

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6



**SOLUCIÓN INFORMÁTICA PARA LA INTEGRACIÓN DEL
SISTEMA DE INFORMACIÓN GEOGRÁFICA GEOQ CON EL
SERVIDOR DE MAPAS MAPSERVER.**

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

AUTORES: Daniel Betancourt Mendoza.
Milayne Cáceres Sánchez

TUTOR: Ing. Alberto Menéndez Romero.

Ciudad de La Habana, 24 de junio de 2011

Año del 53 Aniversario de la Revolución.

Declaración de Autoría

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Geoinformática de la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autores: Daniel Betancourt Mendoza

Milayne Cáceres Sánchez

Tutor: Ing. Alberto Menéndez Romero.

Producto al avance de las Tecnologías de la Información y la Comunicación (TIC) hubo un cúmulo enorme de la información, por lo que los mapas por sí solos no eran suficientes para representarla , de manera que se hizo necesario la creación de un sistema que fuera capaz de manejar el contenido que se guardaba en ellos, de forma tal que se pudieran explotar todas sus potencialidades, es así como surgen lo que hoy se conoce como Sistema de Información Geográfica (SIG) o GIS por sus siglas en inglés.

Con la presente investigación se pretende desarrollar una aplicación informática de utilidad para el Centro de Investigación y Desarrollo GEySED ubicado en la facultad 6, específicamente al proyecto SIG Desktop de que permita integrar el SIG de escritorio GeoQ con el servidor de mapas MapServer.

El aprovechamiento máximo de la aplicación traerá consigo que los usuarios al trabajar con el SIG GeoQ obtengan una visualización semejante a la que alcanzan en la plataforma GENESIG que evitará que existan inconformidades y cuestionamientos, es decir, proveerá un punto de integración transparente al usuario, entre las aplicaciones SIG-Desktop y SIG-Web de la plataforma GeneSIG.

Palabras claves:

Aplicación informática, Sistema de Información Geográfica, visualización.

INTRODUCCIÓN	1
1.1 INTRODUCCIÓN	1
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	1
1.3 OBJETO DE ESTUDIO	3
1.3.1 Descripción general	3
1.3.2 Descripción actual del dominio del problema	3
1.3.3 Situación Problemática	3
1.4 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES	4
1.5 CONCLUSIONES	5
2.1 INTRODUCCIÓN	6
2.2 APLICACIÓN INFORMÁTICA	6
2.2.1 Aplicación de escritorio	6
2.3 ARQUITECTURA DE SOFTWARE	7
2.3.1 MODELO VISTA CONTROLADOR (MVC)	7
2.4 LENGUAJES DE PROGRAMACIÓN	8
2.4.1 LENGUAJE PYTHON	8
2.5 PLUGINS PARA GEOQ UTILIZANDO EL LENGUAJE DE PROGRAMACIÓN PYTHON ...	9
2.6 FRAMEWORKS DE DESARROLLO	10
2.7 METODOLOGÍAS DE DESARROLLO	11
2.7.1 Metodologías ágiles	12
2.7.1.1 RUP Ágil (AUP)	12
2.8 EL LENGUAJE DE MODELADO (UML)	13
2.9 HERRAMIENTA CASE	14
2. 10 CONCLUSIONES	16
3.1 INTRODUCCIÓN	17
3.1.1 MODELO DEL DOMINIO EN EL DESARROLLO DEL SOFTWARE	17
3.2 ENTORNO DONDE TRABAJARÁ EL SISTEMA	17
3.3 REQUISITOS FUNCIONALES	19
3.3.1 ESTRATEGIA DE CAPTURA DE REQUISITOS	20
3.4 REQUISITOS FUNCIONALES (RF)	20
3.5 REQUISITOS NO FUNCIONALES (RNF)	21
3.5 DESCRIPCIÓN DEL SISTEMA PROPUESTO	22
3.6.2 Diagrama de Caso de uso del sistema	23
3.6.3 Descripción textual de los casos de uso del Sistema (Ver Anexo 2).	24
3.7 CONCLUSIONES	27
4.1 INTRODUCCIÓN	28
4.2 PRINCIPIOS DE DISEÑO	28
4.3 DIAGRAMA DE CLASE DEL DISEÑO	29
BIBLIOGRAFÍA CITADA	39
BIBLIOGRAFÍA CONSULTADA	40

ANEXO 1 ARQUITECTURA MODELO VISTA CONTROLADOR	45
ANEXO 2 DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO.....	46
ANEXO 3 DIAGRAMA DE CLASES DEL DISEÑO.....	55
ANEXO 4 DIAGRAMAS DE SECUENCIA DEL DISEÑO	58
ANEXO 5 CASOS DE PRUEBA.....	62

INTRODUCCIÓN

Los seres humanos del planeta Tierra representan un tamaño insignificante comparado con este, de modo que sólo se aprecia lo habitual y cotidiano y así queda imaginar lo demás. Gran parte de las veces se cree en cosas que se dicen sin nunca antes haberlas visto o sin saber realmente si existe o no. Gracias a la gran inteligencia del hombre se ha logrado ver mucho más allá de lo que pueden alcanzar los ojos. Con el uso de la tecnología muchas cosas que antes eran simplemente un misterio hoy en día son parte del quehacer diario; por lo que se convierte en una necesidad saber la mayor cantidad de información posible acerca de este paraíso (Davis).

Hoy día se conoce que el planeta es redondo y tres cuartas partes están formadas por agua. ¿Pero siempre fue así? En la antigüedad se creía que la tierra era una palangana con agua, donde la mayor parte era tierra y más allá de sus bordes solo existía el abismo (AGUINIS). Era necesario saber cómo representar lugares realmente grandes en objetos pequeños y fáciles de manipular. Es así como surge un nuevo medio para portar información: los mapas. Se cree que el primer mapa fue realizado en el siglo VI a.C. por el filósofo griego Anaximandro; tenía forma circular y mostraba el mundo conocido agrupado en torno al mar Egeo y rodeado por el océano.

