idea a defender**: Con la implementación de un subsistema que permita gestionar las vistas de los mapas, se reduce el tiempo de búsqueda por temas asociados a una ubicación geográfica sobre la plataforma GeneSIG.

Durante el desarrollo de la investigación se han utilizado un conjunto de métodos científicos para la obtención, procesamiento y las conclusiones. Los cuales se mencionan a continuación:

Métodos Teóricos:

1. **Análisis Histórico – Lógico:** Se aplica con el objetivo de conocer la existencia de Sistemas de Información Geográfica, que realicen el proceso de gestión de las vistas de los mapas, para tener una mayor visión de la forma de ejecución de estos procesos, en este tipo de sistema y utilizar estos como puntos de referencia.
2. **Analítico – Sintético:** Se aplica con el objetivo de analizar todos los documentos relacionados con el contenido de los Sistemas de Información Geográfica y el proceso de gestión de las vistas de los mapas, recopilando toda la información disponible y sintetizando la misma para una mejor comprensión del tema.
3. **Método Modelación:** Se aplica para generar los artefactos que se construyen durante el proceso de ingeniería de software, como modelo para comprender las propiedades, funcionalidades y relaciones que contiene el plugin vista de los mapas en la plataforma GeneSIG.

Métodos Empíricos:

Método entrevista: Se aplica para obtener información sobre las funcionalidades que debe tener el plugin que provee la gestión de las vistas de los mapas, la cual es realizada de forma individual a profesionales que están vinculados directamente con el desarrollo de la plataforma.

Se espera alcanzar con el desarrollo de la investigación, la construcción de un subsistema que permita gestionar las vistas de los mapas por temas asociados en la plataforma GeneSIG, así como la documentación técnica del proceso ingenieril del plugin vistas de los mapas a implementar.



El presente trabajo de diploma se encuentra estructurado en cuatro capítulos que estarán guiados por los siguientes temas a tratar:

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA DEL PROCESO DE GESTIÓN DE LAS VISTAS DE LOS MAPAS: Está centrado en el análisis de los elementos relacionados con la generación de las vistas de los mapas. Se realiza una presentación del estado del arte en el mundo de las aplicaciones SIG que responden al objetivo general.

CAPÍTULO II. TENDENCIAS Y TECNOLOGÍAS ACTUALES: Está centrado en el análisis de las metodologías, lenguaje de modelado y herramientas CASE para la construcción de los artefactos correspondientes a cada uno de los flujos de trabajo, así como, en el estudio de las tecnologías que responden a las necesidades del proceso de construcción del sistema y el ambiente de desarrollo.

CAPÍTULO III. PRESENTACIÓN DEL MÓDULO VISTAS DE LOS MAPAS: Está centrado en la obtención y descripción de los requisitos funcionales y no funcionales del sistema a construir, en la identificación y descripción de los casos de uso del mismo y en la estructuración del Modelo de Casos de Uso del Sistema (MCUS).

CAPÍTULO IV. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA: Está centrado en la definición de las clases del diseño, sus relaciones, y estructuración de los diagramas de clases del diseño. Se elabora el modelo de datos, de implementación y de despliegue. Luego, se realiza la presentación de las pruebas de caja negra utilizando la técnica de partición de equivalencia.

Para finalizar se presentan las conclusiones, las recomendaciones, las referencias bibliográficas, la bibliografía, un glosario de términos y el conjunto de anexos para un mejor entendimiento de lo expuesto en el trabajo.



CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA DEL PROCESO DE GESTIÓN DE LAS VISTAS DE LOS MAPAS.

INTRODUCCIÓN

El ser humano cuando realiza determinada actividad social, tiene en cuenta muchos factores que conducen al desarrollo de la misma. En la gran mayoría de los casos, divide la actividad general en tareas más pequeñas con el propósito de obtener un resultado final. De esta forma, se puede definir un proceso, como una secuencia de pasos ordenados de manera lógica, que tienen un inicio, fin y un valor agregado.

Con la idea antes expuesta se da inicio al contenido del presente capítulo, en el cual se realiza la descripción y análisis de los elementos que sirven de fundamento teórico para la investigación presentada. Se identifican los conceptos que permiten un mejor dominio del problema, enmarcándose en las características fundamentales de los SIG. Se realiza un análisis del problema que conduce a la elaboración del presente estudio, haciendo referencia a las tecnologías y herramientas, que brindan una posible solución del mismo.

1.1 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA

En este apartado, se refieren conceptos básicos que están relacionados con el proceso que se estudia, para lograr una mejor comprensión del problema y los factores que están vinculados al mismo. Los conceptos que se citan a continuación son: autenticación, búsqueda, capas del mapa, mapa, mapa de un SIG, ubicación, vista de los mapas.

Autenticación

Es el proceso por el cual el sistema valida la información de inicio de sección de usuario. El nombre y la contraseña del usuario se comparan con una lista autorizada, y si el sistema detecta un acierto, el acceso se concede hasta el límite especificado en la lista de permiso para ese usuario. (Corporation, 2005)

Este concepto es aplicado, de manera que permite asociar al usuario con la vista del mapa, es decir, cuando el usuario es autenticado y crea diversas vistas de los mapas, le son correspondidas las vistas creadas.



Búsqueda

Es el proceso de buscar un determinado archivo o un dato específico. La búsqueda la realiza un programa comparando o realizando cálculos para determinar si existe alguna coincidencia con el patrón introducido o si cumplen los criterios especificados. (Corporation, 2005)

Este concepto es aplicado, desde que el usuario realiza una búsqueda de una ubicación geográfica en el mapa, asociada a un determinado tema que sea de su interés. Además permite que a través de criterios, como los atributos de las vistas de los mapas, buscar todas las vistas que cumplen con los mismos.

Capas del mapa

Las capas del mapa son representaciones temáticas de información geográfica, como transporte, agua, elevación. Dentro de cada capa del mapa, se utilizan símbolos, colores y texto para representar información importante que describe cada uno de los elementos geográficos individuales. Las capas de mapas ayudan a transmitir información mediante lo siguiente:

- Entidades discretas como colecciones de puntos, líneas y polígonos.
- Símbolos de mapa, colores y etiquetas que ayudan a describir los objetos del mapa.
- Superficies continuas, como elevación, que se puede representar de varias formas, por ejemplo, como una colección de líneas de curvas de nivel y puntos de elevación o como relieve sombreado.

Este concepto es aplicado, porque es uno de los componentes más importantes de los mapas, que en dependencia de la vista, pueden estar incorporadas todas o algunas de las capas establecidas en la plataforma.

Mapa

Representación geográfica de la Tierra, en su totalidad o en parte, sobre una superficie plana. (Corporation, 2005)

El mapa va a permitir dentro del objeto de estudio, representar la vista, a partir de las propiedades métricas que posee. Posibilita obtener medidas de distancia, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto, que permitirá localizar la vista.



Mapa de un SIG

El mapa de un SIG es una representación de información espacial o geográfica en forma de una serie de capas temáticas de información para un área de interés. El marco del mapa proporciona la vista geográfica de información, mientras que otros elementos como leyendas de símbolos, barras de escala, flecha norte, texto descriptivo y título del mapa ayudan a entender, leer e interpretar el contenido del mapa. Además, el mapa de SIG representa colecciones lógicas de información geográfica como capas de mapas, constituye el núcleo de cómo se utiliza un SIG y proporciona una representación eficaz para planear y organizar la información geográfica.

Los mapas de un SIG contienen una cartografía sólida y datos geográficos (entidades de punto, línea y poligonales), que se utilizan para generar el mapa junto con herramientas analíticas para obtener resultados. Presentan características como:

- Tienen una representación cartográfica.
- Pueden incluir entidades, rasters y datos del terreno.
- Pueden permitir la edición, para compilar nueva información.
- Se componen de capas de mapa, que se pueden combinar y utilizar de maneras diferentes.
- Pueden contener los resultados de modelos de análisis y consultas a sistemas de información especializados.
- Son interactivos y permiten ir más allá del mapa para utilizar información y herramientas más eficaces.

Se hace referencia a este concepto, precisamente porque se necesita conocer las características que presentan los mapas en los Sistemas de Información Geográfica, que se diferencian completamente de los que se encuentran en una superficie plana, para el posterior trabajo que se va a realizar con los mismos.

Ubicación

Ubicación es el lugar en que está ubicado algo o la acción y efecto de ubicar (situar, localizar o instalar en determinado lugar o espacio). El término puede asociarse a un cierto espacio geográfico. (WordPress, 2008).

El término de ubicación es aplicado cuando se hace referencia a la ubicación en el mapa de un lugar determinado, que el usuario desea buscar.



Vista del mapa

Una vista es una visualización de datos o de imágenes desde una perspectiva o ubicación determinada. En sistemas de administración de bases de datos relacionales, es una tabla lógica creada mediante la especificación de una o más operaciones relacionales ejecutadas sobre una o más tablas. Una vista es equivalente a una relación dividida en el modelo relacional. (Corporation, 2005).

Este concepto es clave, porque representa lo que se desea gestionar. Una vista del mapa es el contorno de visualización, de una búsqueda realizada por el usuario de determinada ubicación geográfica, a partir de un tema de interés que le desee asociar a la ubicación seleccionada.

1.2 OBJETO DE ESTUDIO

Las vistas de los mapas representan una visualización del mapa a una escala definida, con el objetivo de obtener información geográfica sobre una superficie terrestre. Cuando se refiere a una vista del mapa sobre una plataforma SIG, se debe tener en cuenta que cambia el entorno de visualización y el tratamiento con los mapas, por lo que se introducen nuevos mecanismos para gestionar la vista geográfica de un mapa contenido en el SIG.

El proceso de gestión de las vistas de los mapas se caracteriza por una previa selección de la ubicación geográfica en el mapa SIG, teniendo la posibilidad a partir de las herramientas que proporciona el sistema, de realizar modificaciones a la ubicación geográfica seleccionada, ya sea agregándole propiedades u obteniendo información geográfica a partir de ella, como la proyección y la escala. Seguidamente, se puede registrar el mapa que identifica la vista seleccionada con las modificaciones correspondientes. También se pueden realizar otras operaciones que se incluyen en el proceso como redefinir la vista seleccionada, eliminar la vista que ha sido registrada en el sistema, localizar la vista en el sistema, entre otras.

Los Sistemas de Información Geográfica son aplicaciones complejas que dada sus propiedades estructurales y funcionales, permiten incorporar procesos como el descrito con anterioridad, por lo que es importante conocer el entorno de desarrollo de los mismos, con el objetivo de tener una visión general de las implicaciones y ventajas que traen consigo la incorporación de estos procesos.

1.2.1 ESTRUCTURA Y FUNCIONAMIENTO DE LOS SIG

El análisis de los conceptos, estructura y funcionamiento de los SIG, permite conocer sobre el entorno de desarrollo estos sistemas, identificar cuáles son las propiedades básicas que permitirán integrar nuevas funcionalidades y la forma que pueden ser integradas.



Concepto de SIG

Entre los autores que realizaron un estudio sobre los SIG, se encuentra Tor Bernhardsen, quien define este sistema como “una aplicación basada en el uso de la computadora, para la manipulación de datos geográficos, que incluye tanto hardware como software, especialmente los elementos para introducir y crear mapas, así como los sistemas de comunicación necesarios para relacionar diversos elementos”. (Bernhardsen, 2001) Además, en el sitio web **gis.com** se incorporan nuevos conocimientos a este concepto. Ya no solo un SIG va a estar compuesto por hardware y software, sino además por personas y datos, y va a permitir capturar, administrar y editar la información geográficamente referenciada. (ESRI, 1995)

En el diccionario de informática e internet de Microsoft, se define un SIG como “una aplicación o un conjunto de aplicaciones para visualizar y crear mapas. Normalmente, los Sistemas de Información Geográfica contienen un sistema de visualización (algunas veces permiten a los usuarios visualizar los mapas con un explorador web), un entorno de creación de mapas y un servidor para administrar mapas y datos para la visualización en línea de tiempo real”. (Corporation, 2005)

El autor Yury Quiroz Ortuño menciona que “los SIG cumplen de manera general con los requerimientos tanto en el apoyo como en el soporte en la toma de decisiones.”

A través del estudio realizado sobre los diferentes conceptos de SIG, como los que se proponen con anterioridad, se puede definir un Sistema de Información Geográfica como una aplicación informática, integrada por datos, hardware, software, métodos y personas, que permite analizar, capturar, almacenar, manipular y desplegar la información geográficamente referenciada en todas sus formas, así como crear, visualizar y administrar mapas en tiempo real. Además brinda la posibilidad a cualquier usuario de resolver problemas de planificación, gestión y tomar decisiones sobre una situación dada.

Estructura de un SIG

Las tecnologías SIG se encuentran en evolución y su estructura se debe ir perfeccionando a medida que se le van incorporando nuevos componentes. Es por ello, que David Andes tiene una visión más general de los elementos que integran un SIG, realizando una combinación de los componentes que se conocen como hardware y software en tecnologías, manteniendo los datos y las personas, e incorporando nuevos elementos a la estructura de un SIG, como los que se muestran en la Figura 1.



Figura 1 Componentes de un SIG

Los **conceptos geográficos básicos** son el núcleo principal de los restantes componentes, producto a la necesidad de tener un conocimiento general de la geografía para la construcción de los SIG. La **tecnología** refiere a la necesidad de combinar el hardware y el software, la incorporación de nuevas tecnologías como administradores de sistemas, administrador de bases de datos, programación y unidades GPS. Los **datos** incluyen cualquier información que se refiere a la geografía, ya sea parcelas, estadísticas, rutas, etc. Las cuales deben de tener buena calidad y exactitud. Los **campos de aplicación** refieren a las aplicaciones de los SIG en los diferentes sectores sociales y la necesidad de obtener información especializada sobre el mismo. Los **métodos** representan las formulaciones que pueden ser sencillas o complejas, desde una simple consulta o medición a elaborados modelos que emplean datos de variables muy numerosas y proyecten resultados complejos. La **visualización** es la representación de los datos para el usuario final, puede tomar formas de mapas, gráficos, tablas, animaciones y simulaciones.

Información geográfica

La información geográfica representa los datos espaciales georreferenciados, se divide en dos componentes principales: el componente espacial que hace referencia a la posición dentro de un sistema de referencia establecido, respondiendo a la pregunta ¿dónde? y la componente temática que establece la naturaleza del fenómeno, que tiene lugar a partir de la localización establecida por la componente espacial.

La información geográfica se puede dividir en dos tipos de división: la horizontal y la vertical. La horizontal permite no sólo dividir la información geográfica en componentes, sino también en criterios



puramente espaciales, separándola en unidades menores que ocupan una región de amplitud más reducida, es decir, se combinan distintos mapas para formar uno que cubra una extensión más amplia. La división vertical de la información o en capas permite dividir la información espacial referida a una zona de estudio de varios niveles, de tal forma, que al coincidir sobre un mismo emplazamiento, la información sobre distintas variables se encuentra recogida de forma independiente. Es decir, en función de la componente temática se establecen distintos bloques de datos espaciales.

1.3 DESCRIPCIÓN DEL DOMINIO DEL PROBLEMA

La plataforma GeneSIG es una herramienta informática que provee el desarrollo de productos SIG aplicados a diversos entornos sociales, formada por un grupo de especialistas de GEOCUBA, de las Fuerzas Armadas Revolucionarias (FAR) y de la Universidad de Ciencias Informáticas. Proporciona servicios de acceso a la información geográfica para su consulta, análisis y visualización. Permite integrar la información raster existente (imágenes de satélites, mapas escaneados) con información vectorial, editar la geometría, así como las capas de un mapa y exportar en formato pdf las vistas de los mapas.

GeneSIG como todo producto informático, es creado y pensado a partir de las necesidades del usuario final. Este usuario desde que es autenticado en el sistema, visualiza un mapa SIG o una ubicación geográfica en el mapa que proporciona la plataforma, sobre el cual puede realizar diferentes operaciones, a partir de los subsistemas que componen la plataforma. Un subsistema está integrado por un plugin que provee un conjunto de funcionalidades específicas, en correspondencia con la operación que debe de realizar el subsistema. Cuando el usuario realiza estas operaciones define un conjunto de propiedades que están implícitas o que son incorporadas a la vista del mapa seleccionado. El usuario puede realizar esta actividad con diversas vistas de los mapas o diferentes ubicaciones geográficas.

Las vistas de los mapas, con las propiedades que tienen implícitas, que han sido consultadas por el usuario, no son registradas en el sistema, por lo que el usuario para obtener nuevamente la vista del mapa que había seleccionado con anterioridad, debe realizar las mismas operaciones que realizó inicialmente, lo que implica un proceso de búsqueda muy extenso y que en muchos de los casos la misma ubicación geográfica, no se obtenga con la misma exactitud. En correspondencia con la situación planteada, se necesitan encontrar mecanismos para simplificar este proceso de búsqueda, de una ubicación geográfica en el mapa SIG, en correspondencia, con un tema de interés que el usuario desee asociar a la ubicación seleccionada.



1.4 ESTUDIO DE LAS APLICACIONES EXISTENTES

El acceso al conocimiento geográfico más fácil y disponible, ha encauzado a muchos especialistas del mundo, a buscar alternativas para mejorar las tecnologías SIG y con ello el tratamiento con los mapas, que son la fuente de la información geográfica. La forma de crear, mostrar, acceder, compartir e interactuar con un mapa SIG, está en dependencia de las características específicas de un sistema. Es de interés conocer durante la investigación, la capacidad que tienen los SIG de realizar estas operaciones, para ello se hace necesario realizar un análisis de las propiedades de estos y las ventajas que proporciona su uso, como solución a la simplificación del proceso de búsqueda por temas asociados a una ubicación geográfica.

Para la realización del análisis, se tienen en cuenta las diferentes formas de representación de las tecnologías SIG, específicamente los SIG de escritorio y como clientes web.