La precisión de los mapas posteriores aumentó mucho debido a las determinaciones más precisas sobre latitud y longitud y a los cálculos sobre el tamaño y forma de la Tierra. Pero los mapas por si solos no eran suficientes para mostrar toda la información que el hombre quería representar, era necesaria la creación de un sistema que fuera capaz de manipular la información de estos mapas y realizar operaciones explotando toda su potencialidad dando paso así a los Sistemas de Información.

Con el surgimiento de las Tecnologías de la Información y la Comunicación (en lo adelante TIC), la informática no se quedó atrás, desarrollando una potencialidad en el trabajo con los mapas, dando origen a los Sistemas de Información Geográfica (en lo adelante SIG).

Los SIG, surgieron hace 25 años atrás por la necesidad de almacenar, manipular y visualizar datos que tengan un componente geográfico. Desde su aparición tuvieron

como objetivo permitir a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones. En la actualidad son utilizados por muchas empresas debido a su gran utilidad.

En Cuba existen instituciones y organizaciones que se destacan por el uso de los SIG entre las que se encuentran: GEOCUBA, el Ministerio de Ciencia, Tecnología y Medio Ambiente e incluso las Fuerzas Armadas Revolucionarias (FAR), cada una de ellas orientada a esferas como son: la telefonía, la minería, la meteorología, el transporte, la salud, etc., en esta última se han explotado principalmente los llamados SIG en epidemiología o simplemente SIG-Epi que sirven para determinar la mortalidad infantil de una región específica, así como para calcular la superficie de un polígono territorial. Estos procesos buscan lograr más eficacia y eficiencia, que permitan una mayor generación de riquezas y hagan sustentable el aumento sistemático de la calidad de vida de los cubanos; sin embargo muchas veces se hace difícil el empleo de los mismos debido a que son sistemas privativos, lo que imposibilita tener acceso pleno al código fuente o de lo contrario son sistemas antiguos que ya no se les brinda soporte. Por estas limitaciones se ha hecho necesaria la creación de un SIG que sea capaz de mitigar estos problemas.

La Universidad de las Ciencias Informáticas (en lo adelante UCI) como parte del proceso de informatización de la sociedad incluyó dentro de su ámbito de trabajo en el curso 2006-2007 los SIG, dentro de los cuales se pueden mencionar: SIG-UCI y la Plataforma GENESIG, esta última constituye la base para el desarrollo de SIG soportados sobre la Web. Los SIG para la web brindan facilidades para el acceso de la información ya que puede ser accedida desde diferentes lugares; sin embargo presentan problemas para realizar la edición de mapas. Por tal motivo a la facultad 6 se le asignó la tarea de confeccionar una aplicación de escritorio capaz de efectuar las mismas funciones que GENESIG e implementar nuevas funcionalidades que sean capaces de eliminar las limitantes que hasta hoy presenta la plataforma. Cabe destacar que la plataforma GENESIG trabaja con el motor de mapas MapServer, conocido como un referente en el servicio de mapas web pudiéndose utilizar desde

varios lenguajes de programación; a pesar de ello MapServer no ha sido incluido en un SIG de Escritorio como motor de mapas.

Para darle cumplimiento a esta tarea se creó el proyecto SIG Desktop, de cuyo esfuerzo y dedicación se obtuvo GeoQ que no es más que un SIG de escritorio que utiliza como base al SIG Quantum GIS el cual cuenta con su propio servidor de mapas. Esto trae consigo que al trabajar con algunos de los SIG creados con la plataforma GENESIG y con GeoQ, en una misma cartografía se obtienen resultados distintos desde el punto de vista visual. Como consecuencia, las personas que utilizan ambas aplicaciones, tanto web como escritorio, no tienen una uniformidad visual incluso trabajando en el mismo mapa, lo que en ocasiones pudiera generar resultados de análisis incorrectos.

Por lo anteriormente planteado se define como *problema científico*: ¿Cómo lograr la integración del SIG de escritorio GeoQ con el servidor de mapas MapServer?

Planteándose como *objeto de estudio*: Los procedimientos de integración de MapServer con los Sistemas de Información Geográfica, específicamente en los mecanismos de integración de MapServer con los Sistemas de Información Geográfica tipo escritorio lo que constituye *el campo de acción* de la investigación.

Se define como *objetivo general* del presente trabajo de diploma: Desarrollar una solución informática que permita la integración del SIG GeoQ con el servidor de mapas MapServer.

Como parte de esta investigación se defiende la siguiente *idea*:

El desarrollo de una solución informática que permita la integración de GeoQ con MapServer proveerá un punto de integración transparente al usuario, entre las aplicaciones SIG-Desktop y SIG-Web de la plataforma GeneSIG.

Para lograr darle cumplimiento absoluto a los objetivos anteriormente expuestos se desarrollarán las siguientes tareas en la presente investigación:

1. Caracterizar los mecanismos de integración del servidor de mapas MapServer con los sistemas de Información Geográfica tipo Escritorio.
2. Caracterizar las tendencias y tecnologías actuales a desarrollar.
3. Identificar las funcionalidades que debe brindar la solución que se propone.
4. Diseñar la solución que se propone.
5. Construir la solución propuesta.
6. Realizar las pruebas correspondientes para validar la construcción de la solución propuesta.

Con la culminación de todas las tareas y actividades desarrolladas luego de un largo proceso investigativo se espera obtener los siguientes resultados:

1. Solución informática para la integración del Sistema de Información Geográfica GeoQ con el servidor de mapas MapServer.
2. Documentación técnica asociada al proceso ingenieril durante la construcción de la solución informática.

A lo largo de esta investigación se tendrán en cuenta métodos científicos que serán de suma importancia para la realización satisfactoria de la misma que a continuación se presentan:

1. **Histórico – lógico:** Se utilizará para el estudio de los procesos de gestión de información, trabajos e investigaciones anteriores, es decir para un mejor entendimiento y análisis del estado del arte lo que permitirá un eficaz desarrollo de la presente investigación.
2. **Modelación:** Servirá de guía para la representación mediante diagramas del modelo del sistema en términos de casos de uso y actores
3. **Analítico - Sintético:** Se utilizará para el estudio de la bibliografía y la selección de la más adecuada para el desarrollo del presente trabajo; además, para dividir el problema de investigación en elementos por separado, profundizar en el estudio de cada elemento y luego sintetizarlos en la confección de la solución propuesta.

El presente trabajo está dividido en 4 capítulos fundamentales:

Capítulo 1: Se expone el objeto de estudio, la descripción del entorno donde se encuentra el negocio y su organización, se describe detalladamente la situación problemática y el análisis de otras soluciones que puedan brindar respuesta al problema científico planteado en el presente trabajo.

Capítulo 2: Se explican las principales tecnologías, lenguajes de programación y herramientas que se utilizarán para la construcción de la solución propuesta, así como las ventajas de utilizarla.

Capítulo 3: Se describe la solución propuesta, se seleccionan los actores, los trabajadores, se describe el negocio, los requisitos funcionales y no funcionales y los casos de uso del sistema.

Capítulo 4: Se plantea la construcción propuesta en el Capítulo 3 en función de diagramas de clases y estándares del diseño, generalidades de la implementación, diseño de la base de datos, modelo de despliegue y modelo de implementación, finalizando el capítulo con los diseños de casos de pruebas que validen la implementación.

Capítulo 1 Fundamentación Teórica

1.1 Introducción

En este capítulo se expone el objeto de estudio, la descripción del entorno donde se encuentra el negocio y su organización, se describe detalladamente la situación problemática y se analizan otras soluciones existentes que puedan brindar respuesta al problema científico planteado. Se hace además mención de los principales conceptos asociados al dominio de dicho problema.

1.2 Conceptos asociados al dominio del problema

Los conceptos asociados al dominio del problema son aquellos ayudan a comprender de manera precisa el contexto del negocio. Sirven de punto de inicio para el desarrollo exitoso de la investigación.

Sistema de información

Es un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopila, elabora y distribuye la información necesaria para las operaciones de dicha empresa y para actividades de dirección y control, correspondientes para desempeñar su actividad de acuerdo a sus estrategias de negocio. **(Peralta, 2006)**

Sistema de información geográfica

Es una colección de software que permite crear, visualizar, consultar y analizar datos geoespaciales. Particulariza un conjunto de procedimientos sobre una base de datos no gráfica o descriptiva de objetos del mundo real que tienen una representación gráfica y que son susceptibles de algún tipo de medición respecto a su tamaño y dimensión relativa a la superficie de la tierra. Cuenta además con una base de datos gráfica con información georreferenciada o de tipo espacial y de alguna forma ligada a la base de datos descriptiva. La información es considerada geográfica si es medible y tiene localización. **(Collins, 1993)**

Capítulo 1 Fundamentación Teórica

Datos geoespaciales

Se refieren a información sobre la localización geográfica de una entidad. Esto habitualmente implica el uso de una coordenada geográfica, como los valores latitud o longitud. **(Córdova)**

GeoQ

Es un SIG de Escritorio, del proyecto SIG Desktop en el departamento Geoinformática de la Facultad 6 de la UCI, que se basa en el SIG Quantum GIS, el cual es un “proyecto que nació en mayo del 2009.”

Fichero .Map

En este fichero, los datos del mapa se organizan en capas, a su vez dividida en una o más clases, donde en cada una de las cuales se pueden definir diferentes estilos visuales. Esta estructura permite la generación de mapas con una definición de estilos muy flexible, que también puede depender de la escala del mapa. También incluye la posibilidad de fusionar la producción de una plantilla de HTML MapServer, para generar una página web de lectura fácil y agradable. **(Fawcett & Butler)**

MapServer

Es un entorno de desarrollo en código abierto (Open Source Initiative) para la creación de aplicaciones SIG en Internet/Intranet con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Internet Map Server (IMS). Su funcionamiento básico está configurado en un fichero de texto, que generalmente tiene la extensión ".map". **(Fawcett & Butler)**

SIG Desktop

Proyecto creado en la facultad 6 en el Centro de Investigación y Desarrollo GEySED cuya misión fundamental es la construcción de un SIG de escritorio para fortalecer la plataforma GENESIG.

1.3 Objeto de estudio

1.3.1 Descripción general

La UCI, específicamente la facultad 6, cuenta con un departamento de Geoinformática el cual tiene entre sus planes poner al mercado una infraestructura para garantizar la representación de datos geospaciales a través de diferentes plataformas; con lo que cuenta en estos momentos con una solución para la Web probada en diferentes proyectos; esta solución utiliza como motor de representación gráfica el servidor de mapas MapServer.

El departamento pretende además el desarrollo de una solución tipo Escritorio para garantizar funcionalidades que no son alcanzables de manera razonable por soluciones Web. Para ello se pretende tomar de base al sistema GeoQ que está desarrollado sobre la base del SIG Quantum GIS.

Teniendo en cuenta que las soluciones de entorno Web utilizan como motor de representación gráfica el servidor de mapas MapServer se debería garantizar que las personalizaciones de GeoQ dispongan de funcionalidades similares para lograr la integración entre ambas partiendo de los orígenes de datos.

1.3.2 Descripción actual del dominio del problema

GeoQ al estar implementado sobre la base del Quantum GIS tiene un motor de visualización propio lo que posibilita la representación de datos en forma de vectores o imágenes permitiendo el acceso y edición de información en diferentes bases de datos geográficos como PostgreSQL/PostGIS y SQLite/SpatialLite. Esto hace posible que sea capaz de realizar los procesos de geoprocésamiento y georreferenciación. Otra de las ventajas que proporciona es el uso herramientas de análisis complejos como fTool y GDAL.

1.3.3 Situación Problemática

Actualmente los SIG existentes en la UCI trabajan con MapServer como servidor de mapas todos soportados en un entorno Web pero ninguno para aplicaciones de

entorno escritorio. Todas estas soluciones o podrían llamarse personalizaciones del tipo Web han sido creadas sobre la base de la plataforma GENESIG la cual hoy en día presenta problemas para la realización de ediciones de mapas.