1.4.1 SIG DE ESCRITORIO

Se utilizan para crear, editar, administrar, analizar y visualizar los datos geográficos. Se pueden representar como Visor SIG software que pueden ser sencillos, permitiendo desplegar la información geográfica a través de una ventana que funciona como visor y donde se pueden agregar varias capas de información; como editor SIG el cual es un software orientado al tratamiento previo de la información geográfica para su posterior análisis; como un SIG de análisis que disponen de funcionalidades de análisis espacial y modelización cartográfica de procesos.

gvSIG Desktop

Es un potente Sistema de Información Geográfica libre, diseñado para dar solución a todas las necesidades relacionadas con el manejo de información geográfica. Se caracteriza por ser una solución completa. Es capaz de acceder a los formatos más comunes, como vectoriales, rasters, locales y remotos, integra estándares de Open Geospatial Consortium (OGC), y cuenta con un amplio número de herramientas para trabajar con información de naturaleza geográfica (consulta, creación de mapas, geoprocésamiento, redes, etc.) que lo convierten en una herramienta ideal para usuarios que trabajen con la componente territorial. Funciona en distintas plataformas Linux, Windows y Mac OS. El lenguaje de programación es Java. Se distribuye bajo la licencia GNU/ GPL, lo que permite su libre uso, distribución, estudio y mejora.

En gvSIG Desktop se pueden encontrar un amplio abanico de funcionalidades, es de interés conocer cómo este permite realizar los procesos de generación de las vistas de los mapas, a partir de una descripción concreta de la gestión de los mapas que realiza el sistema. Primeramente, se inicia la



aplicación donde aparecerá una interfaz Gestor de proyectos, que por defecto permite realizar las operaciones de crear mapas, vistas y tablas (Ver anexo 1.1).

Con el propósito de crear una vista del mapa, se selecciona la opción (Mapa), luego se introduce el nombre del mapa a crear, a partir de la opción (Nuevo) que proporciona el sistema en la interfaz gestor de proyecto; seguido se genera en un listado el nuevo mapa creado y se visualiza en esta interfaz. Con las herramientas que proporciona el sistema se puede insertar en el mapa, una vista que contiene atributos como capas, formatos, sistemas de referencias, simbología y etiquetado que refiere al color, las letras, etc. Además, se puede incorporar al mapa propiedades como líneas, puntos, polígonos, leyenda, escala, texto, etc., de esta forma se genera una vista del mapa. El sistema permite además, que se puedan concebir otras vistas a partir del procedimiento descrito; eliminar, visualizar y modificar una vista seleccionada.

ArcGIS Desktop

Es un complejo Sistema de Información Geográfica para compilar, usar y administrar la información geográfica. Incluye un conjunto de aplicaciones: ArcMap, ArcCatalog, ArcGlobe, ArcScene, ArcToolbox y ModelBuilder, que admiten diversas tareas SIG, incluidas la representación cartográfica, la compilación de datos, el análisis, la administración de geodatabases y el uso compartido de información geográfica. Es compatible con diversos sistemas operativos como Windows, GNU/Linux y Unix. El lenguaje de programación es Java. Se puede usar en cualquiera de los tres niveles de producto ArcView, ArcEditor y ArcInfo. ArcView se distribuye bajo una licencia de uso individual, por lo que se puede instalar y usar cada copia de ArcView solamente en una máquina, pero también se puede obtener una licencia flotante, al igual que ArcEditor y ArcInfo. Esta licencia flotante ofrece a los usuarios de ArcGIS Desktop una gran flexibilidad, ya que permite instalar el software en todos los computadores que se quiera.

Las actividades claves que realizan los usuarios en ArcGIS Desktop son el trabajo con los mapas, análisis espacial y compilación de datos. Los mapas son esenciales, porque hacen que toda la información cobre vida y son el mecanismo utilizado para editar y proporcionar análisis espacial a muchos usuarios. Es de interés conocer sobre el trabajo con los mapas, porque refiere al estudio de la generación de las vistas de los mapas.

Cada mapa en ArcGIS Desktop es una especificación o diseño de cómo se utiliza, consulta, muestra, administra y analiza la información geográfica y las herramientas. Los mapas SIG se pueden capturar, guardar y compartir como documentos de mapas. Estos documentos y paquetes se pueden publicar como servicio web, si se utiliza ArcGIS Desktop junto con ArcGIS Server, se puede convertir cualquier mapa, geodatabase o modelo en un servicio web de SIG para compartirlo en un grupo de trabajo, en



toda la empresa o abiertamente en Internet (Ver en Anexo 1.2). Además, se pueden crear y guardar paquetes de mapas y paquetes de capas que reúnen todos los aspectos de SIG necesarios para la visualización y utilización de mapas, análisis, compilación de datos y administración. También, brinda la posibilidad de compartir paquetes de mapas o paquetes de capas con otros usuarios.

MapInfo Professional

Es una potente herramienta de Sistemas de Información Geográfica que permite realizar diversos y complejos análisis geográficos ideales para facilitar la toma de decisiones: captura, consulta, edición, análisis y reportes de información geográfica dinámicamente relacionada con bases de datos. Es compatible con Microsoft Windows 98, Windows 2000, Windows NT 4.0, Windows XP Professional y Windows XP Home y Office, GNU/Linux, no es compatible con Unix. Es un software propietario, que provee las licencias la propia institución.

MapInfo proporciona la creación de mapas por ordenador, donde se puede visualizar los datos como puntos, regiones zonificadas temáticamente, gráficos, etc. Puede llevar a cabo operaciones de zonificación, combinación y división de objetos, y definición de áreas de influencia. Es de interés conocer el trabajo con los mapas en MapInfo, porque refiere al estudio de la generación de las vistas de los mapas.

MapInfo tiene entre sus características que permite obtener numerosas vistas de datos en tres formatos: ventanas de mapas, listado y gráfico. Las ventanas de mapas permiten que se puedan crear mapas o editar mapas existentes, para ello se debe de ir al menú ARCHIVO, haciendo clic en ABRIR o seleccionado NUEVA VENTANA en el menú VENTANA (ver en anexo 1.3). Los mapas en MapInfo están formados por capas de objetos, a partir de cinco tipos básicos de objetos:

1. Regiones: objetos cerrados que cubren un área concreta. Éstos incluyen polígonos, elipses y rectángulos.
2. Objetos de puntos: representan ubicaciones simples de datos.
3. Objetos de línea: objetos abiertos que cubren una distancia determinada.
4. Objetos de texto: texto que describe un mapa u otro objeto, como etiquetas y títulos.
5. Objetos de recopilación: combinación de regiones, objetos de líneas y de multipuntos.

1.4.2 CLIENTES WEB SIG

Permite la visualización de datos y el acceso a funcionalidades de análisis y consulta de servidores SIG a través de Internet o Intranet. Generalmente se distingue entre cliente ligero y pesado. Los clientes ligeros (por ejemplo, un navegador web para visualizar mapas de Google) sólo proporcionan una funcionalidad de visualización y consulta, mientras que los clientes pesados (por ejemplo, Google



Earth o un SIG de escritorio) a menudo proporcionan herramientas adicionales para la edición de datos, análisis y visualización.

- **ArcGIS.com** es una aplicación web basada en un navegador que sirve para utilizar el contenido de ArcGIS on-line. Conecta a los usuarios con mapas web, aplicaciones web y servicios SIG. Incluye:
 - Una galería de mapas y aplicaciones web que publica la comunidad de ArcGIS, los cuales pueden utilizar en un navegador web y en dispositivos móviles sin tener que descargar el software del SIG especial (ver anexo 1.4).
 - Un visor de mapas basado en el navegador que puede utilizar para crear y compartir sus propios mapas web.
 - Una forma para que las comunidades de usuarios y grupos se organicen y compartan información (ver anexo 1.5).
 - Permite buscar mapas SIG e información que comparten en ArcGIS.com, que se ha distribuido desde un sitio web de ArcGIS Server o que está disponible en Internet (ver anexo 1.6).

Muchos profesionales de ArcGIS crean, guardan y comparten mapas e información geográfica con ArcGIS.com; esto permite a la comunidad de ArcGIS incluir cualquier usuario con acceso web para que descubran y utilicen este contenido compartido. No se requiere ningún software para utilizar este cliente web, solamente una conexión a Internet.

1.5 ANÁLISIS DE LAS APLICACIONES EXISTENTES

La tecnología gvSIG desktop presenta las características idóneas para resolver el problema planteado en la investigación, como por ejemplo que esta herramienta se distribuye bajo la licencia GPL, permitiendo su uso, la libre distribución y modificación, cumpliendo con la política de migración al software libre que impulsa el país, es compatible con los principales sistemas operativos como Windows y Linux. Sin embargo, la forma en que realizan el proceso de gestión de las vistas de los mapas, no responde a la necesidad que tiene la plataforma GeneSIG, porque ya en ésta, los mapas están creados, solamente se necesita guardar una ubicación geográfica que ha sido seleccionada por el usuario, mientras que en gvSIG a partir de la creación del mapa, es que se almacena la vista.

En el caso de ArcGIS desktop, es una potente herramienta que sí da respuesta a la necesidad de la plataforma. Incluso permite compartir carpetas con las vistas de los mapas entre varios usuarios, pero es muy costosa, es un software propietario y las licencias limitan una mejor distribución del sistema SIG en los ordenadores.



También ArcGIS.com muestra una solución al problema a resolver, pero en este caso se necesita tener acceso a Internet, limitante que imposibilita su uso.

El estudio de las tecnologías mencionadas con anterioridad, ha permitido tener una visión general de las características específicas de estas herramientas y cómo proveen de diferentes formas una solución en la búsqueda de una ubicación geográfica a partir de temas asociados. Además, ha permitido identificar si es posible hacer uso de estos sistemas SIG, con el objetivo de integrarlos a la plataforma GeneSIG.

CONCLUSIONES PARCIALES

Una tecnología que se ha creado en el pasado, es transformada en el presente y puede tener un nuevo paradigma en el futuro. Los Sistemas de Información Geográfica forman parte de estas tecnologías, que surgieron como una simple imagen asociada a una información y en la actualidad son complejos sistemas, que han revolucionado el pensamiento humano y la visión que tiene el hombre sobre el mundo que conoce. En el capítulo recién concluido, se realiza un análisis de las aplicaciones SIG, permitiendo elaborar una idea general de cómo incorporar nuevas funcionalidades en estos sistemas, teniendo en cuenta, su estructura y funcionamiento. Además, se realiza una disertación de la situación problemática planteada y del objeto de estudio, permitiendo arribar a las siguientes conclusiones:

1. La descripción de los conceptos identificados durante la investigación realizada, ha permitido profundizar en los conocimientos sobre las características de estos elementos y la forma en que son aplicados dentro del objeto de estudio.
2. El estudio de la estructura y funcionamiento de los SIG, ha permitido corroborar, que por las condiciones y cualidades que presentan estos sistemas, se puedan incorporar nuevas funcionalidades que den respuesta a la situación problemática planteada.
3. El estudio de las aplicaciones existentes a nivel internacional, aunque ninguna representa una solución al problema a resolver, por la característica de ser sistemas desktop y lo que se desea desarrollar debe ser aplicado a un sistema web, sirvió de base para elaborar una idea general de la forma en que es ejecutado el proceso de gestión de las vistas de los mapas, en los sistemas SIG y las ventajas que proporciona el desarrollo de un plugin que permita ejecutar dicho proceso.



CAPÍTULO II. TENDENCIAS Y TECNOLOGÍAS ACTUALES

INTRODUCCIÓN

La construcción de un módulo para gestionar las vistas de los mapas para la plataforma GeneSIG, requiere que sea organizado todo el proceso de desarrollo, con la finalidad de obtener un producto con calidad y que satisfaga las necesidades apremiantes del usuario final.

Teniendo en cuenta la idea antes planteada, se da pie al contenido del presente capítulo, en el cual se realiza una descripción de la metodología, lenguaje de modelado y herramienta CASE que son utilizadas en la elaboración de los artefactos ingenieriles y para organizar todo el proceso de desarrollo del módulo. Además son mencionados los lenguajes de programación, servidores web y sistemas gestores de bases de datos que forman parte de las tecnologías a utilizar, durante la construcción del sistema.

2.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo de software permiten conocer cuáles son las actividades y tareas que se deben ejecutar y en qué momento se deben realizar, los recursos que se necesitan y los productos que se deben obtener en el proyecto. Es de vital importancia seleccionar una metodología que muestre cómo se desarrollará el producto y cómo se trabajará con el cliente de la aplicación; selección que se debe llevar a cabo sobre la base de las características del flujo, profundidad de las descripciones, compromiso y participación del cliente en el desarrollo y calidad de las representaciones de los resultados.

Las metodologías permiten presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas, procedimientos y ayuda a la documentación, así como una visión general del producto, debe reflejar la vinculación con el cliente, establecer un modelo de ciclo de vida, un plan de desarrollo, integración del proyecto, medidas de progreso del proyecto, métricas para evaluar la calidad, maneras de medir el riesgo, cómo gestionar los cambios y establecer una línea base. Es por ello, que aunque sea muy complejo identificar una metodología que se ajuste eficientemente a las condiciones del equipo de desarrollo y del cliente, se debe realizar una correcta selección, ya que de ello depende la calidad de desarrollo del software y el resultado final del producto.

Si el proceso de desarrollo de software es complejo, exige de un mayor énfasis en la planificación y control del proyecto, así como en la especificación de los requisitos y modelado, se recomienda el uso de metodologías pesadas, orientadas al control de los procesos, estableciendo rigurosamente las



actividades a desarrollar, herramientas y notaciones a utilizar. Sin embargo, si el proceso de desarrollo no requiere de un mayor compromiso y complejidad, valorando mucho más el software que la documentación, al equipo de desarrollo que a los procesos y herramientas, a la colaboración total con el cliente que a la negociación contractual, se recomienda la utilización de metodologías ligeras o ágiles, orientadas al desarrollo incremental del software, interactuando con el cliente, para que éste pueda evaluar y sugerir cambios en el producto según se va desarrollando.

El proyecto de desarrollo de la plataforma soberana GeneSIG donde será integrado el módulo para la gestión de las vistas de los mapas, es complejo y de gran envergadura, existe una gran inestabilidad en el grupo de trabajo, lo que implica que se necesite una rigurosa definición de actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. El período de tiempo disponible para desarrollar el módulo es de 9 meses, con la facilidad de acceso a los recursos que requiere el mismo. Se solicita llevar un control de las tareas y actividades que son ejecutadas durante todo el proceso de desarrollo, ya que el cliente no se encuentra directamente vinculado al mismo.

Se ha determinado el uso de una metodología pesada, en este caso, Rational Unified Process (RUP), por el grado de conocimiento que se tiene sobre la misma, aplicada en un amplio abanico de proyectos, la generación de una documentación óptima, la adecuada planificación y control del proceso de desarrollo, la relación que establece con el Lenguaje Unificado de Modelado (UML), la facilidad de integración entre las etapas de desarrollo proporcionando una cobertura total de las mismas, el desarrollo de software sobre cualquier tecnología, así como las actividades que mejoren el mismo, brindando soporte al mantenimiento y a la reutilización de software.

2.1.1 PROCESO UNIFICADO DE DESARROLLO

El marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto es el Proceso Unificado de Desarrollo (conocido por sus siglas en inglés como Rational Unified Process, RUP) según expresan los autores Ivar Jacobson, Grady Booch y James Rumbaugh; convirtiéndose no sólo en un proceso de desarrollo de software, sino que constituye un conjunto de metodologías adaptables al contexto y necesidades del proyecto.

Se quiere reflejar cuando se construye un software lo que los futuros usuarios necesitan y desean por medio de casos de uso, que constituyen la guía fundamental para las actividades a realizar durante todo el proceso de desarrollo de software, incluyendo diseño, implementación y prueba. Es por ello,



que el Proceso Unificado se encuentra dirigido por casos de uso, avanzando a través de una serie de flujos de trabajo que parten de un caso de uso. La arquitectura es una vista del diseño completo con las características más importantes y está influenciada por la plataforma en la que tiene que funcionar el software, por lo que centrado en la arquitectura es otra de las características fundamentales de RUP.

La metodología RUP consta de cuatro fases que son: inicio, elaboración, construcción y transición. Dentro de cada fase hay hitos y artefactos a construir, asociados a resultados de cada iteración, en la terminación de una iteración se produce un incremento. Es por ello, que se puede plantear que RUP se caracteriza por ser iterativo e incremental.

2.2 LENGUAJE DE MODELADO

La modelación visual provee una plantilla del sistema. El visualizar esta plantilla, ayuda a entender el sistema que se crea, la estructura o comportamiento del mismo, sirve de guía durante el proceso de construcción del software y permite documentar las decisiones que se toman. Grandes empresas desarrolladoras de software como ORACLE plantea que la modelación visual permite representar los objetos del negocio y entender todas las definiciones de las bases de datos. Por su parte ERISSEON menciona que garantiza la construcción de la arquitectura del software, permite entenderla de manera completa y reduce los riesgos. Por último, HEWLETT PACKARD refiere a que ayuda al equipo a comprender y ver cuál es su trabajo dentro del contexto de construcción del software.

Sobre la base de la observación realizada, se tiene en cuenta, la selección del Lenguaje Unificado de Modelado (Unified Modeling Language, UML), para la especificación, visualización, construcción y documentación de los artefactos que se generan durante el proceso de desarrollo del sistema, es el utilizado por la metodología RUP para preparar todos los esquemas del software. Está demostrado que el trabajo con UML incrementa la productividad, reduce el ciclo de vida de construcción del software e incrementa la calidad del sistema. Es uno de los lenguajes más usados y reconocidos a nivel mundial, ya que provee múltiples beneficios como: modelar sistemas utilizando conceptos orientados a objetos, crear un lenguaje de modelado utilizado tanto por los humanos como por las máquinas, proporcionar un mayor soporte a la planeación y al control de proyectos.