GeoQ es un SIG para entorno de escritorio que emplea su propio servidor de mapas, lo que trae como consecuencia que los usuarios que trabajan con soluciones Web obtengan una visualización diferente de la que obtienen de las aplicaciones de escritorio. Esta situación provoca cuestionamiento e inconformidad para los beneficiarios que en algunos casos sabrán el motivo por el cual esto ocurre si están empleando la misma cartografía, pero la gran mayoría sólo se preguntará qué están haciendo mal. La realidad es que no existe hoy un mecanismo que sea capaz de integrar el SIG GeoQ con el servidor de mapas MapServer.

1.4 Análisis de otras soluciones existentes

A lo largo de la investigación y después de búsquedas por Internet se revelaron soluciones que sirven de guía para un satisfactorio desarrollo. Las respuestas de muchas preguntas se convirtieron de abstractas a concretas al descubrir que actualmente MapServer combina su instalación con lenguajes de programación tales como: PHP, Java, Python y C#, además con gestores de base de datos como PostgreSQL y MySQL.

Para facilitar el trabajo se utilizan aplicaciones que complementan las funcionalidades de MapServer, como MapLab que provee herramientas Web para crear y administrar aplicaciones de mapeo y archivos de mapas e interfaz gráfica para crear y editar los mismos. Sin embargo no se tiene conocimiento alguno sobre la existencia de aplicaciones que realicen la vinculación del SIG GeoQ con el servidor de mapas MapServer, actualmente existe una solución que permite exportar los datos salvándolos en un archivo .MAP

No existe hoy ninguna solución que sea capaz de integrar el servidor de mapas MapServer con el SIG de escritorio GeoQ por lo que esta investigación proporcionaría la primera de su tipo.

1.5 Conclusiones

Durante el desarrollo de este capítulo de la investigación se analizaron las soluciones existentes que integran a MapServer con disímiles lenguaje de programación y gestores de base de datos. Además se tuvo en consideración un conjunto de conceptos importantes para un mejor entendimiento del contexto donde coexiste el negocio posibilitando una correcta descripción de la situación problemática y, como parte del análisis exhaustivo, se describió de manera general el objeto de estudio del presente trabajo de diploma.

2.1 Introducción

En este capítulo se explican las principales tecnologías, lenguajes de programación y herramientas que se utilizarán para la construcción de la solución propuesta, así como las ventajas de utilizarlas.

2.2 Aplicación informática

Una aplicación informática es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación. **[Std. 729, IEEE].**

Una aplicación informática o software no es solamente programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta. **[Sommerville, 2004].**

Un programa informático, sinónimo de software, es el conjunto de instrucciones que ejecuta un ordenador o computadora donde una vez ejecutadas realizarán una o varias tareas (**Stair, Ralph M, 2003**); también suele referirse al código fuente original o a la versión ejecutable.

2.2.1 Aplicación de escritorio

Cuando se refiere al término aplicación de escritorio o desktop se está en presencia de una aplicación informática que se ejecuta en una computadora de escritorio o en una portátil (laptop) de manera local, en contraste a las aplicaciones web.

Este tipo de aplicación informática se ejecuta en el ordenador en su propia dirección de memoria física o virtual y no requiere cargarse en otras aplicaciones como navegadores para poder funcionar. No obstante, una aplicación de escritorio puede encontrarse distribuida en varios ordenadores y comunicarse a través de diferentes mecanismos de la red, tales como TCP/IP, UDP, RPC, entre otros; además de poder hacer uso de servidores de almacenamiento de datos.

Entre las ventajas de las aplicaciones de escritorio, se destacan:

- Mayor capacidad gráfica visual en comparación con las aplicaciones Web debido a que el HTML que es el lenguaje que entienden estos programas no ofrece Interfaces mucho mejores que la de escritorio.
- Menor tiempo de respuesta (aplicación más rápida).

2.3 Arquitectura de Software

Bass, Paul Clement y Kazman precisan que la arquitectura de software es como la estructura del sistema que incluyen los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos. Además hacen referencia que dos de las razones claves por las cuales es importante la arquitectura del software son:

- Las representaciones de la arquitectura del software permiten la comunicación entre todas las partes (participantes) interesadas en el desarrollo de un sistema de cómputo.
- La arquitectura destaca las decisiones iniciales relacionadas con el diseño que tendrán un impacto.

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 **(Reynoso, 2004)**.

2.3.1 Modelo Vista Controlador (MVC)

Modelo Vista Controlador es un patrón de diseño de arquitectura de software usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralela e independiente. MVC sugiere la separación del software en 3 capas **(Alejandro Rivera López, 2008)**: Modelo, Vista y Controlador, los cuales serán explicados a continuación.

Capítulo 2 Herramientas y Tecnologías Actuales

Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario o a la aplicación misma.

Vista: Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación Web, la “Vista” es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones.

Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario.

2.4 Lenguajes de Programación

La Real Academia Española define el término lenguaje como el “conjunto de sonidos articulados con que el hombre manifiesta lo que piensa o siente” o al “conjunto de señales que dan a entender algo” y se define para el ámbito informático como el “conjunto de signos y reglas que permite la comunicación con un ordenador.” (RAE).

Los lenguajes de programación son herramientas que permiten crear programas y software. Facilitan las tareas para el desarrollo de una aplicación informática, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de computador a utilizar.

Un lenguaje de programación permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, tal como sucede con el lenguaje Léxico.

2.4.1 Lenguaje Python

Capítulo 2 Herramientas y Tecnologías Actuales

Es un lenguaje de programación dinámico y orientado a objetos que puede ser usado de muchas maneras en el desarrollo de software. Ofrece gran soporte e integración con otros lenguajes y herramientas, viene con una extensiva cantidad de librerías.

Se puede ejecutar en Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, y teléfonos celulares Nokia. Python también ha sido portado para las máquinas virtuales de Java y .NET.

Es distribuido bajo la licencia Open Source OSI que lo hace libre para ser usado inclusive en el desarrollo de productos comerciales.

¿Por qué usar Python?

Entre las potencialidades que brinda se tienen:

- Portabilidad.
- Versatilidad.
- Simplicidad.
- Interactividad.
- Sintaxis clara y legible.
- Productividad.
- Popularidad.
- Facilidad y rapidez de aprendizaje.
- Open Source

2.5 Plugins para GeoQ utilizando el lenguaje de programación Python

Un plugin es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande, es decir, incrementa o aumenta funcionalidades de un programa principal.