2.2.1 LENGUAJE UNIFICADO DE MODELADO (UML)

En la informática se necesita una estandarización en la manera de representar gráficamente un modelo para compartir entre distintos diseñadores. Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de



análisis de un problema. Surgiendo así, el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) centrado en la representación gráfica de un sistema. (Orallo, 2011)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones, como visualizar expresando de forma gráfica un sistema para que otra persona lo pueda entender, especificar identificando cuáles son las características de un sistema antes de su construcción, construir a partir de los modelos especificados los sistemas diseñados y por último, documentar los propios elementos gráficos del sistema desarrollado. Entre las partes de UML se tiene: las vistas, los diagramas, los elementos del modelo y los mecanismos de extensión.

2.3 HERRAMIENTAS CASE

Existen un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de un sistema informático, llamados herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) diseñadas para mejorar la productividad en el desarrollo y mantenimiento del software, así como la reducción en cuanto al tiempo y coste de desarrollo.

El uso de una herramienta CASE facilita la verificación y mantenimiento de la consistencia de la información del proyecto, la aplicación de las técnicas de una metodología, de técnicas de reutilización y reingeniería, la planificación y gestión del proyecto informático, la disponibilidad de funciones automatizadas tales como: obtención de prototipos, generación de código, generación de pantallas e informes, generación de diseños físicos de bases de datos, verificadores automáticos de consistencia. A partir de los beneficios que brinda, la utilización de estas herramientas en el desarrollo de la plataforma GeneSIG y el objetivo de mantener el mismo estándar de diseño, se utilizan las mismas herramientas CASE que se seleccionaron para desarrollar los restantes módulos de la plataforma.

2.3.1 VISUAL PARADIGM FOR UML

Visual Paradigm para UML es una herramienta profesional de desarrollo visual, compatible con el ciclo completo de una aplicación y combina el modelado con excelentes herramientas para la generación de códigos, ingeniería inversa y la interoperabilidad con otras aplicaciones. Proporciona una interfaz intuitiva, que se integra perfectamente con aplicaciones como Eclipse/webSphere, JBuilder, NetBeans/Sun ONE, JDeveloper y webLogic Workshop, ofreciendo una sincronización sofisticada y en tiempo real de los códigos y modelos. Presenta además las siguientes características: es multiplataforma, disponible para los Sistemas Operativos Linux, Windows y Mac OS, soporta una amplia gama de bases de datos donde se incluyen: Oracle, Microsoft SQL Server, PostgreSQL y



MySQL., permite la transformación de diagramas Entidad Relación en tablas de bases de datos. y la distribución automática de diagramas. Reorganización de figuras y conectores de los diagramas UML.

Incluye los objetos más recientes de UML, además de diagramas de casos de uso, de clases y de componentes, ofrece soporte para Rational Rose, integración con Microsoft Visio y permite generar reportes y documentación en HTML/PDF.

2.4 LENGUAJES DE PROGRAMACIÓN

En la construcción del módulo de generación de las vistas de los mapas debe prevalecer el modelo arquitectónico cliente-servidor, porque este es el utilizado por la plataforma, donde será integrado el módulo.

Este modelo se caracteriza por tener un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador web) para la presentación. El servidor de aplicaciones, cuya tarea es proporcionar los recursos solicitados por el cliente, pero que requiere de otro servidor para hacerlo. El servidor de bases de datos, que proporciona al servidor de aplicaciones los datos que solicita. De esta forma, es necesario tener en cuenta que se deben seleccionar los lenguajes de programación del lado del cliente y del servidor, así como los servidores que requiere el desarrollo del sistema, como se muestra en la Figura 2.



Figura 2 Modelo Cliente-Servidor

Cualquier lenguaje artificial utilizado para definir una secuencia de instrucciones que la computadora podrá finalmente procesar y ejecutar, es un lenguaje de programación. Definir con exactitud qué es o no un lenguaje de programación puede ser arriesgado, pero el uso del mismo permite que otro programa especial (denominado compilador) traduzca de manera automática el código fuente



(desarrollado utilizando el lenguaje de programación) al código máquina (que es el lenguaje que utiliza la computadora para poder trabajar).

Los lenguajes de programación del lado del servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes del lado del cliente son aquellos que pueden ser directamente digeridos por el navegador y no necesitan un pretratamiento. La principal diferencia es que los lenguajes de programación del lado del cliente se ejecutan en un mismo navegador web, en cambio en los del servidor, las aplicaciones se ejecutan en el servidor sin tener en cuenta el navegador web del cliente.

A continuación se hace referencia a los lenguajes de programación del lado cliente a utilizar como JavaScript, HTML, CSS, Smarty y al lenguaje del lado servidor PHP.

2.4.1 HYPERTEXT PRE-PROCESSOR (PHP)

PHP 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 es posible 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 (Hypertext Pre-Processor).

Entre los competidores principales de PHP se pueden citar a Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JPS) y Allaire ColdFusion. En comparación con estos productos, PHP cuenta con muchas ventajas entre las que se encuentran:

Alto rendimiento y bajo costo: Es muy eficiente, mediante el uso de un único servidor, puede servir millones de accesos al día. Es gratuito.

Interfaces para una gran cantidad de sistemas de bases de datos diferentes: Dispone de una conexión propia a todos los sistemas de bases de datos. Se puede conectar directamente a las bases de datos de PostgreSQL, MySQL, Oracle, dbm, filePro, Hyperwave, Informix, InterBase y Sybase, entre otras.

Bibliotecas incorporadas: Agrega una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con pocas líneas de código.



Portabilidad: Está disponible en todos los sistemas operativos gratuitos de tipo Unix, como Linux y FreeBSD, versiones comerciales de Unix, como Solaris e IRIX o en las diferentes versiones de Microsoft Windows. Su código funcionará sin necesidad de aplicar ninguna modificación a los diferentes sistemas que ejecute PHP.

Acceso al código abierto: Dispone de acceso al código fuente, por lo que si se desea modificar o agregar un elemento al programa, se puede hacer con toda libertad.

Las características y ventajas que presenta PHP ante otros de los lenguajes de programación del lado del servidor web, son motivos claros para que se utilice la versión 5.1 de PHP en el desarrollo de la plataforma y específicamente en el módulo gestión de las vistas de los mapas a incorporar. Se puede mencionar que en las primeras versiones, se había conseguido una plataforma potente y estable para la programación de páginas del lado del servidor. Sin embargo, todavía existían puntos negros en el desarrollo PHP que se han tratado de solucionar con la versión 5, referente a la programación Orientada a Objetos (POO) que, a pesar de que estaba soportada a partir de PHP 3, sólo implementaba una parte muy pequeña de las características de este tipo de programación. El principal objetivo de PHP 5 ha sido mejorar los mecanismos de POO para solucionar las carencias de las versiones anteriores.

2.4.2 LENGUAJE DE MARCADO DE HIPERTEXTO (HTML)

El lenguaje de marcado de hipertexto (HTML, Hypertext Markup Language) es el lenguaje de marcas que se utiliza actualmente en la world wide web. Tradicionalmente, ha combinado conceptos estructurales, físicos y lógicos. Los elementos, en forma de etiqueta como `` y ``, se incrustan en los documentos de texto para indicar a los procesadores cómo deben generar las páginas. Permite describir, hasta cierto punto la apariencia de un documento, y que puede incluir un script, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

2.4.3 HOJA DE ESTILO EN CASCADA (CSS)

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) se basan en una estructura de marcas subyacentes, como por ejemplo HTML. Un estilo no tiene mucho significado si no está vinculado a un elemento. El objetivo de una hoja de estilo es crear una presentación para un elemento en particular o para un conjunto de elementos. La vinculación de un elemento a una especificación de estilo es un proceso muy sencillo; simplemente consiste en un selector, en este caso el nombre del elemento, seguido de su información de estilo asociada, denominada reglas, entre llaves. Su utilización en el desarrollo del módulo de gestión de las vistas de los mapas, es para darle todos los estilos necesarios



al módulo en correspondencia con el de la plataforma, además de ser un pilar importante por gran parecido a un ambiente de escritorio que le da a la plataforma, ofreciendo al usuario una interacción amigable y cómoda.

2.4.4 JAVASCRIPT

JavaScript es un lenguaje de programación “ligero” y orientado a objetos. El corazón del lenguaje se embebe en los navegadores para interpretar los códigos (scripts) que se escriben en las páginas. Estos códigos pueden hacer que exista interactividad dinámica entre el usuario y la página, controlar el navegador o crear páginas HTML sin que tenga que ir nuevamente al servidor.

La máxima interactividad entre el usuario y una página, la verificación de los datos introducidos por el usuario antes de enviar el formulario al servidor, la ejecución de pequeñas cantidades de información al igual que en la bases de datos, el manejo de applets y plug-ins dentro de múltiples marcos HTML y el preprocesado de información antes de enviarla al servidor, son funcionalidades que facilitan que JavaScript sea una de las opciones a utilizar en el desarrollo del módulo para la plataforma GeneSIG.

El lenguaje de programación JavaScript provee el uso de la librería ExtJS en su versión 3.1.0 potenciada por la rica colección de componentes para el diseño y para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, HTML y DOM. Se utiliza porque hace mucho más fácil la creación de la interfaz visual con la que interactúa el cliente final del producto, además de hacerlo un ambiente amigable y muy parecido al ambiente de escritorio.

2.5 ENTORNO DE DESARROLLO INTEGRADO (IDE)

Un Entorno de Desarrollo Integrado (Integrated Development Environment, IDE) es un conjunto de herramientas integradas para el desarrollo de software. Las herramientas se ejecutan a través de una interfaz de usuario y constan de un compilador, un editor y un depurador, entre otros.

Actualmente los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado (por ejemplo C, C++, C#, Java, Python y Visual Basic, entre otros). Además es posible que un mismo entorno de desarrollo tenga la posibilidad de utilizar varios lenguajes de programación, como es el caso de Eclipse.

2.5.1 IDE NETBEANS

Es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso. Permite crear ventanas, menús,



barras de herramientas y acciones fácilmente. Se ejecuta sobre sistemas operativos como Linux, OpenSolaris, Windows y Mac OS. Provee soporte para PHP, mejoras en el editor de código, instalación y actualización más simple y características visuales para desarrollo web. (colectivo de autores, 2012)

Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, resaltados de sintaxis y creación de aplicaciones compatibles con teléfonos móviles. Es utilizada la versión 7.0.1 brindando un soporte completo para Java SE 7 y presenta nuevas funciones de programación.

2.6 FRAMEWORK O COMPONENTES

Un framework es un conjunto de clases base que interactúan en forma cooperativa para dar solución a una determinada necesidad y pueden ser reutilizadas para la construcción de un nuevo software. Incluye múltiples clases y componentes, cada una de las cuales provee una abstracción de un determinado concepto. Describe cómo esas abstracciones trabajan en conjunto para resolver las necesidades a las que da solución y sus componentes son altamente reutilizables.

2.6.1 CARTOWEB

CartoWeb es un software de publicación webGIS construida en PHP sobre UMN MapServer que explora AJAX¹. Su característica más diferenciadora respecto a otros proyectos clientes web ligeros sobre MapServer, es que ofrece un framework que ha sido diseñado con una arquitectura bastante modular y escalable, permitiendo separar la lógica de un servidor (CartoServer) encargado del diálogo con el servidor de mapas y la provisión de servicios a un cliente (CartoClient), cuya misión es acceder mediante SOAP² a los servicios proporcionados por servidores CartoWeb y renderizar de la manera apropiada la información hacia el cliente final (HTML, PDF, etc.). Esto permite separar lógica y físicamente los clientes (CartoClient) de los servidores con múltiples configuraciones (N clientes – M servidores) y posibilidades de escalado. Funcionalmente presenta un abanico muy complejo de características propias de un geoportal, con la posibilidad de ir añadiendo o desarrollando nuevos plugins, siendo otra de las fortalezas del sistema. Permite la representación geoespacial de la información asociada a cualquier negocio que lo requiera. Proporciona servicios de acceso a la

¹ JavaScript asíncrono y XML (Asynchronous JavaScript And XML, AJAX) es una técnica de desarrollo web, que permite la comunicación asíncrona entre el cliente y el servidor.

² SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.



información geográfica, para su consulta, análisis y visualización, mediante una interfaz de usuario sencilla y de fácil manejo que pueda ser utilizada por usuarios no especializados en tecnologías SIG. Integra la información socioeconómica existente (recursos, humanos, activos fijos, entidades de servicios, lugares de interés, etc.) con la información geográfica asociada.

2.7 SERVIDOR WEB

Un servidor web puede definirse como una aplicación informática que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que se soliciten a través del protocolo HTTP o el protocolo HTTP(s). Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar el siguiente bucle:

1. Espera peticiones en el puerto TCP indicado.
2. Recibe una petición.
3. Busca el recurso.
4. Envía el recurso utilizando la misma conexión por la que recibió la petición.

Existen diversos servidores web como Microsoft IIS que corre solamente en plataforma Windows, Sun Java System Web Server, Ngnix, Lighttpd que proveen múltiples beneficios, pero la selección de utilizar Apache en su versión 2.2, se basa no sólo en que este es gratuito y de código abierto, sino que corre sobre cualquier plataforma y es el más común y utilizado en todo el mundo.

2.7.1 APACHE

Apache 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, como BeOS. A causa de estar basado en código existente con una serie de parches, fue conocido al principio como un <<Servidor irregular (poco uniforme)>> (patchy en inglés), lo que llevó el nombre oficial de Apache.

El diseño modular de Apache permite elegir qué características se van a incluir en el servidor al seleccionar los módulos que se van a cargar, ya sea al compilar o al ejecutar el servidor. Apache trabaja con Perl, PHP y otros lenguajes script, permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor, es posible configurarlo para cuando se ejecute un determinado script, cuando ocurra un error en concreto, tiene una alta configurabilidad en la creación y



gestión de logs, permitiendo la creación de ficheros logs, posibilitando tener un mayor control sobre lo que sucede en el servidor.

Funcionamiento de Apache

El servidor viene con una serie de Módulos de Multiprocesadores que son responsables de conectar con los puertos de red de la máquina, aceptar las peticiones, y generar los procesos hijo que se encargan de servirlos.

Es utilizada la versión 2.2 de Apache ya que incorpora variedad de novedades y mejoras, como módulos de cacheo mejorados (mod_cache, mod_disk_cache, mod_mem_cache), una nueva estructura para el soporte de autenticación y autorización que reemplaza los módulos de autenticación proporcionados en las versiones anteriores, soporte para balanceo de carga proxy (mod_proxy_balancer), soporte para manejo de archivos grandes (más de 2GB) en plataformas de 32bits.

2.8 SISTEMA GESTOR DE BASES DE DATOS

Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre las bases de datos, el usuario y las aplicaciones. Se compone de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server. Se caracterizan principalmente porque permiten definir una base de datos: especificar tipos, estructuras y restricciones de datos; construir las bases de datos: guardar los datos en algún medio controlado por el mismo SGBD; manipular las bases de datos: realizar consultas, actualizarlas, generar informes.

2.8.1 POSTGRESQL COMO SERVIDOR (SGBD)

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo la licencia BSD y su código fuente disponible libremente. Es un sistema de gestión de bases de datos de código abierto. 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.

Para el desarrollo de la plataforma PostgreSQL se utiliza la versión 8.3, por lo que para el desarrollo del módulo gestión de las vistas de los mapas, se mantiene la misma versión, teniendo en cuenta



además, que la misma, incluye mayor consistencia en el rendimiento que las versiones anteriores, soporte SQL/XML de acuerdo al estándar ANSI, exportación en formato XML, búsqueda en texto con la herramienta TSearch, soporte de autenticación GSSAPI y SSPI, nuevos tipos de datos: UUIDs, ENUMs y arreglos de tipos compuestos.

El sistema de bases de datos objeto-relacional PostgreSQL, proporciona el uso de la extensión PostGIS, como bases datos subyacente para sistemas de información geográfica y aplicaciones cartográficas web, por tal motivo es utilizado durante el proceso de desarrollo de la plataforma GeneSIG.

PostGIS es estable, rápido, compatible con estándares, tiene cientos de funciones espaciales como buffers, uniones, overlays, distancia, etc. Actualmente es la base datos de código abierto más ampliamente utilizada. La administración de las bases de datos está disponible a través de pgAdmin y phpPgAdmin, entre otros. Además, hay diversos clientes SIG de escritorio y web para visualizar datos PostGIS. Provee integridad transaccional ACID, índice espacial R-Tree, soporte multiusuario, bloqueo a nivel de fila, replicación, particionado, seguridad basada en roles, table-spaces, esquemas.

2.8.2 PGADMIN COMO CLIENTE PARA POSTGRESQL

pgAdmin es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados (EnterpriseDB Postgres Plus Advanced Server y Greenplum Database). Incluye: interfaz administrativa gráfica, herramienta de consulta SQL, editor de código procedural y agente de planificación SQL/shell/batch. PgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Se puede ejecutar en sistemas operativos como Windows, Linux, FreeBSD, Mac OSX y Solaris. Se encuentra bajo la licencia de PostgreSQL.

2.9 SERVIDOR DE MAPAS

Un servidor de mapas, más conocido por sus siglas en inglés IMS (Internet Map Server) provee mapas o cartografía a través de Internet, el uso de la web como medio de difundir mapas es un avance en la cartografía, facilitando el acceso de publicación de Información Geográfica actualizada, en tiempo real, y de forma más barata a cualquier parte del mundo.

El proceso de diseñar, implementar, generar y difundir mapas en la world wide web es conocido como Web Mapping, aprovechando la arquitectura cliente-servidor, el cliente que puede ser un navegador con un visualizador de mapas con alguna tecnología del lado del cliente (javascript) o una aplicación



cliente puede realizar una petición IMS para obtener información para visualización, consulta o análisis, a través de Internet o la Intranet corporativa, y el servidor de mapas interpretará la petición, recuperará la información de la geodatabase o archivo y devolverá una imagen o un objeto geográfico de forma interactiva y dinámica.

La arquitectura consta principalmente de los siguientes componentes: Un computador con un cliente para mapas, una conexión de red a la Intranet o Internet, un servidor web, Web Application Server, Servidor de Web de Mapas, Datos y Metadatos Geoespaciales.

2.9.1 MAPSERVER

MapServer 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 CGI 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 CGI.

MapServer debe ser compilado con una serie de librerías de terceros dependiendo de la funcionalidad que se espera. Las capacidades de MapServer pueden ser extendidas a través de MapScript y customizada mediante templating. MapScript provee una interfaz de programación para la construcción de aplicaciones web y de escritorio, puede ser utilizado independientemente de CGI MapServer, y viene como un módulo con soporte para PHP, Perl, Python, Ruby, Tcl, Java y .Net.

CONCLUSIONES PARCIALES

El estudio realizado sobre el entorno de las tecnologías, abarcando sus principales propiedades, funcionalidades y usos, ha permitido identificar cuáles de ellas responden a las necesidades del módulo a desarrollar y se ajustan a las características y condiciones del ambiente de desarrollo, aprovechando los múltiples beneficios que proveen, para alcanzar las expectativas del sistema en construcción. Además, se determina que para lograr una documentación óptima, producto a los constantes cambios del equipo de trabajo, llevar un mayor control y planificación de las actividades, ganar en rapidez por el dominio que se tiene, siendo aplicada en un abanico de proyectos de la universidad y a nivel mundial con buenos resultados, seleccionar como metodología de desarrollo al Proceso Unificado de Desarrollo (RUP). Para modelar los artefactos que se van generando durante el proceso de desarrollo del módulo gestión de las vistas de los mapas el lenguaje de modelado UML en su versión 1.0 y representar los mismos la herramienta CASE Visual Paradigm en su versión 6.4.



CAPÍTULO III. PRESENTACIÓN DEL MÓDULO VISTAS DE LOS MAPAS

INTRODUCCIÓN

En el presente capítulo, se realiza un estudio del ambiente donde será implantado el sistema y los factores que permiten representar los conceptos u objetos del mundo real identificados en el mismo. Se obtienen y describen los requisitos tanto funcionales como no funcionales que debe cumplir el sistema a implementar, sobre la base del análisis de las soluciones existentes especificadas en el capítulo anterior y las entrevistas realizadas a los especialistas del proyecto, para elaborar la estructura del Modelo de Casos de Uso del Sistema, identificando los actores, los casos de uso y las relaciones que existen entre estos, para luego realizar una descripción detallada de los mismos.

3.1 MODELO DE DOMINIO

Un Modelo de Dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema y los conceptos del mundo real, no de los componentes del software.

Una clase conceptual puede ser una idea o un objeto físico (símbolo, definición y extensión). Es un diagrama de clases en el que se muestran:

- . Conceptos u objetos del dominio del problema: clases conceptuales.
- . Asociaciones entre las clases conceptuales.
- . Atributos de las clases conceptuales.

Un Modelo del Dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, además de que no incluyen las responsabilidades de las personas que ejecutan las actividades. Según RUP este tipo de modelo representa un subconjunto del Modelo de Objeto del Negocio, lo cual no significa que siempre que haya un Modelo de Dominio tenga que existir obligatoriamente un Modelo de Negocio.

3.1.1 REPRESENTACIÓN DEL MODELO DE DOMINIO

Luego de haber realizado un estudio del entorno donde estará el sistema, se determina realizar un modelo de dominio sobre la base de que no se tienen bien definidos los procesos de negocio, ni los responsables de estos procesos, por la existencia de solapamiento de responsabilidades y la dificultad en el establecimiento de las reglas del funcionamiento del producto a implementar. De esta forma, queda representado el modelo de dominio.



Modelo de dominio

Se muestra el diagrama de clases del modelo de dominio correspondiente al entorno donde estará el sistema, teniendo en cuenta los conceptos u objetos más importantes y las relaciones que existen entre estos. (Ver Figura 3 Modelo de Dominio)

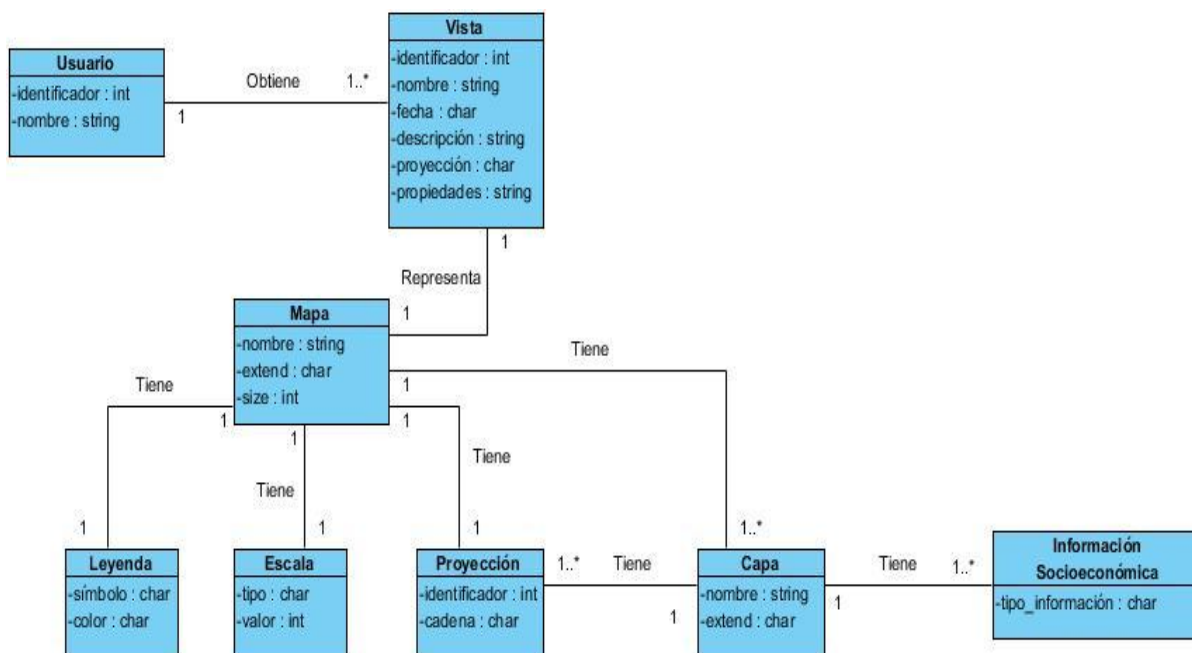


Figura 3 Modelo de Dominio

Definición de las clases del modelo de dominio

Usuario

Persona que accede a la plataforma GeneSIG, que necesita crear, modificar y eliminar una vista del mapa.

Mapa

Es una representación gráfica y métrica de una porción de territorio sobre la superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los globos terráqueos. El que el mapa tenga propiedades métricas significa que ha de ser posible tomar medidas de distancia, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto.



Leyenda

Explicación de los símbolos, los colores, las tramas y los sombreados empleados en un mapa, suele encontrarse a pie de página o en un recuadro, situado en sus márgenes o bien en su dorso. Los símbolos empleados en los mapas pueden llegar a contener un gran volumen de información, que por su facilidad de lectura permiten una rápida interpretación.

Escala

Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre. En los mapas, la escala puede expresarse de tres modos distintos: en forma de proporción o fracción, con una escala gráfica o una expresión en palabras y cifras. Cuanto mayor es la escala, más se aproxima al tamaño real de los elementos de la superficie terrestre. Los mapas a pequeña escala generalmente representan grandes porciones de la Tierra y, por tanto, son menos detallados que los mapas realizados con escala más grandes.

La relación matemática entre las dimensiones en el mapa, carta o plano y la superficie terrestre que representa. Por extensión puede referirse a la mayor o menor profundidad del enfoque en un tema geográfico.

Proyección

Es un sistema de representación gráfico que establece una relación ordenada entre los puntos de la superficie curva de la Tierra y los de una superficie plana (mapa). Estos puntos se localizan auxiliándose en una red de meridianos y paralelos, en forma de malla. La única forma de evitar las distorsiones de esta proyección sería usando un mapa esférico pero, en la mayoría de los casos, sería demasiado grande para que resulte útil.

Capa

Conjunto lógico de elementos temáticos descritos y almacenados en una biblioteca. Estas capas organizan la biblioteca según temas.

Información socioeconómica

Es un conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier lugar de interés del país.



Descripción del diagrama de clases del dominio

El usuario realiza la búsqueda de determinada ubicación geográfica, de esta forma, obtiene una vista, que está representada por un mapa.

Los mapas están compuestos por varios atributos como leyenda, escala representativa, proyección, capas que permiten un mejor entendimiento y análisis de los mismos. Las capas deben tener una proyección y por su diversidad proporcionan toda la información socioeconómica de un lugar, que se muestra al incorporarla al mapa.

3.2 REQUISITOS

Lo más difícil en la construcción de un sistema software es decidir precisamente qué construir, plantea Frederick Phillips Books. Indiscutiblemente los ingenieros en el desarrollo de software, como él, se enfrentan frecuentemente a situaciones en las que los usuarios no saben lo que quieren, no saben cómo hacer más eficiente la operación en su conjunto, no saben qué partes de su trabajo pueden transformarse en software, no saben detallar lo que conocen de forma precisa.

La metodología RUP, mediante el flujo de trabajo de requisitos, proporciona una respuesta a situaciones como las que se plantea con anterioridad, de forma tal, que trata principios, métodos, técnicas y herramientas que permiten descubrir, documentar y mantener los requisitos de forma sistemática y repetible.

Sobre la base del análisis de las soluciones existentes en el capítulo anterior, la representación de los conceptos u objetos del mundo real mediante el modelo de dominio, las entrevistas realizadas a los especialistas del proyecto y la necesidad de realizar una correcta captura de los requisitos, para evitar demoras en la construcción del módulo y errores en los mismos, fueron identificados los requisitos que debe cumplir el sistema a construir, que se presentan en los subepígrafos que se relacionan a continuación.

3.2.1 ESPECIFICACIÓN DE LOS REQUISITOS FUNCIONALES

Los requisitos funcionales son las capacidades o condiciones que el sistema debe cumplir, para que el usuario resuelva su problema o cumpla sus objetivos, es decir, especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. Deben estar claros y libres de ambigüedades, asegurando que los involucrados comprendan claramente el significado de cada uno.



El sistema a implementar debe llevar el control y registro de las vistas de los mapas por cada usuario, permitiendo reducir el tiempo de búsqueda de una ubicación en el mapa.

Requisitos del sistema para que el usuario pueda interactuar con las vistas de los mapas.

RF1. <Adicionar vista del mapa> Permitirá al usuario crear una vista del mapa, introduciendo para ello el nombre de la vista y la descripción de la misma.

RF2. <Mostrar vista del mapa > Permitirá visualizar en un listado todas las vistas de los mapas creadas por el usuario, mostrando de cada vista, el nombre, la fecha de creación, una descripción de la misma y las propiedades de los contornos de visualización.

RF3. <Modificar vista del mapa> Permitirá modificar los datos entrados en la creación de la vista.

RF4. <Eliminar vista del mapa> Permitirá que el usuario pueda eliminar una vista del mapa y los atributos asociadas a esta.

RF5. <Actualizar vista del mapa> Permitirá que el usuario pueda actualizar las vistas de los mapas.

RF6. <Localizar vista del mapa> Permitirá que el usuario pueda localizar una vista del mapa, mostrándose la vista en la interfaz principal de la aplicación.

RF7. <Buscar vista del mapa> Permitirá buscar una vista del mapa que haya sido creada con anterioridad. De ello, debe admitir que se pueda introducir al menos un criterio de búsqueda, en este caso, los criterios son la fecha, la descripción y el nombre de la vista.

Requisitos del sistema que relacionen al usuario con la vista creada.

RF8. <Asignar una vista del mapa a un usuario> Permitirá asignar a un usuario autenticado una vista del mapa.

RF9. <Compartir una vista del mapa> Permitirá compartir una vista del mapa al usuario que se encuentra autenticado, con otro usuario del sistema.

Requisitos del sistema para que un usuario, que tenga un rol de administrador, pueda acceder a todas las vistas de los mapas de cada usuario.

RF10. <Mostrar vistas de mapas de usuarios> Permitirá mostrar todas las vistas de los mapas creadas por cada uno de los usuarios que se encuentren registrados en el sistema.



3.2.2 ESPECIFICACIÓN DE LOS REQUISITOS NO FUNCIONALES

Los requisitos no funcionales son las propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. Normalmente están vinculados a requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Los requerimientos no funcionales también añaden funcionalidad al producto, pues hacen que un producto sea fácil de usar, seguro, o demanda cierta cantidad de procesamiento.

RNF1. Apariencia o interfaz externa

Se desea que la interfaz externa del producto sea de fácil navegación por el usuario, sencilla y legible, que mantenga los estándares y características de la plataforma soberana GeneSIG como son: los colores, la estructura de los botones, el estilo y tamaño de letra, ya que el módulo a construir va a ser incorporado a la plataforma.

RNF2. Usabilidad

En este requisito se propone que los usuarios que accedan a la plataforma GeneSIG, puedan utilizar el módulo para la generación de las vistas de los mapas, aunque no tengan vastos conocimientos sobre la rama de la informática.

RNF3. Portabilidad

La portabilidad del módulo la adquiere con la característica de la plataforma GeneSIG, de tener la posibilidad de operar en varios sistemas operativos, sin necesidad de modificar el código fuente que se encuentra en el servidor. Además el uso del lenguaje de programación PHP, le resulta provechoso a la hora de realizar una migración de sistema operativo.

RNF4. Software

Se requiere para las estaciones de trabajo servidoras las siguientes condiciones:

- Servidor web Apache en su versión 2.0 o superior.
- PostgreSQL en su versión 8.3, como Sistema Gestor de Bases de Datos.
- PostGIS como extensión de PostgreSQL, como soporte de datos espaciales.



Se requiere para las estaciones de trabajo cliente las siguientes condiciones:

- Un navegador web que pueda ser Netscape 3 (o superior), Internet Explore 4.2 (o superior), Mozilla Firefox 2.3 (o superior).
- Sistemas Operativos GNU/Linux, Windows 95 o superior.

RNF5. Hardware

En el caso de las PC clientes donde se ejecutará la aplicación, se requiere de las siguientes condiciones:

Procesador Pentium IV 2x2 cache, con velocidad de procesamiento de 3 GHz como mínimo y 2 GB de RAM.

En dependencia de la funcionalidad concebida para cada uno de los servidores, son las condiciones que se requiere que deban cumplir:

Servidor web

Requiere un procesador Pentium IV 2X2 cache, velocidad del procesador de 3GHz como mínimo y memoria de 2GB de RAM.

Servidor de datos

Requiere un procesador Pentium IV 2x2 cache, velocidad del procesador de 3 GHz como mínimo, memoria de 2 GB de RAM, almacenamiento en disco con una capacidad igual o superior a los 10GB (como mínimo) y tecnología de respaldo de datos históricos.

Servidor de mapas

Requiere un procesador Pentium IV ó superior, con velocidad de procesamiento de 3 GHz como mínimo, memoria de 2 GB de RAM y disco duro de 5 GM libre como mínimo.

RNF. 6 Confiabilidad

Se garantiza la confiabilidad a partir de que los servidores de bases de datos poseen respaldos de la información y formas de recuperación en caso de pérdida de la misma.

RNF. 7 Seguridad



Por la importancia que tienen los datos geográficos y socioeconómicos, tienen un gran valor para las empresas o entidades que lo utilizan, la seguridad del software es primordial, por lo que la plataforma consta de un módulo que se encarga de la seguridad de estos datos, además de contar con la seguridad que brindan la totalidad de las herramientas que se utilizaron para su desarrollo.

3.3 DESCRIPCIÓN DEL MODELO DE CASOS DE USO DEL SISTEMA

El Modelo de Casos de Uso del Sistema es un modelo que contiene actores, casos de uso y sus relaciones; describe lo que el sistema debería hacer por sus usuarios y bajo qué restricciones. Permite que los desarrolladores y clientes lleguen a un acuerdo sobre los requerimientos, y proporciona la entrada fundamental para el análisis, diseño y las pruebas.

3.3.1 DESCRIPCIÓN DE LOS ACTORES QUE INTERACTÚAN CON EL SISTEMA

Un actor es un usuario del sistema. Esto incluye usuarios humanos y otros sistemas computacionales. El conjunto de casos de uso al que un actor tiene acceso define un rol en el sistema y el alcance de su acción. A continuación, se mencionan los actores que van a interactuar con el sistema a construir, definiendo el rol que le ocupa dentro del mismo.

Actor	Descripción
Usuario	Representa los usuarios que van a interactuar con el sistema, ya sea administrador o usuario registrado.
Administrador	Encargado de visualizar todas las vistas de los mapas creadas por los restantes usuarios.

Tabla 1 Descripción de los Actores del Sistema

Modelado del Sistema

Después de haber identificado, analizado y descrito cada uno de los requisitos, se procede al modelado del sistema. Utilizando UML se construyeron los diagramas de casos de uso, según lo requirió el sistema.

3.3.2 IDENTIFICACIÓN DE LOS CASOS DE USO

Los casos de uso son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para los actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema debe llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de una



secuencia. A continuación se identifican los casos de uso que hacen referencia a los requisitos funcionales que requiere el sistema a construir.