Los plugins de Quantum GIS son directorios con una estructura determinada, deben tener un archivo `__init__.py` que sirva para reconocer el plugin y para inicializarlo (Sherman, 2011).

¿Cómo construir un plugin para GeoQ utilizando el lenguaje de programación Python?

- ❖ Crear estructura de clases requerida.
- ❖ Crear ícono propio.
- ❖ Modificar la UI del formulario principal.
- ❖ Compilar la UI con pyuic4.

Estructura de directorios

- ❖ `__init__.py`: Es el punto de inicio del plugin. Contiene la información general, versión, nombre y clase principal.
- ❖ `resources.qrc`: El documento *.XML* creado por el QT-Designer. Contiene direcciones de los recursos de las formas, por ejemplo los íconos utilizados.
- ❖ `resources.py`: Contiene la misma información de `resources.qrc` pero compilada para Python.
- ❖ `form.ui` Este es el formulario principal creado en QT-Designer.
- ❖ `form.py`: Contiene el código del formulario traducido para Python.
- ❖ `ejemplo.py`: Esta clase no es necesario modificarla ya que es la encargada de iniciar y terminar la ejecución del plugin en sí, agregándolo o quitándolo de la interfaz de GeoQ.
- ❖ `ejemploDialog.py`: Esta clase es la más importante ya que contiene el código del formulario principal y en ella se implementa la interfaz del plugin. Hereda de `form.py`

2.6 Frameworks de desarrollo

La palabra en inglés *framework* se usa para definir una estructura conceptual destinada a servir de apoyo o guía para la construcción de algo útil. *Framework* en español significa estructura, marco, armazón.

En los sistemas informáticos, un *framework* es una estructura en capas que indica qué tipos de programas pueden o deben ser desarrollados y cómo se relacionan entre sí.

Algunos frameworks incluyen programas, especifican las interfaces y ofrecen herramientas de programación.

Un framework puede ser un conjunto de funciones dentro de un sistema y cómo se interrelacionan: las capas de un sistema operativo, las capas de una sub-aplicación, determina estándares de comunicación en determinados niveles de una red, etc. Un framework es generalmente más amplio que un protocolo y más prescriptivo que una estructura.

2.6.1 Eclipse como IDE de desarrollo

Es una plataforma de software de código abierto independiente. Permite editar, diseñar y desarrollar módulos; también dispone de su propio depurador para ir chequeando, en busca de errores, cada uno de los módulos que se van diseñando.

2.7 Metodologías de desarrollo

Las Metodologías de Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software. Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero los de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software. Se podrían clasificar en dos grandes grupos **(Carrillo Pérez, y otros, 2008)**:

Las metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Estas metodologías son llamadas Metodologías Pesadas.

Las metodologías orientadas al intercambio directo con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Estas son llamadas Metodologías ligeras/ágiles. **(Carrillo Pérez, 2008)**.

Capítulo 2 Herramientas y Tecnologías Actuales

Según la norma 1074 de IEEE toda metodología de desarrollo de software debe incluir la forma en que se va a realizar la captura de requisitos, el diseño, la implementación y prueba.

2.7.1 Metodologías ágiles

Este tipo de metodología nace en febrero del 2001 en una reunión celebrada en Utah, EEUU. Entre las principales ideas que aborda la metodología ágil se encuentran: **(Carrillo Pérez, 2008)**

- Se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados.
- Se hace mucho más importante crear un producto software que funcione que escribir mucha documentación.
- El cliente está en todo momento colaborando en el proyecto.
- Es más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan.

2.7.1.1 RUP Ágil (AUP)

RUP Ágil describe un enfoque simple y fácil de entender para el desarrollo de software usando técnicas y conceptos que aún se mantienen vigentes en RUP. La disciplina de Modelado abarca las disciplinas de Modelado del Negocio, de Requisitos y de Análisis y Diseño de RUP.

El modelado es una parte importante en AUP, pero no domina el proceso, puede seguir ágil creando modelos y documentos los cuales son apenas lo suficientemente bueno.

Las disciplinas de la Administración de la Configuración y Cambios ahora son la disciplina de la Administración de la Configuración. En el desarrollo ágil las actividades de administración de cambios son típicamente parte del esfuerzo de la administración de requisitos, la cual es parte de la disciplina de Modelado.

La naturaleza serial en RUP Ágil es capturada en cuatro fases:

Capítulo 2 Herramientas y Tecnologías Actuales

Iniciación: El objetivo es identificar el alcance inicial del proyecto, una arquitectura potencial de su sistema, y obtener la financiación inicial del proyecto y la aceptación del involucrado.

Elaboración: El objetivo es mejorar la arquitectura del sistema.

Construcción: El objetivo es construir software funcional en una base regular e incremental, la cual cumpla con las necesidades de prioridad más alta de los involucrados de su proyecto.

Transición: El objetivo es validar y desplegar su sistema en su ambiente de producción.

Las disciplinas son ejecutadas en una manera iterativa, definiendo las actividades las cuales los miembros del equipo ejecutan para construir, validar y liberar software funcional que cumpla con las necesidades de sus involucrados. Las disciplinas son:

Modelado: El objetivo es entender el negocio de la organización, el problema de dominio que se aborda en el proyecto e identificar las soluciones viables para manejar el dominio del problema.

Implementación: El objetivo es transformar su modelo en código ejecutable y llevar a cabo un nivel básico de las pruebas, en particular, la unidad de prueba.

Pruebas: El objetivo es ejecutar una objetiva evaluación para asegurar la calidad. Esto incluye la detección de defectos, validaciones de que el sistema funciona como fue diseñado, y verificar que se cumplan los requisitos.

Despliegue: El objetivo de ésta disciplina es planificar la entrega del proyecto de desarrollo y ejecutar el plan, para dejar disponible el sistema al usuario final.

2.8 El lenguaje de Modelado (UML)

UML es un lenguaje estándar para crear planos de software, no es un lenguaje de programación; sin embargo permite hacer una rápida transición del modelo al código.