Referencia a requisitos	Nombre del caso de uso	Prioridad
RF: 1, 2 ,3 ,4 y 5	Gestionar vista del mapa.	Secundario
RF: 6	Localizar vista del mapa.	Secundario
RF: 7	Buscar vista del mapa.	Secundario
RF: 8 y 9	Compartir vista del mapa.	Secundario
RF: 10	Mostrar vistas de mapas de usuarios.	Secundario

Tabla 2 Casos de Uso

3.3.3 DIAGRAMAS DE CASOS DE USO DEL SISTEMA (DCUS)

El diagrama describe el comportamiento del sistema “Vistas de los Mapas”, muestra los casos de uso (funcionalidades) que pueden realizar los actores (usuario Y administrador). (Ver Figura 4 Diagrama de Casos de Uso)

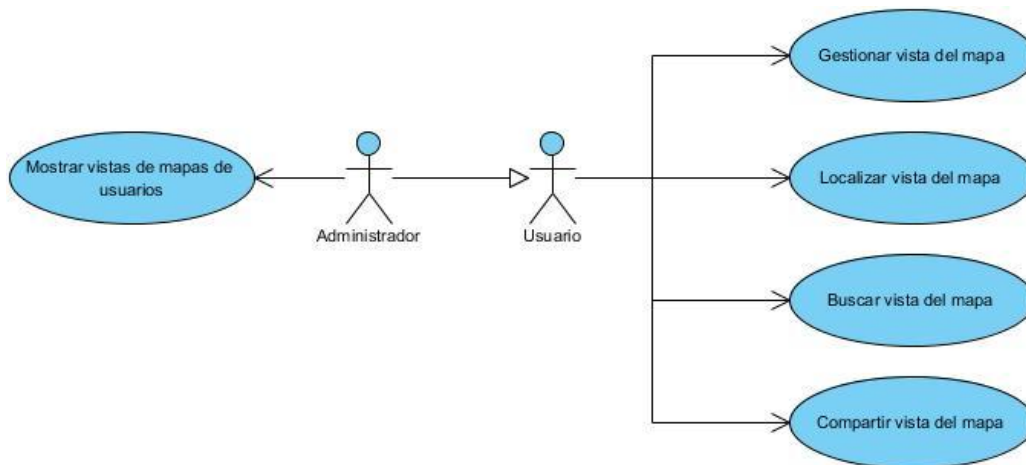


Figura 4 Diagrama de Casos de Uso

3.3.4 DESCRIPCIÓN EXTENDIDA DE LOS CASOS DE USO DEL SISTEMA

Debido a la existencia de varios casos de uso y que las descripciones textuales son muy extensas, se propone la del caso de uso Gestionar Vista del Mapa en el siguiente apartado 3.3.4.1 y la de los restantes en el Anexo #2.

3.3.4.1 Descripción extendida del Caso de Uso Gestionar Vista del Mapa.



Caso de Uso:	Gestionar Vista del Mapa
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de gestionar una vista del mapa.
Resumen:	El caso de uso se inicia cuando el usuario desea adicionar, modificar, mostrar y eliminar una, o (las) vistas de los mapas y termina cuando han sido completadas todas las acciones mencionadas.
Precondiciones:	El usuario tiene que estar autenticado.
Referencias	RF 1, RF 2, RF 3, RF4 y RF5
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El caso de uso se inicia cuando el usuario selecciona la opción Vista en la barra de herramienta de la plataforma. (Ver Interfaz 1.1)</p>	<p>2. El sistema crea un Tab Vista en el Panel Izquierdo de la aplicación. (Ver Interfaz 1.2)</p> <ul style="list-style-type: none"> - Un botón para cerrar el Tab Vista. - Cuatro botones, Adicionar, Eliminar, Modificar y Compartir. - Grid que contiene las columnas: nombre, fecha, descripción y propiedades. - En la parte inferior un paginado para mostrar la cantidad de páginas, un refrescador de la ventana. <p>En caso de existir alguna vista creada, la visualiza con los atributos de la misma.</p> <ul style="list-style-type: none"> - Si el usuario desea adicionar una vista, ir a la sección “Adicionar Vista del Mapa”.

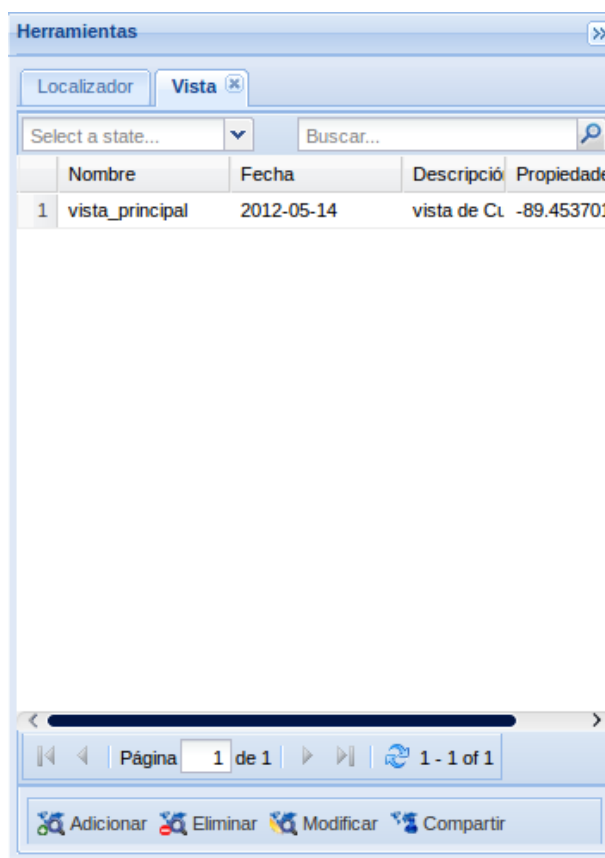


- Si el usuario desea eliminar una vista, ir a la sección “Eliminar Vista del Mapa”.
- Si el usuario desea modificar los datos de una vista, ir a la sección “Modificar Vista del Mapa”.

Prototipo de Interfaz



Interfaz 1 1



Interfaz 1 2

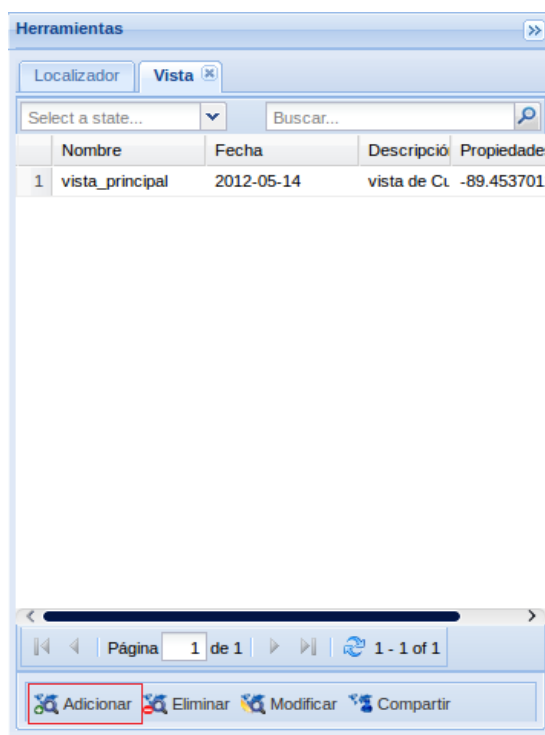
Sección “Adicionar Vista de Mapa”

Acción del Actor

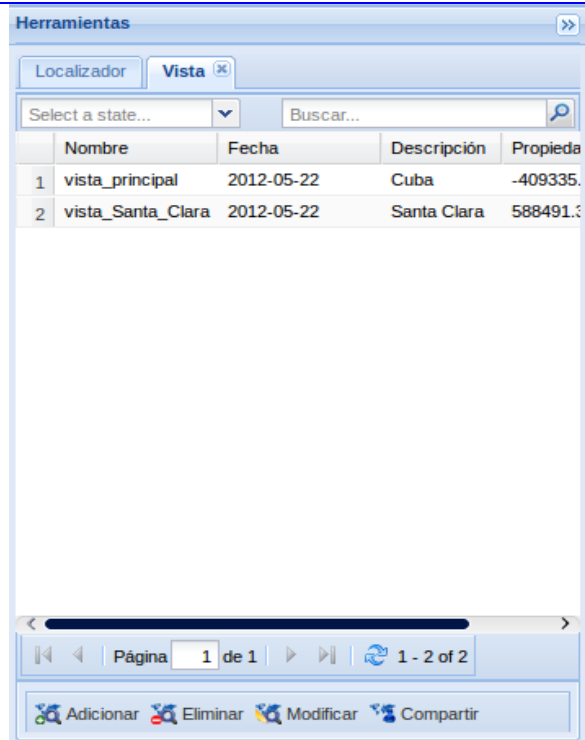
Respuesta del Sistema



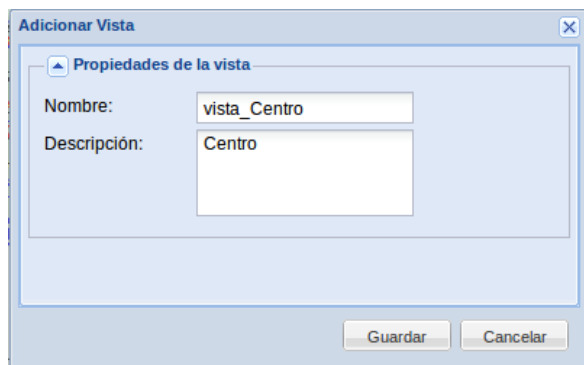
<p>1. El usuario da clic en el botón Adicionar. (Ver Interfaz. 1.3).</p>	<p>2. El sistema muestra una ventana para adicionar: (Ver interfaz 1.4).</p> <ul style="list-style-type: none"> - FieldSet Propiedades de la vista que identifica que son los atributos de la vista del mapa. - Un TextBox Nombre para introducir el nombre de la vista del mapa. - Un texttarea Descripción para introducir la descripción de la vista del mapa. - Dos botones, Guardar y Cancelar.
<p>3. El usuario introduce el nombre de la vista del mapa, la descripción y da clic en el botón Guardar.</p>	<p>4. El caso de uso termina cuando el sistema adiciona la nueva vista del mapa, muestra una ventana de confirmación con el mensaje “Se ha adicionado la vista satisfactoriamente” (Ver interfaz 1.5) y actualiza los campos nombre, fecha, descripción, propiedades y el paginado del Grid. (Ver interfaz 1.6).</p>
<p>Flujo Alterno</p>	
<p>3.1 El usuario no introduce el nombre de la vista del mapa y da clic en el botón Guardar.</p>	<p>3.2 El sistema delimita con color rojo el campo que el usuario debe llenar (Ver interfaz 1.7) y muestra una venta de error indicándole al usuario que debe llenar el campo nombre y vuelve al paso 3 del flujo normal de los eventos. (Ver interfaz 1.8).</p>
<p>Prototipo de Interfaz</p>	



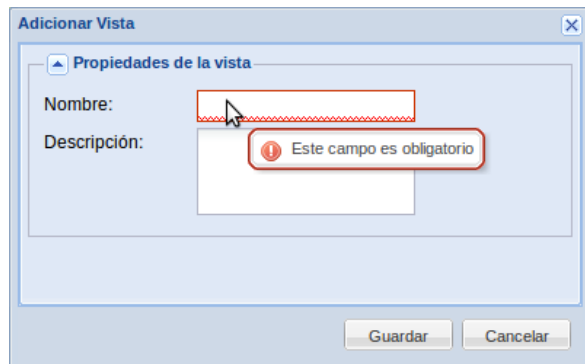
Interfaz 13



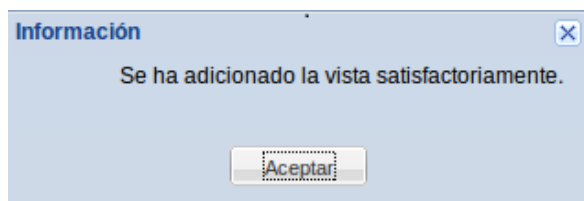
Interfaz 16



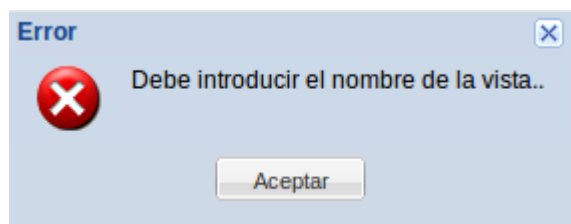
Interfaz 14



Interfaz 17



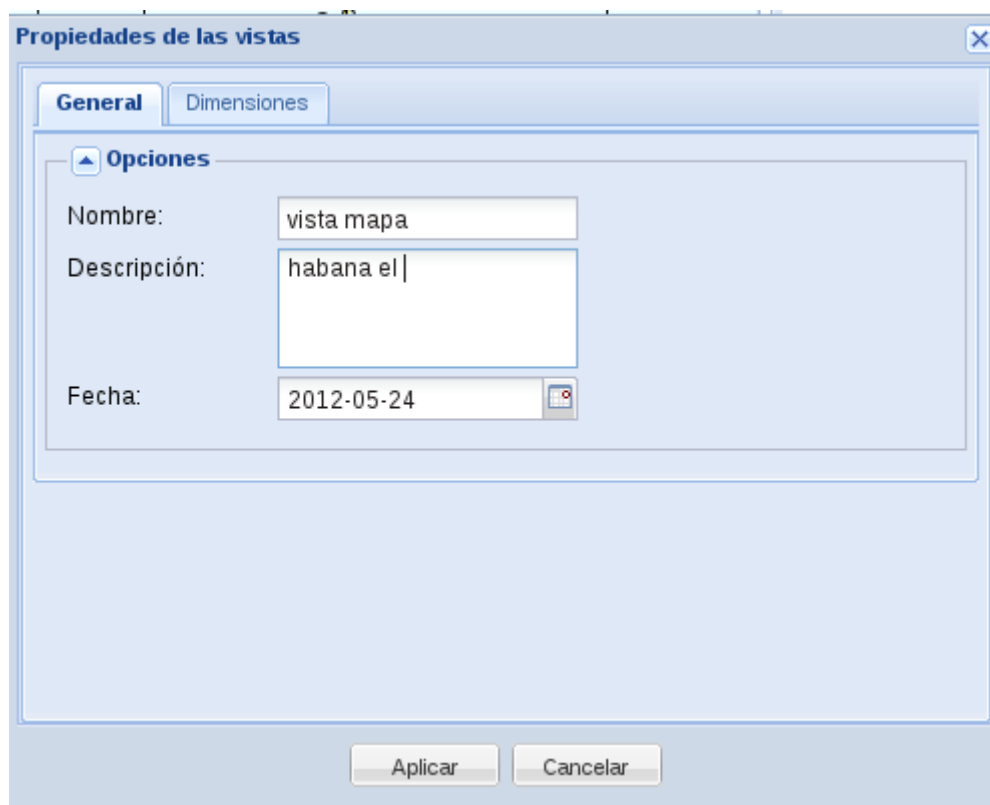
Interfaz 15



Interfaz 18



Sección “Modificar Vista de Mapa”	
Acción del Actor	Respuesta del Sistema
<p>1. El usuario selecciona la vista del mapa que desea modificar y da clic en el botón Modificar.</p>	<p>2. El sistema muestra la ventana Modificar Vista del Mapa para que el usuario introduzca los datos y realice las modificaciones. (Ver Interfaz 1. 9)</p> <ul style="list-style-type: none"> - Con un TextBox para modificar el nombre de la vista del mapa. - Con un texttarea para modificar la descripción de la vista del mapa. - Con un TextBox para modificar la escala de visualización de la vista. - Con dos botones, Guardar y Cancelar
<p>3. El usuario modifica los valores que considera y da clic en el botón Aceptar.</p>	<p>4. El caso de uso termina cuando el sistema actualiza los nuevos datos modificados por el usuario y los campos nombre, descripción y fecha de creación y fecha de modificación del Grid.</p>
Flujos Alternos	
<p>1.1 El usuario no selecciona la vista del mapa que desea modificar y da clic en el botón Modificar.</p>	<p>1.2 El sistema le muestra una ventana de error al usuario donde le indica que debe elegir la vista del mapa que desea modificar, y vuelve al paso 1.</p>
Prototipo de Interfaz	



Interfaz 19

Sección “Eliminar Vista de Mapa”

Acción del Actor	Respuesta del Sistema
<p>1. El usuario selecciona la vista del mapa que desea eliminar y da clic en el botón Eliminar. (Ver interfaz 1.10).</p>	<p>2. El sistema muestra una ventana al usuario donde le indica que debe confirmar si está seguro que desea eliminar la vista del mapa seleccionada. (Ver interfaz 1.11)</p>
<p>3. El usuario da clic en el botón Aceptar.</p>	<p>4. El caso de uso termina cuando el sistema elimina la vista del mapa seleccionada por el usuario y se actualiza el Grid. (Ver interfaz 1.12)</p>

Flujos Alternos

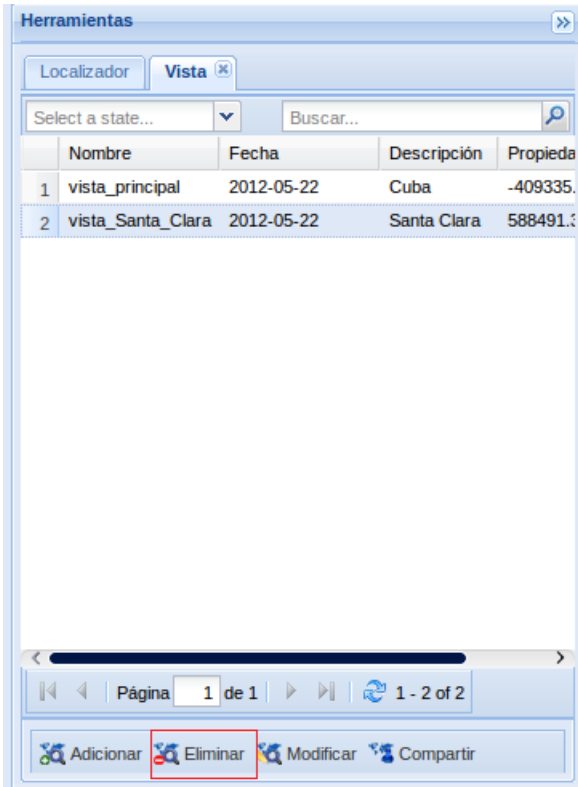
<p>1.1 El usuario no selecciona la vista del mapa que desea eliminar y da clic en el botón</p>	<p>1.2 El sistema le muestra una ventana de error al usuario donde le indica que debe elegir la vista del</p>
--	---



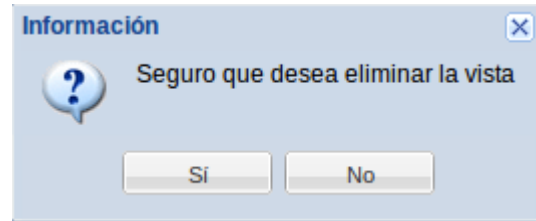
Eliminar.

mapa que desea eliminar, y vuelve al paso 1.

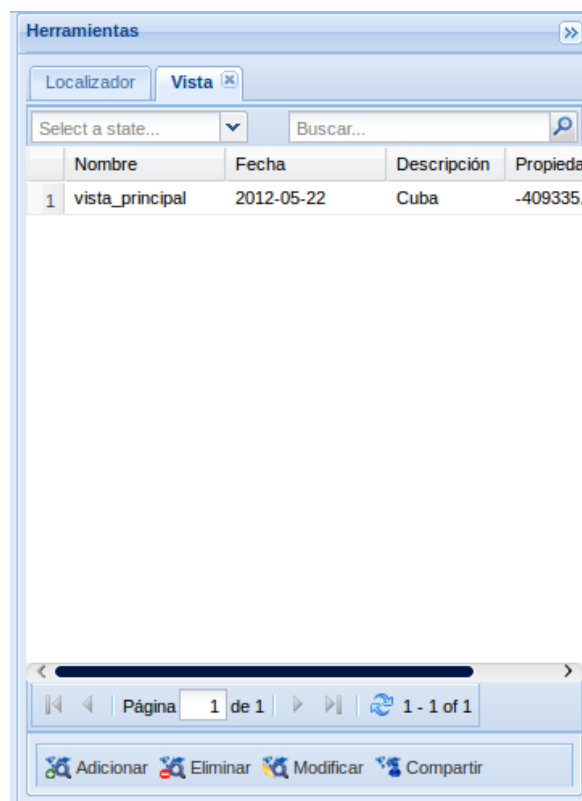
Prototipo de Interfaz



Interfaz 1 10



Interfaz 1 9



Interfaz 1 2

Tabla 3 Descripción del caso de uso "Gestionar Vista del Mapa"



CONCLUSIONES

La presentación de la solución tecnológica para el módulo de generación de las vistas de los mapas, realizada en el presente capítulo, luego de haber generado los artefactos de los dos primeros flujos de trabajo que propone la metodología RUP, ha permitido que se puedan arribar a las siguientes conclusiones:

1. La captura e identificación correcta de los requerimientos funcionales del sistema permite que se obtenga un producto, lo más consecuente con las solicitudes del cliente.
2. La identificación adecuada de los requisitos no funcionales permite que no ocurran retrasos en la construcción del sistema por fallos en algunas de las aplicaciones o servidores, se mantenga la seguridad de la información y en caso de pérdida de la misma, se pueda restablecer y recuperar.
3. La identificación de los casos de uso a partir de los requerimientos funcionales y la descripción de los mismos, permite que se tenga una mayor visión del módulo que se quiere construir, el comportamiento del mismo y la secuencia de actividades que debe de realizar el usuario para lograr su objetivo.



CAPÍTULO IV. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

INTRODUCCIÓN

Cuando se inicia la construcción de un sistema de software, luego de conocer cuáles son las necesidades básicas del usuario final, el ambiente donde será desarrollado el mismo, las condiciones para ello, las tecnologías y recursos necesarios para lograr un ambiente de trabajo adecuado, entonces se sientan las bases del esqueleto arquitectónico del sistema.

En el presente capítulo se realiza una propuesta de la vista arquitectónica para el plugin Generación de las vistas de los mapas, teniendo en cuenta los patrones arquitectónicos y de diseño GRASP que permitirán describir la estructura y comportamiento del plugin. Luego de haber definido el esqueleto arquitectónico, son creadas las condiciones para pasar directamente al modelo de diseño del sistema, justificando de manera concreta que el uso de la metodología RUP permite que se pueda prescindir del modelo de análisis. Se realiza el diagrama de paquetes y se provee una propuesta del diagrama de clases del diseño que permite representar las clases y las relaciones entre estas. Además, se define el modelo de datos, a partir del modelo entidad relación, así como el modelo de despliegue e implementación.

4.1 ARQUITECTURA DE SOFTWARE

Según el estándar de la IEEE (Institute of Electrical and Electronics Engineers): “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”.

La arquitectura del software de un programa o sistema de cómputo es la estructura o las estructuras del sistema, que incluyen los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos. Es una representación que permite analizar la efectividad del diseño para cumplir con los requisitos establecidos, considerar opciones arquitectónicas en una etapa en que aún resulta relativamente fácil hacer cambios al diseño y reducir los riesgos asociados a la construcción del software.

4.1.1 PATRONES ARQUITECTÓNICOS

Un patrón arquitectónico impone una transformación en el diseño de la arquitectura, su alcance es más específico ya que se concentra en un aspecto, en lugar de hacerlo en toda la arquitectura. Un patrón aplica una regla sobre la arquitectura, describe la manera en que el software maneja algún aspecto de su funcionalidad al nivel de la infraestructura, abarca aspectos específicos del comportamiento dentro



del contexto de la arquitectura y es usado para determinar la forma de la estructura general de un sistema.

El módulo vista de mapas, se ha desarrollado bajo el patrón arquitectónico basado en componentes. Este define cómo organizar el modelo en componentes funcionales, exponiendo interfaces de comunicación que contienen métodos, eventos y propiedades. El mismo permite que se pueda representar el módulo como un componente que es incorporado a la plataforma GeneSIG, de manera que no altere los restantes componentes, ni la estructura de la misma; brindando la posibilidad de que el sistema sea flexible y fácil de personalizar.

4.1.2 PATRONES DEL DISEÑO

Uno de los autores del libro “A Pattern Language”, Christopher Alexander, comenta que “Cada patrón de diseño describe un problema que ocurre una y otra vez en un entorno, para describir el núcleo de la solución a ese problema, de tal manera que la solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma”. El patrón es un esquema de solución que se aplica a un tipo de situación, esta aplicación del mismo no es mecánica, sino que requiere de adaptación y matrices.

En el diseño de la propuesta de solución se aplican los patrones GRASP, el cual es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). A continuación se mencionan los patrones GRASP utilizados, que permiten describir los principios fundamentales de diseño de objetos para la asignación de responsabilidades.

Experto: Se aplica para asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. En este caso, se tiene la clase ClientView, que es experta en procesar los datos que son enviados a través del navegador web, la clase ServerView, que es la responsable de la interacción con el servidor de mapas y el servidor de bases de datos y la clase view.ajax, experta en el comportamiento por la parte cliente del plugin. El uso de este patrón permite que se conserve el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento.

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. En este caso, el patrón se refleja en las clases ClientView



y ServerView, encargadas de crear una instancia de las clases ViewRequest y ServerRequest, clases que describen las variables que contienen la información que forman parte de los valores de entrada de la solicitud que realiza el ClientView al ServerView y viceversa.

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 plugin vistas de los mapas, con el objetivo de que una clase no dependa de muchas clases, de esta forma, no se afectan las clases por cambios de otros componentes, son fáciles de entender por separado y fáciles de reutilizar.

Alta Cohesión: La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. En este caso, se garantiza que cada una de las clases del plugin vista de los mapas, posean alta cohesión, de manera que las clases posean la característica de tener las responsabilidades estrechamente relacionadas y que no realicen un trabajo enorme. El uso de este patrón permite que se pueda mejorar la claridad y facilidad en que se entiende el diseño, se simplifique el mantenimiento y existan mejoras de funcionalidad.

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 view.ajax, encargada del comportamiento del plugin y gestiona todos los eventos que ocurren en el mismo.

4.2 DIAGRAMA DE PAQUETES

Un diagrama de paquetes muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Los paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Con estas líneas maestras sobre la mesa, los paquetes son buenos elementos de gestión. Se propone de esta forma, el siguiente diagrama de paquetes, para el plugin vistas de los mapas. (Ver Figura 5 Diagrama de Paquetes).

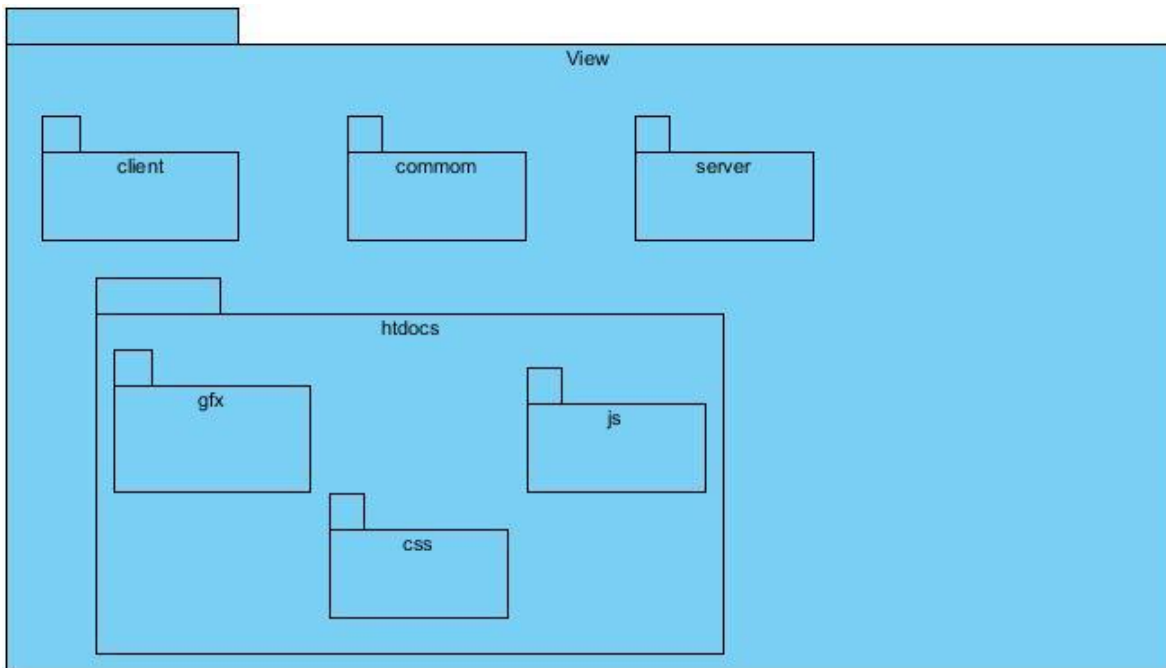


Figura 5 Diagrama de Paquetes

Breve descripción de los paquetes del plugin vista de mapas.

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

server/ contiene todos los componente(de extensión 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 client.php y server.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.

4.3 MODELO DE DISEÑO

Durante el desarrollo del sistema se ha determinado no realizar el flujo de trabajo de análisis que propone RUP y pasar directamente al flujo de trabajo de diseño, sobre la base de la observación



realizada por los autores del libro: El Proceso Unificado de Desarrollo, quienes plantean no realizar el análisis cuando se analizan los requisitos como parte integrada en la captura de requisitos, teniendo mayor formalismo en el modelo de casos de uso, a partir de que el cliente sea capaz de comprender los resultados. Además de que son analizados los requisitos como parte integrada del diseño, a partir de que los requisitos son simples y/o son bien conocidos, siendo fácil identificar la forma del sistema (incluyendo su arquitectura), o si los desarrolladores cuentan con cierta comprensión, intuitiva pero correcta, de los requisitos, y son capaces de construir un sistema que los encarne de manera directa (Ivar Jacobson, 2004).

Teniendo en cuenta la observación realizada con anterioridad, el conocimiento y dominio que se tiene sobre los requisitos que debe de cumplir el sistema, la utilización de la metodología RUP, que por ser altamente configurable, garantiza que existan un conjunto de artefactos que no sean de obligatoria construcción, como es el caso, del diagrama de clases del análisis, la correcta identificación de la tecnología a usar, el lenguaje de programación, los componentes a reutilizar, se está en condiciones de elaborar directamente el modelo de diseño, permitiendo reducir el coste en recursos y tiempo que se tendría que emplear en el modelo de análisis.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Los subsistemas de diseño y las clases de diseño representan abstracciones del subsistema y componentes de la implementación del sistema.

Las clases del diseño representan una abstracción de una o varias clases (o construcción similar) en la implementación del sistema. El lenguaje utilizado para especificar una clase del diseño es el mismo que el lenguaje de programación utilizado, los métodos tienen correspondencia directa con el correspondiente método de la implementación de clases, puede aparecer como un estereotipo que se corresponde con una construcción en el lenguaje de programación dado. Teniendo en cuenta lo antes expuesto, se propone el diagrama de clases del diseño para el plugin vistas de los mapas. (Ver Figura 6 Diagrama de Clases del Diseño).

El diagrama de clases del diseño está integrado por la clase `view.ajax` que es la encargada del comportamiento del plugin, esta se comunica a través de ajax con el controlador frontal o `index` y este último es el encargado de instanciar las clases `CartoClient` y `CartoServer`. Estas clases son las responsables de establecer la comunicación entre las clases que intervienen entre los módulos clientes



y servidor. En el módulo cliente se encuentra la clase ClientView que se encarga de formatear los datos que son enviados a través del navegador web. En el módulo servidor se encuentra la clase ServerView que es la encargada de la iteración con el servidor de mapas y el de bases de datos. La integración entre la clase ClientView y ServerView se logra a través de las clases ViewRequest que es la encargada de describir las variables que contienen la información que forma parte de los valores de entrada de la solicitud que realiza la ClientView a la ServerView y la ViewResult que es la encargada de describir las variables que contienen la información que forma parte de los valores de entrada de la solicitud que realiza la ServerView a la ClientView.

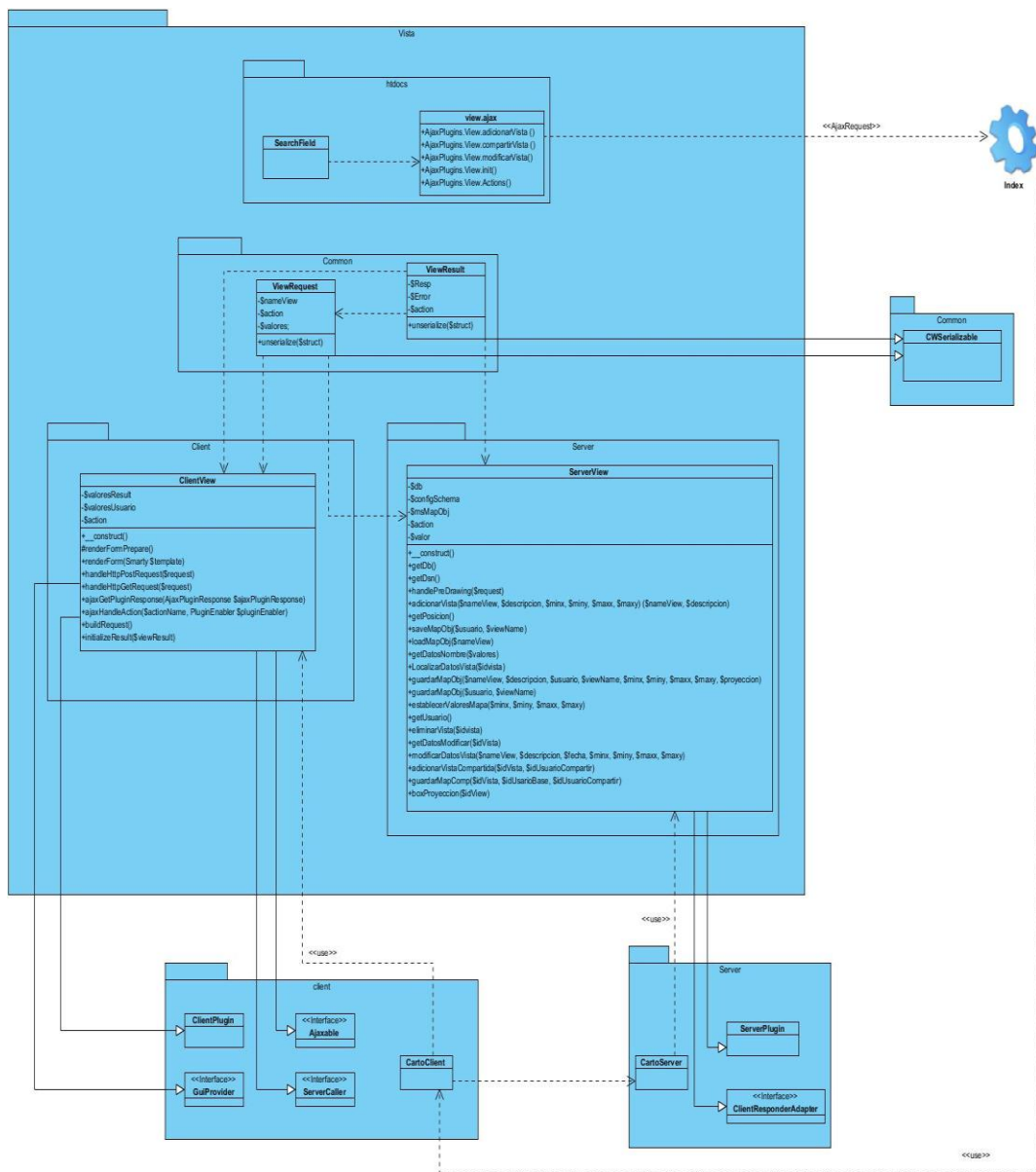


Figura 6 Diagrama de Clases del Diseño



4.4 MODELO DE DATOS

Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido, y recuperar la información. La estructura tiene que ser desarrollada para satisfacer la necesidad de las aplicaciones que la usarán, esto puede ayudar a realizar un proceso del negocio para alcanzar un valor agregado para el cliente.