Capítulo 2 Herramientas y Tecnologías Actuales

La necesidad de diseños sólidos ha traído consigo la creación de una notación de diseño que los analistas, desarrolladores y clientes acepten con pauta (tal como la notación en los diagramas esquemáticos sirve como pauta para los trabajadores especializados en electrónica); el UML es esa misma notación.

Es una notación (esquemática en su mayor parte) con que se construyen sistemas por medio de conceptos orientados a objetos (**LARMAN**).

UML puede usarse con varios procesos de desarrollo, fue diseñado para usarse con un proceso iterativo, incremental, guiado por casos de uso y centrado en la arquitectura, el tipo de proceso que se considera más apropiado para el desarrollo de sistema complejos modernos. (**Rumbaugh, y otros**).

UML es el primer método en publicar un meta-modelo en su propia notación, incluyendo la notación para la mayoría de la información de, análisis y diseño. Se trata pues de un meta-modelo auto-referencial, cualquier lenguaje de modelado de propósito general debería ser capaz de modelarse a sí mismo (**Morataya, 2007**).

¿Por qué utilizar UML?

- Porque recoge lo mejor de diferentes sistemas de modelado.
- Porque es adaptable a casi cualquier sistema.
- Porque puede ser utilizado en la mayoría de fases de un proyecto.

2.9 Herramienta CASE

CASE son las siglas, que corresponde a: Computer Aided Software Engineering; y en su traducción al español significa Ingeniería de Software Asistida por Computación.

El concepto de CASE es muy amplio; y una buena definición genérica, que pueda abarcar esa amplitud de conceptos, sería la de considerar a la Ingeniería de Software Asistida por Computación como la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación.

Capítulo 2 Herramientas y Tecnologías Actuales

Las herramientas CASE representan una forma que permite modelar los *Procesos de Negocios* de las empresas y desarrollar los *Sistemas de Información Gerenciales*.

Son como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del *ciclo de vida* de desarrollo de un software (investigación preliminar, análisis, diseño, implementación e instalación).

Objetivos de las Herramientas CASE:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

2.9.1 Visual Paradigm como herramienta CASE.

Visual Paradigm permite representar todo tipo de diagramas UML para las distintas fases por la que transita un software en desarrollo como la captura de requisitos, análisis, diseño e implementación.

Esta poderosa herramienta facilita la generación de código, lo que facilita el trabajo de los desarrolladores y además es disponible en múltiples plataformas. Su notación es muy parecida a la estándar, permite configurar las líneas de redacción, el modelado de

base de datos, el modelado de requisitos, el modelado del proceso de negocio, la interoperabilidad, la generación de documentación y la generación de código base para diferentes lenguajes de programación como Java, C# y PHP además de permitir la integración con herramientas de desarrollo (IDE).

Algunas de las características más relevantes de Visual Paradigm son:

- Obtener un producto de calidad.
- Se puede realizar el modelado de aplicaciones web.
- Disponible en varios idiomas.
- Fácil de instalar y actualizar.
- Compatibles entre ediciones.
- Licencia gratuita y comercial.

¿Por qué utilizar Visual Paradigm?

Visual Paradigm soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, prueba y despliegue, además de ser poderoso y fácil de utilizar. Trae consigo la generación de código, lo que agiliza el trabajo de los implementadores y disponibilidad en múltiples plataformas. Además posee una sencilla integración con un control de versiones como Subversion.

2. 10 Conclusiones

Para la construcción de la herramienta propuesta se emplearán como lenguajes de programación Python siguiendo como metodología de desarrollo RUP Ágil. Como lenguaje de modelado se utilizará UML y finalmente para el modelado, la herramienta Visual Paradigm.

Capítulo 3 Presentación de la solución propuesta

3.1 Introducción.

En el presente capítulo se describe la solución propuesta, se realiza un estudio del negocio y el entorno donde se desenvuelve, se identifican los principales conceptos del dominio y la relaciones entre estos, se identifican los requisitos funcionales y los no funcionales; finalmente se arriba al modelo de Casos de Usos del Sistema para luego ser descrito textualmente.

3.1.1 Modelo del dominio en el Desarrollo del Software.

El modelo del dominio ayuda a comprender los conceptos que utilizan los usuarios, es decir los conceptos con los que trabaja y deberá trabajar la aplicación. Es una representación visual estática del entorno real objeto del proyecto. Se llama dominio para distinguirlo del modelo del negocio. El modelo del dominio se centra en una parte del negocio, la relacionada con el ámbito del proyecto.

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases de software ni objetos de software con responsabilidades, sino más bien representa las clases conceptuales u objetos del mundo real en un dominio de interés.

El modelo de dominio se debe concebir como un diccionario visual de abstracciones que será utilizado en fases posteriores y cuya función principal es ayudar a comprender el problema a tratar.

3.2 Entorno donde trabajará el sistema

Al realizarse un estudio profundo del problema al que se pretende brindar una solución, no fue posible identificar con claridad los diferentes procesos del negocio en que se desarrolla el problema existente; de igual modo no se pudo concretar con seguridad cuáles eran los actores implicados en el negocio, los trabajadores y las actividades que éstos últimos realizan. Por tal motivo se le atribuyó mayor importancia a la información que puede brindar un modelo de dominio del problema existente. Se

muestra de manera clara y precisa el modelo del dominio correspondiente al entorno dentro del cual permanecerá el sistema teniendo en cuenta el proceso de visualización correspondiente al SIG GeoQ, así como los conceptos asociados, actividades y reglas del negocio.