A partir de las clases del diseño se pretende realizar la actividad: “Diseño de las Bases de Datos”, que tiene como propósito garantizar que los datos persistentes sean almacenados consistente y eficientemente y que sea definido el comportamiento que debe ser implementado en las bases de datos. El artefacto resultante de esta actividad es el Modelo de Datos, que describe la representación lógica y física de los datos. Se muestra a continuación una propuesta del Modelo Entidad-Relación que permite representar las entidades persistentes y las relaciones entre ellas para el módulo vistas de los mapas. (Ver Figura 4.3 Modelo Entidad-Relación)

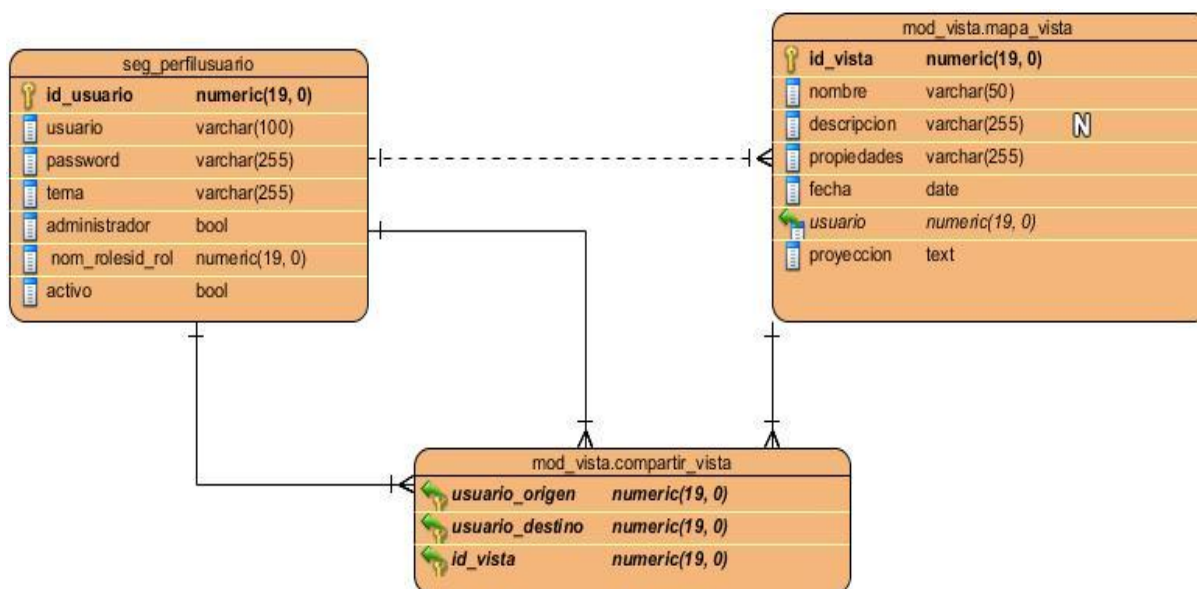


Figura 7 Modelo Entidad-Relación

Definición del Modelo Entidad Relación

seg_perfilusuario: Es la entidad que se encarga de almacenar los datos correspondientes a los usuarios.

mapa_vista: Es la entidad que se encarga de almacenar los datos correspondientes a las vistas de los mapas.



compartir_vista: Es la entidad que se encarga de almacenar los datos generados a partir de la relación entre las entidades `seg_perfilusuario` y `mapa_vista`.

seg_perfilusuario - mapa_vista: Relación de asociación binaria con una cardinalidad de uno a muchos.

seg_perfilusuario - mapa_vista: Relación de asociación ternaria, donde la entidad `seg_perfilusuario` se asocia con el rol de usuario base y usuario destino.

4.5 MODELO DE DESPLIEGUE

El Modelo de Despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos, permitiendo modelar mediante este diagrama la vista de despliegue estática, describir la arquitectura en tiempo de ejecución de procesadores, dispositivos y los componentes de software que ejecutan esta arquitectura, además de la topología del sistema, estructura de hardware y el software que se ejecuta en cada unidad.

Nodos: Es un elemento físico que existe en tiempo de ejecución, representando un recurso computacional que, por lo general, dispone de algo de memoria y con frecuencia capacidad de procesamiento. Un conjunto de componentes puede residir en un nodo y puede también migrar de un nodo a otro.

Conexiones: El tipo de comunicación es representado por un estereotipo que identifica el protocolo de comunicación o el tipo de red usado.

Elementos de anotación: Los elementos de anotación son las partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clasificar y hacer observaciones sobre cualquier elemento de un modelo. El tipo principal de anotación es la nota que simplemente es un símbolo para mostrar restricciones y comentarios junto a un elemento o un conjunto de elementos.

Se muestra a continuación el modelo de despliegue, representando cómo quedará desplegado el módulo vistas de los mapas (Ver Figura 8 Modelo de Despliegue).

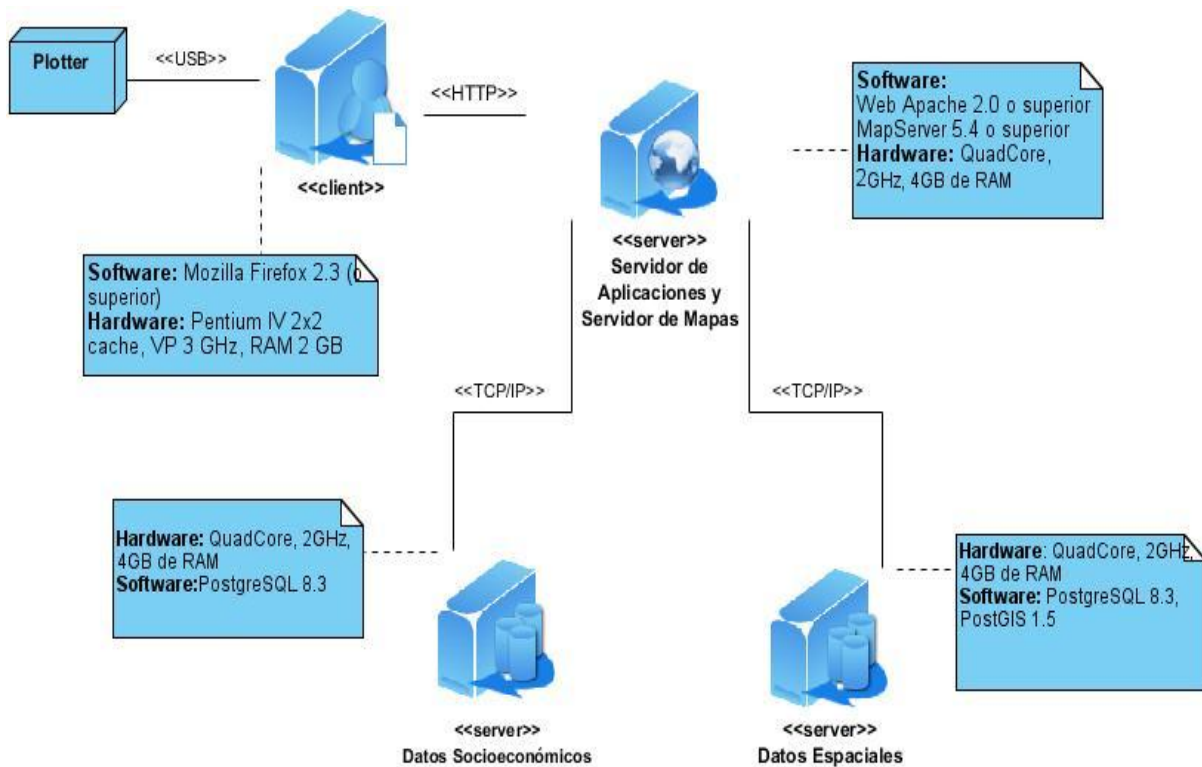


Figura 8 Modelo de Despliegue

4.6 MODELO DE IMPLEMENTACIÓN

Un componente es una unidad modular que puede reemplazarse en su propio entorno. Sus elementos internos quedan ocultos, pero tiene una o varias interfaces proporcionadas bien definidas a través de las cuales se puede obtener acceso a sus funciones. Un componente también puede tener interfaces necesarias. Es una interfaz necesaria, se definen las funciones o servicios de otros componentes que son necesarios. Mediante la conexión de las interfaces proporcionadas y las interfaces necesarias de distintos componentes, puede construirse un componente mayor. Un sistema de software completo se puede concebir como un componente. El diagrama de componentes permite concebir el diseño atendiendo a los bloques principales, ayuda al equipo de desarrollo a entender un diseño existente y crear uno nuevo. Al pensar en el sistema como una colección de componentes con interfaces proporcionadas y necesarias bien definidas, se mejora la separación entre los componentes. Esto, a su vez, facilita la comprensión y los cambios cuando se modifican los requisitos. A continuación, se propone el siguiente diagrama de componentes, para el módulo vistas de los mapas, representando de forma física el sistema.

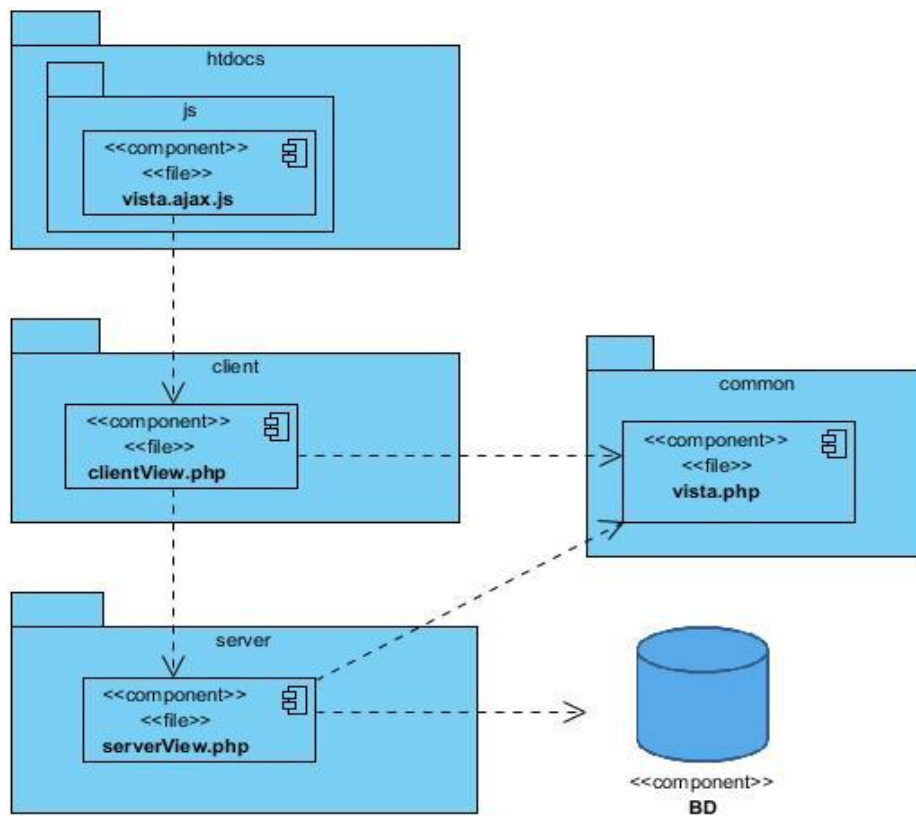


Figura 9 Diagrama de Componentes

4.7 PRUEBAS

La fase de pruebas es una de las fases más importante del ciclo de desarrollo del producto de software, porque es la encargada de validar si el producto desarrollado, cumple las expectativas funcionales, proporcionando la información necesaria sobre la calidad del producto, detectando los posibles errores que se puedan generar al hacer uso del mismo, antes de que sea entregado a un usuario final. Durante esta etapa, se realizaron 5 diseños de casos de prueba, en correspondencia con cada uno de los casos de uso del módulo vistas de los mapas, permitiendo de esta forma, detectar los errores que hasta el momento no se habían descubierto, con la menor cantidad de tiempo y de esfuerzo. Para cada caso de uso se tuvo en cuenta los diferentes escenarios y el resultado de cada una de las pruebas, permitió validar el correcto funcionamiento de cada uno de los requisitos, obteniendo como resultado la correcta implementación de los mismos. Se propone un ejemplo de los entes expuesto, en el Diseño del Caso de Prueba para el Caso de Uso: Gestionar Vista del Mapa.

Las pruebas realizadas fueron pruebas de caja negra de manera que se pudo comprobar la operatividad de cada función, es decir los valores de entradas se aceptan de forma adecuada y se produce una salida correcta. Estas pruebas se llevan a cabo sobre la interfaz del software, obviando el



comportamiento interno y la estructura del programa. Los errores que permiten encontrar están dirigidos a funciones incorrectas o ausentes, errores en la interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación.

El método aplicado de prueba de caja negra es la prueba de partición equivalente, el mismo, divide el dominio de entrada de un programa en clases de datos, a partir de las cuales deriva los casos de prueba. El diseño se procede en dos pasos:

-Se identifican las clases de equivalencia. Las clases de equivalencia son identificadas tomando cada condición de entrada (generalmente una oración o una frase en la especificación) y repartiéndola en dos o más grupos. Es de notar que dos tipos de clases de equivalencia están identificados: las clases de equivalencia válidas representan entradas válidas al programa, y las clases de equivalencia inválidas que representan el resto de los estados posibles de condición (es decir, valores erróneos de la entrada)

-Se definen los casos de prueba. Se asigna un número único a cada clase de equivalencia. Hasta que todas las clases de equivalencia válidas han sido cubiertas por los casos de prueba, se escribe un nuevo caso de prueba que cubra la clase de equivalencia válida. Y por último hasta que los casos de prueba hayan cubierto todas las clases de equivalencia inválidas, se escribe un caso de la prueba que cubra una, y solamente una, de las clases de equivalencia inválidas descubiertas.



4.7.1 DISEÑO DEL CASO DE PRUEBA PARA EL CASO DE USO: GESTIONAR VISTA DEL MAPA.

1. Descripción general.

El caso de uso inicia cuando el usuario desea adicionar, modificar, mostrar y eliminar una, o (las) vistas de los mapas y termina cuando han sido completadas todas las acciones mencionadas.

2. Condiciones de ejecución.

El usuario debe estar autenticado

3. Secciones a probar en el caso de uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: "Adicionar vista del mapa"	EC 1.1: "Adicionar nueva vista del mapa"	El usuario introduce el nombre de la vista del mapa, la descripción y da clic en el botón Guardar. El caso de uso termina cuando el sistema adiciona la nueva vista del mapa, muestra una ventana de confirmación con el mensaje "Se ha adicionado la vista satisfactoriamente" y actualiza los campos nombre, fecha, descripción, propiedades y el paginado del Grid.	Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Adicionar/Clic en el botón Aceptar.



	<p>EC 1.2: “Adicionar nueva vista del mapa dejando campos vacíos”</p>	<p>El usuario no introduce todos los datos correspondientes y da clic en el botón “Aceptar”. El sistema muestra un mensaje informando que hubo un error al introducir los datos y marca los campos vacíos en rojo.</p>	<p>Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Adicionar/Clic en el botón Aceptar</p>
	<p>EC 1.3: “Cancelar la adición de una nueva vista del mapa”</p>	<p>El usuario da clic en el botón “Cancelar”. El sistema cierra la ventana Adicionar vista del mapa.</p>	<p>Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Adicionar/Clic en el botón Cancelar.</p>
	<p>EC 1.4: “Vista en uso”</p>	<p>El usuario introduce un identificador de una vista que ya está registrada. El sistema le muestra un mensaje diciendo que la vista ya está en uso.</p>	<p>Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Adicionar/Clic en el botón Aceptar</p>
<p>SC 2: “Modificar vista del mapa”</p>	<p>EC 2.1: “Modificar vista del mapa correctamente”</p>	<p>El usuario selecciona la vista del mapa que desea modificar y da clic en el botón Modificar. El sistema muestra la ventana Modificar Vista del Mapa para que el usuario introduzca los datos y realice las modificaciones. El usuario modifica los valores que considera y da clic en el botón “Aceptar”. El sistema actualiza los datos de la vista del mapa seleccionada.</p>	<p>Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Modificar/Clic en el botón Aceptar</p>



	EC 2.2: “Modificar vista del mapa sin haber seleccionado una vista”	El usuario da clic en el botón Modificar sin haber seleccionado previamente una vista del mapa. El sistema muestra un mensaje indicándole que debe seleccionar una vista del mapa.	Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Modificar/Clic en el botón Aceptar
	EC 2.3: “Cancelar la modificación de la vista del mapa”	El usuario da clic en el botón cancelar. El sistema cierra la ventana Modificar vista del mapa y mantiene visible la ventana Gestionar vista del mapa.	Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Modificar/Clic en el botón Cancelar.
	EC 2.4: “Modificar funcionalidad incorrectamente”	El administrador selecciona la vista del mapa a modificar, introduce la denominación de una vista del mapa que ya está registrada y da clic en el botón Aceptar. El sistema muestra un mensaje indicando que esa vista del mapa ya está registrada.	Módulo Vista de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Modificar/Clic en el botón Aceptar.
SC 3: “Eliminar vista del mapa”	EC 3.1: “Eliminar vista del mapa con éxito”	El usuario selecciona la vista del mapa que desea eliminar y da clic en el botón Eliminar. El sistema muestra una ventana al usuario donde le indica que debe confirmar si está seguro que desea eliminar la vista del mapa seleccionada. El usuario da clic en el botón Aceptar. El caso de uso termina cuando el sistema elimina la vista del mapa seleccionada por el usuario y se actualiza el Grid.	Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Eliminar/Clic en el botón Aceptar.



	EC 3.2: "Eliminar vista del mapa incorrectamente"	El usuario da clic en el botón Eliminar sin haber seleccionado previamente una vista del mapa. El sistema muestra un mensaje indicándole que debe seleccionar una vista del mapa.	Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Eliminar/
	EC 2.3: "Cancelar la eliminación de una vista del mapa"	El usuario da clic en el botón cancelar. El sistema no elimina la vista del mapa y mantiene visible la ventana Gestionar Vista del Mapa.	Módulo Vistas de los Mapas/Opción Gestionar Vista del Mapa /Clic en el botón Cancelar/

Tabla 4 Diseño de Caso de Prueba del "CU Gestionar Vista del Mapa"

4. Descripción de variable.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nombre de la vista del mapa.	textfield	No	Permite introducir el nombre de la vista del mapa.
2	Descripción de la vista del mapa.	textarea	Si	Permite introducir la descripción de la vista.
3	Fecha de creación de la vista del mapa.	Combobox	No	Permite seleccionar el módulo al que pertenece.
4	Propiedades de la vista del mapa.	textfield	No	Permite seleccionar el plugin del cual depende.

Tabla 5 Descripción de las variables

**Matriz de datos****SC 1: “Adicionar nueva vista del mapa”**

ID del escenario	Escenario	Nombre de la vista del mapa	Descripción de la vista del mapa	Respuesta del sistema	Resultado de la prueba
EC 1.1	Adicionar nueva vista del mapa con éxito.	V/	V/	El sistema actualiza el grid con la vista registrada además le muestra un mensaje confirmando que la operación fue ejecutada con éxito.	Satisfactorio
EC 1.2	Adicionar nueva vista del mapa dejando campos vacíos.	l/”campo vacío”	V/	El usuario no introduce todos los datos correspondientes y da clic en el botón “Aceptar”. El sistema muestra un mensaje informando que hubo un error al introducir los datos y marca los campos vacíos en rojo.	Satisfactorio
EC 1.3	Cancelar la adición de una nueva funcionalidad	N/A	N/A	El administrador da clic en el botón “Cancelar”. El sistema cierra la ventana Adicionar vista del mapa.	Satisfactorio
EC 1.4	Vista del mapa en uso	l/” una vista ya existente”	V	El administrador introduce un identificador de la vista del mapa que ya existe. El sistema le muestra un mensaje diciendo que la denominación ya está en uso.	Satisfactorio

Tabla 6 Matriz de Datos "SC 1: Adicionar vista del mapa"



SC 2: “Modificar vista del mapa”

ID del escenario	Escenario	Nombre	Descripción	Fecha de creación	Propiedad	Respuesta del sistema	Resultado de la prueba
EC 2.1	Modificar vista del mapa correctamente	V/	V/	V/	V/	El sistema actualiza los datos de la vista modificada.	Satisfactorio
EC 2.2	Modificar vista del mapa sin haber seleccionado una vista	N/A	N/A	N/A	N/A	El sistema muestra un mensaje de error para que se seleccione una vista.	Satisfactorio
EC 2.3	Cancelar la modificación de una vista del mapa.	N/A	N/A	N/A	N/A	El sistema cierra la ventana Modificar vista del mapa.	Satisfactorio
EC2.4	Modificar vista del mapa incorrectamente	I/“Nombre en uso”	V	V	V/	El sistema muestra un mensaje indicando que el identificador introducido ya está registrado.	Satisfactorio

Tabla 7 Matriz de Datos "SC 2: Modificar vista del mapa"



SC 3: "Eliminar vista del mapa"

ID del escenario	Escenario	Vistas Registradas	Respuesta del sistema	Resultado de la prueba
EC 3.1	Eliminar funcionalidad con éxito.	V/"seleccionada la vista del mapa"	El sistema elimina la vista del mapa. El sistema muestra un mensaje después que es eliminada la vista del mapa confirmando que la acción fue ejecutada satisfactoriamente.	Satisfactorio
EC 3.2	Eliminar vista del mapa sin seleccionar la vista.	I/"no se seleccionó una vista del mapa"	El sistema muestra un mensaje diciendo que se debe seleccionar una vista del mapa.	Satisfactorio
EC 3.3	Cancelar la eliminación de una vista del mapa	V/"seleccionada una vista del mapa"	El usuario da clic en el botón cancelar. El sistema no elimina la vista del mapa y mantiene visible la ventana Gestionar vista del mapa.	Satisfactorio

Tabla 8 Matriz de datos "SC 3: Eliminar vista del mapa"



CONCLUSIONES PARCIALES

En el presente capítulo, se sientan las bases para dar paso a la implementación del plugin “Vistas de los mapas”. Teniendo en cuenta, que se definió el esqueleto arquitectónico del sistema, el cual permitió tener una vista lógica de la estructura y comportamiento del plugin, para analizar la efectividad del diseño y cumplir con los requisitos establecidos. Se realizó una propuesta del diagrama de paquetes, permitiendo representar el sistema en agrupaciones lógicas y las dependencias entre las mismas. Se presentó el modelo de diseño, generando el diagrama de clases del diseño, el cual permitió mostrar una abstracción de las clases que posteriormente serán definidas durante la implementación del plugin. Se elaboró el modelo de datos, a partir del diagrama Entidad-Relación, el cual permitió describir la forma en que será almacenada la información referente a las vistas de los mapas en las Bases de Datos y quedó representado el modelo de despliegue, que permitió definir la distribución física del sistema y el modelo de implementación, a partir, del diagrama de componentes que reflejó la parte modular, desplegable y reemplazable, que encapsula la implementación.



CONCLUSIONES GENERALES

Las soluciones para muchos problemas requieren acceso a varios tipos de información que están relacionada con la geografía. Los SIG permiten administrar este tipo de información, analizar patrones, relaciones y tendencia de la misma, con el propósito de contribuir a la toma de mejores decisiones. Al finalizar el presente trabajo, se consideran cumplido el objetivo propuesto para la investigación y la idea a defender, arribando a las siguientes conclusiones:

El estudio y análisis de las soluciones existentes, permitió profundizar en los conocimientos sobre la forma de generación de las vistas de los mapas en los SIG.

La identificación y utilización correcta de las tecnologías, metodología y herramientas, permitió organizar todo el proceso de desarrollo del módulo, generar todos los artefactos necesarios para su construcción y facilitar la implementación del sistema.

El Modelo de Casos de Uso, permitió describir las funcionalidades que provee el módulo vistas de los mapas, en correspondencia, con las necesidades del usuario final.

El Modelo de Diseño, permitió describir cómo quedarían implementadas las clases y la comunicación existente entre ellas.

El módulo “Vistas de los Mapas” para la plataforma GeneSIG, brinda la posibilidad de:

- . Simplificar la búsqueda de ubicaciones geográficas en el mapa.
- . Compartir vistas de los mapas con usuarios de la aplicación.
- . Localizar con exactitud y de forma rápida la vista almacenada.

La validación de la solución propuesta, a partir de las pruebas de caja negra, permitió detectar los posibles errores y obtener un producto con la calidad adecuada.

El usuario que acceda a la aplicación puede trabajar con el módulo y almacenar las vistas de los mapas que sean convenientes, logrando de esta forma complementar el objetivo trazado al iniciar la investigación.



RECOMENDACIONES

Con el objetivo de mejorar los resultados obtenidos durante el desarrollo del presente trabajo, se propone las siguientes recomendaciones para las futuras investigaciones que se deseen realizar:

1. Aplicar el nuevo módulo de “Vistas de los mapas”, en los proyectos productivos que desarrollen Sistemas de Información Geográfica en la UCI y en Cuba, con perfiles más específicos, que utilicen la Plataforma GeneSIG.
2. Para la mejora de la solución presentada, incorporar nuevos elementos funcionales, como gestionar vistas de capas, de barras, ubicar una imagen para las vistas del mapa y modificar otras propiedades que tiene incorporada el mapa a través del módulo.
3. Para lograr mayor aprovechamiento de las funcionalidades que provee el subsistema, continuar estudios investigativos de los múltiples beneficios que son proporcionados por el mismo.



REFERENCIAS BIBLIOGRÁFICAS

- Bernhardsen, Tor. 2001.** *Geographic information systems:an introduction*. 2001. 0-471-41968-0.
- Corporation, MapInfo. 2003.** *MapInfo Professional, Guía del usuario v7.5*. EE.UU : s.n., 2003.
- Corporation, Microsoft. 2005.** *Diccionario de informática e internet de Microsoft*. s.l. : Vuelapluma, S. L trad, 2005. 84-481-4531-3.
- ENRI. 1995.** ArcGIS Resources Center. [En línea] ESRI, 1995. <http://help.arcgis.com/es/arcgisdesktop/10.0/help/index.html#/na/00r90000001n000000/>. 92373-8100.
- Esri. 1995.** gis.com. [En línea] United States of America, 1995. <http://www.gis.com/content/what-gis>.
- Ivar Jacobson, Grady Booch, James Rumbaugh. 2004.** *El Proceso Unificado de Desarrollo de Software*. Habana : Félix Varela, 2004. 0-201-89535-8.
- 2012.** NetBeans. [En línea] 2012. <http://netbeans.org/features/ide/index.html>.
- NOTAS O APUNTES SOBRE EL DESARROLLO DE LA INDUSTRIA CUBANA DEL SOFTWARE – ICSW. [En línea] eva.uci.cu.
- Orallo, Enrique Hernández. 2011.** *El Lenguaje Unificado de Modelado UML* . 2011.
- OSGeoLive. 2011. [En línea] http://live.osgeo.org/es/overview/postgis_overview.html.
- Oyala, Víctor. 2010.** *Sistemas de Información Geográfica* . 2010.
- Powell, Thomas A. 2001.** *HTML 4 Manual de referencia* . España : s.n., 2001. 84-481-3172-X.
- Rafael Martínez. 2010.** PostgreSQL-es. [En línea] 02 de 10 de 2010. http://www.postgresql.org.es/sobre_postgresql.
- Salinas, J. 2007.** *Información Geográfica, Software Libre e Infraestructuras de Datos Espaciales*. Cuba : s.n., 2007.
- WordPress. 2008.** Definición.de. [En línea] 2008. <http://definicion.de/ubicacion/>.



BIBLIOGRAFÍA

2009. Asociación gvSIG. [En línea] 2009. <http://www.gvsig.com/productos/gvsig-desktop>.

autores, colectivo de. 1999. masadelante.com. [En línea] 1999. <http://www.masadelante.com/faqs/plugin-in>.

BEIT. 2006. Servicios TIC by BEIT. *Servicios TIC by BEIT*. [En línea] 2006. <http://www.serviciostic.com/las-tic/definicion-de-tic.html>.

Bernhardsen, Tor. 2001. *Geographic information systems:an introduction*. 2001. 0-471-41968-0.

Camy, Lazaro Issi. 2002. *La biblia de javaScript*. Madrid : ANAYA MULTIMEDIA , 2002. 84-415-1384-8.

Ciencia, Sociedad y Tecnología. Colectivo de Autores representados por el Dr. Fidel Castro Díaz-Balart. 2002. s.l. : Debate, 2002, Cuba. Amanecer del Tercer Milenio. 959-05-0283-0.

Computación., Departamento de Sistemas Informáticos y. 2011. *Metodología y Tecnología de la Programación*. Universidad Politécnica de Valencia : s.n., 2011.

Córdoba, Fernando García. 2001. *La tesis y el trabajo de tesis: recomendaciones metodológicas para la elaboración de trabajos de trabajos de tesis*. México : Limusa, 2001. 968-18-6235-X.

Corporation, MapInfo. 2003. *MapInfo Professional, Guía del usuario v7.5*. EE.UU : s.n., 2003.

Corporation, Microsoft. 2005. *Diccionario de informática e internet de Microsoft*. s.l. : Vuelapluma, S. L trad, 2005. 84-481-4531-3.

David Andes . 1999. GIS Lounge and DM Geographics. [En línea] 1999. <http://gislounge.com/the-components-of-gis-evolve/>.

Desoft S.A. 2002-2010. Ministerio de la Informática y las Comunicaciones. [En línea] 2002-2010. <http://www.mic.gov.cu/sitiomic/servlet/hinfosocplan>.

Díaz, José R. Hernández Santana y Jorge Luis Díaz. 1994. Revista Internacional de Ciencias de la Tierra. *INSTITUTO DE GEOGRAFÍA DE LA ACADEMIA DE CIENCIAS DE CUBA: UNA NUEVA PROYECCIÓN CIENTÍFICA HACIA EL SIGLO XXI*. [En línea] marzo de 1994. http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1123#0.

2011. EcuRed. [En línea] 2011. <http://www.ecured.cu/index.php/PostGreSQL>.

EcuRed . 2011. EcuRed. [En línea] 2011. http://www.ecured.cu/index.php/Proceso_Unificado_de_Desarrollo.

Emilio Avilés Ávila. Curso UML. [En línea] <http://www.slideshare.net/techmi/curso-uml-25-diagramas-de-implementación>.



- ENRI. 1995.** ArcGIS Resources Center. [En línea] ESRI, 1995. [http://help.arcgis.com/es/arcgisdesktop/10.0/help/index.html#/na/00r90000001n000000/](http://help.arcgis.com/es/arcgisdesktop/10.0/help/index.html#/na/00r90000001n000000/.). 92373-8100.
- ESRI. 1995.** gis.com. [En línea] United States of America, 1995. <http://www.gis.com/content/what-gis>.
- Ivan, Reyes Lomeli Walter. 2007.** Optimización de la Web del ITM, Aplicación servidor Web . [En línea] marzo de 2007. <http://es.scribd.com/doc/52208534/29/CARACTERISTICAS-Y-VENTAJAS-DEL-APACHE>.
- Ivar Jacobson, Grady Booch, James Rumbaugh. 2004.** *El Proceso Unificado de Desarrollo de Software*. Habana : Félix Varela, 2004. 0-201-89535-8.
- Jorge Elías. 2010.** Proyecto Pragmalia. *Implementar el Sistema de información Geográfica (SIG) y el Mapa de Riesgo Medio Ambiental como herramienta de desarrollo*. [En línea] 28 de 10 de 2010. [Citado el: 30 de 11 de 2011.] <http://proyectopragmalia.blogspot.com/2010/10/302-implementar-el-sig-y-el-mapa-de.html>.
- 2006 . Lambda .** [En línea] 2006 . [Citado el: 27 de febrero de 2012.] <http://www.lambdasi.com.ar/textocomp.asp?id=326>.
- 1997.** *Ley No.81 del Medio Ambiente*. s.l. : Gaceta oficial de la República de Cuba, 1997.
- Lime, Steve. 2011.** MapServer open source web mapping . [En línea] 2011. <http://mapserver.org/es/mapfile/index.html>.
- Miguel Angel Alvarez. 2004.** DesarrolloWeb.com:. [En línea] 16 de noviembre de 2004. <http://www.desarrolloweb.com/articulos/1696.php>.
- 2012.** NetBeans. [En línea] 2012. <http://netbeans.org/features/ide/index.html>.
- NOTAS O APUNTES SOBRE EL DESARROLLO DE LA INDUSTRIA CUBANA DEL SOFTWARE – ICSW. [En línea] eva.uci.cu.
- Omar Sánchez .** Modelos, Control y Sistema de Visión . [En línea] <http://omarsanchez.net/conceptomod.aspx>.
- Orallo, Enrique Hernández. 2011.** *El Lenguaje Unificado de Modelado UML* . 2011.
- OSGeoLive. 2011. [En línea] http://live.osgeo.org/es/overview/postgis_overview.html.
- Oyala, Víctor. 2010.** *Sistemas de Información Geográfica* . 2010.
- Powell, Thomas A. 2001.** *HTML 4 Manual de referencia* . España : s.n., 2001. 84-481-3172-X.
- Radcliffe, Mr. Mark. 2000.** Open Source Initiative . [En línea] 2000. <http://www.opensource.org/docs/osd>.
- Rafael Martinez. 2010.** PostgreSQL-es. [En línea] 02 de 10 de 2010. http://www.postgresql.org.es/sobre_postgresql.



2011. redhat. [En línea] 2011. http://docs.redhat.com/docs/es-ES/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/ch-httpd.html#s1-httpd-v2.

Rico, Francisco. 2000. *Gran diccionario de la Lengua Materna Española*. Barcelona : Larousse editorial, S.A, 2000. 84-8016-496-4.

rogetrips. 2008. El Blog de rogetrips. [En línea] febrero de 2008. <http://rogetrips.wordpress.com/2008/02/04/postgresql-83-lanzado/>.

Salinas, J. 2007. *Información Geográfica, Software Libre e Infraestructuras de Datos Espaciales*. Cuba : s.n., 2007.

2008. Servidor de mapas. [En línea] 2008. <http://dspace.ups.edu.ec/bitstream/123456789/56/10/Capitulo4.pdf>.

Silva, José Luis Batista. 2005. Revista Internacional de Ciencias de la Tierra. *APLICACIÓN DE SISTEMAS DE INFORMACIÓN GEOGRÁFICA EN CUBA*. [En línea] noviembre de 2005. http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051.

Thomson, Luke Welling y Laura. 2003. *Desarrollo Web con PHP y MySQL*. Madrid : ANAYA MULTIMEDIA , 2003. 84-415-1569-7.

VENEGAS, MARIO ANDRÉS ALBORNOZ. 2006. *CATASTRO COMPLEMENTADO CON INFORMACIÓN SOBRE USO DE RECINTOS Y PUBLICACIÓN WEB UTILIZANDO MAPSERVER COMO HERRAMIENTA DEL PLAN MAESTRO DE LA UNIVERSIDAD DE SANTIAGO DE CHILE*. Santiago de Chile : s.n., 2006.

Visual Paradigm. 2009. Visual Paradigm for UML 8.3. [En línea] 2009. <http://www.visual-paradigm.com/product/vpuml/>.

WordPress. 2008. Definición.de. [En línea] 2008. <http://definicion.de/ubicacion/>.