3.2.1 Modelo del dominio

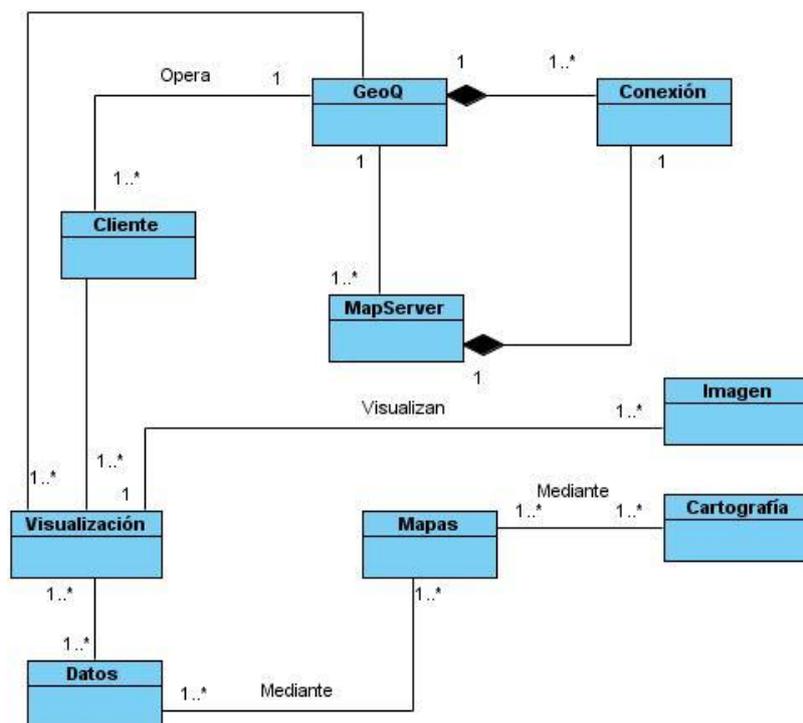


FIGURA 1 MODELO DEL DOMINIO.

3.2.3 Glosario de Términos del Dominio

Cliente: Persona que trabaja en una empresa, organismo u organización que necesite utilizar GeoQ con principal fuente de información MapServer.

Visualización: Es una disciplina transversal que utiliza el inmenso poder de comunicación de las imágenes para explicar de manera comprensible las relaciones

Capítulo 3 Presentación de la solución propuesta

de significado, causa y dependencia que se pueden encontrar entre las grandes masas abstractas de información que generan los procesos científicos y sociales.

Mapa: Es una representación gráfica y métrica de una porción de territorio sobre una 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.

Cartografía: Constituye un conjunto de operaciones que permiten a partir de observaciones y mediciones, la representación de una parte o la totalidad de la Tierra.

MapServer: Es un entorno de desarrollo en código abierto (Open Source Initiative) para la creación de aplicaciones SIG en Internet/Intranet con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Internet MapServer (IMS). Su funcionamiento básico está configurado en un fichero de texto, que generalmente tiene la extensión ".map" (Fawcett & Butler).

GeoQ: SIG de escritorio basado en Quantum GIS.

Conexión: Se refiere al vínculo o integración que debe de existir entre GeoQ y MapServer.

Datos: Descripción elemental o una información.

Imagen: Forma de representación de la realidad, entendida no sólo como algo físico sino también como un pensamiento.

3.3 Requisitos Funcionales

De acuerdo con el Proceso Unificado de Rational ("RATIONAL Unified Process" - RUP), el término requisito puede definirse como una condición que el sistema debe cumplir o capacidad que debe tener. Para que un requisito se considere satisfactorio debe cumplir las siguientes características: complementos, implementación

Capítulo 3 Presentación de la solución propuesta

independiente, consistente y no ambiguo, preciso, verificable, que pueda ser leído y modificable.

Mientras más grande e intrincado sea el sistema en desarrollo, más tipos de requisitos aparecen cuando se inicia la recolección de estos. Mediante la identificación de los tipos de requisitos, los equipos de desarrollo de software pueden separar grandes cantidades de requisitos en grupos que faciliten su manejo, también se logra una comunicación más clara entre los miembros del equipo, y en general se mejora el manejo del proyecto en su totalidad.

Existen dos grandes categorías en las que pueden clasificarse los requisitos, estas son: requisitos funcionales y requisitos no funcionales.

3.3.1 Estrategia de captura de requisitos.

Para la captura de requisitos se pretende utilizar como método de apoyo para su análisis *la lluvia de ideas*. Esta estrategia se basa en expresar ideas acerca del problema y su posible solución donde ningún criterio deberá ser criticado. Además se empleará *el análisis de discurso* como una alternativa complementaria del método anteriormente planteado; para ello se establecerán un conjunto de conversaciones en tiempo real, donde las preguntas y respuestas deben hacerse de manera rápida incluyendo los puntos de vistas sociales.

3.4 Requisitos Funcionales (RF)

R1: Añadir capas MapServer: Poder cargar una o varias capas desde el servidor MapServer y añadirlas a GeoQ para su visualización.

R1.1: Establecer una conexión con el servidor MapServer, ya sea de forma local o remota.

R1.2: Luego de haber establecido una conexión con MapServer es necesario explorar las capas que brinda para lograr su visualización en GeoQ.

R2: Mostrar Capa MapServer: Encargado de mostrar la capa MapServer que se quiera visualizar.

Capítulo 3 Presentación de la solución propuesta

R3: Realizar paneo a la capa MapServer: Encargado de enfatizar la región de la capa con la cual se quiere trabajar.

R4: Realizar zoom a la capa MapServer: Brindar la posibilidad de poder acerca o alejar la vista del mapa con la que se trabaja

R4.1: Realizar zoom+ a la capa MapServer.

R4.2: Realizar zoom- a la capa MapServer.

R4.3: Realizar zoom general a la capa MapServer.

R5: Añadir transparencia a la capa MapServer: Encargado de cambiar la nitidez a la capa una vez que haya sido seleccionada.

R6: Desactivar capa MapServer: Encargado de mostrar sólo las capas que no han sido desactivadas o activas.

R7: Gestionar conexión con MapServer: Incluye las siguientes operaciones:

R7.1 Adicionar conexión

R7.2 Eliminar conexión.

R7.3 Editar conexión.

R8: Listar ficheros mapfile: Listar los ficheros mapfile con los cuales se ha trabajado.

3.5 Requisitos No Funcionales (RNF)

Los RNF son propiedades o cualidades que el producto debe tener. Normalmente están vinculados a requisitos funcionales, es decir una vez 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 (**PRESSMAN, 2005**).

3.5.1 Usabilidad

3.5.1.1 El sistema debe poder ser utilizado por cualquier persona con conocimiento básico en SIG y MapServer.

3.5.1.2 La información deberá estar disponible en todo momento, limitada solamente por las restricciones establecidas como parte de las políticas de seguridad establecidas en el sistema.

3.5.1.3 La interfaz debe ser amigable y entendible, fácil de acceder y manipular.

3.5.2 Apariencia o interfaz externa

3.5.2.1 El sistema debe tener una apariencia fácil de utilizar, que guíe al usuario paso a paso hasta alcanzar el objetivo deseado.

3.5.3 Requisitos de portabilidad y operatividad.

3.5.3.1 La aplicación deberá poder ejecutarse tanto en plataformas UNIX/Linux, como en plataformas Windows 2000 o superior.

3.5.4 Requisitos de Software

3.5.4.1 Las computadoras que utilizarán el software deberán tener instalado:

- ❖ Cualquier distribución de UNIX/Linux, así como el sistema operativo Windows 2000 o una versión superior.
- ❖ SIG GeoQ en su versión 1.5
- ❖ MapServer en su versión 4.0 o superior.

3.5.4.2 Se deberán incluir:

- ❖ Las librerías: libmap.dll, pdflib.dll, oci.dll, libfcgi.dll.
- ❖ El binding de QT Creator con Python (PyQt).

3.5.5 Requisitos de Software de Hardware

- El servidor deberá contar con 1Gb de RAM y con un espacio libre de disco duro 1GB adicional luego de haberse instalado el sistema (aproximadamente 10GB en total).
- Las PC Cliente deberán contar con 512 Mb de RAM y con un espacio libre de 1GB luego de haberse instalado el sistema (aproximadamente 15GB en total).

3.5 Descripción del sistema propuesto

Capítulo 3 Presentación de la solución propuesta

Los casos de uso no son artefactos orientados a objetos -son simplemente historias escritas-. Sin embargo, son una herramienta muy popular en análisis de requisitos y son una parte importante del Proceso Unificado **(Larman)**.

Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema que se está construyendo de la misma forma **(Ceria)**.

	<p>Interviene en el proceso de:</p> <p>Añadir Capa MapServer, Mostrar Capa MapServer, Realizar paneo a la capa MapServer, Realizar zoom a la capa MapServer, Añadir transparencia a la capa MapServer, Desactivar capa MapServer, Listar ficheros mapfile, Gestionar Conexión.</p>
---	--

TABLA 1 DESCRPCIÓN DE LOS ACTORES DEL SISTEMA.

3.6.2 Diagrama de Caso de uso del sistema

A continuación se muestra el diagrama de caso de uso del sistema donde se representan los actores y su relación con el sistema:

Capítulo 3 Presentación de la solución propuesta

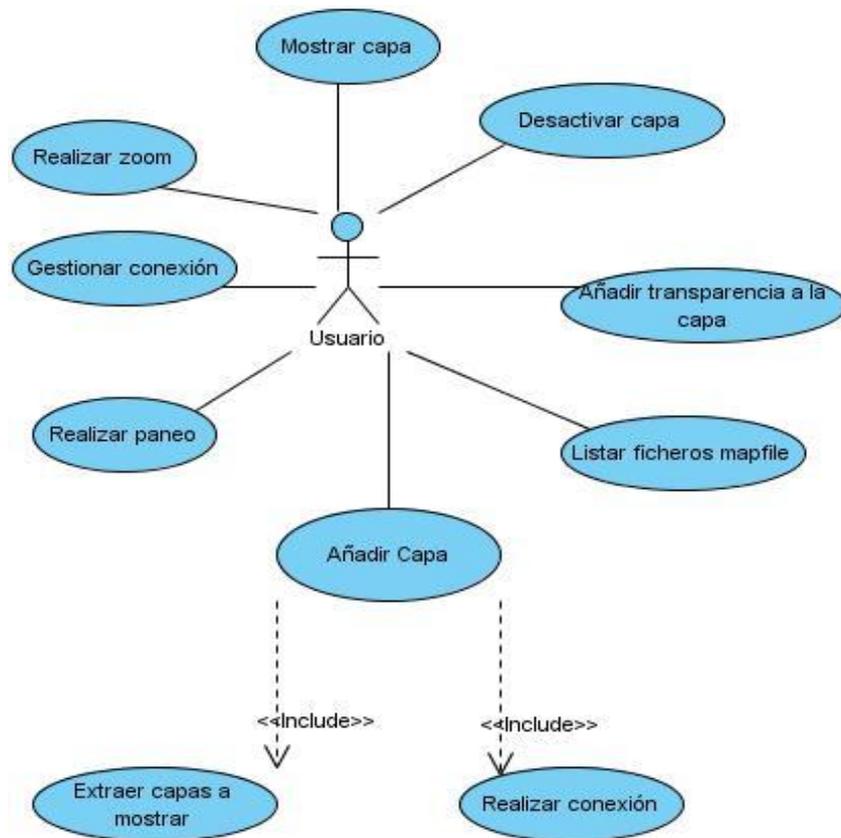


FIGURA 2 DIAGRAMA DE CU DEL SISTEMA.

3.6.3 Descripción textual de los casos de uso del Sistema (Ver Anexo 2).

Caso de Uso:	Gestionar conexión con MapServer
Actores:	Usuario
Resumen:	El caso de uso comienza cuando el usuario necesita gestionar la conexión con MapServer. El caso de uso termina cuando el usuario completa las operaciones a realizar (adicionar, editar, eliminar) con la conexión.
Precondiciones:	Debe estar creada la conexión.
Referencias	R7,R7.1,R7.2,R7.3
Prioridad	Crítico

Capítulo 3 Presentación de la solución propuesta

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El usuario selecciona la opción Gestionar Conexión con MapServer.	2- El sistema muestra todas las conexiones existentes.
3- El usuario selecciona la conexión deseada.	4- El sistema muestra una ventana con la dirección del fichero mapfile abierto, así como todas las capas de la conexión seleccionada.
Sección "Adicionar Conexión"	
Acción del Actor	Respuesta del Sistema
1- El usuario selecciona la opción "Adicionar".	2- El sistema muestra una ventana para introducir los datos de la conexión a adicionar: nombre y dirección del fichero mapfile.
3- El usuario introduce los datos y da clic en el botón "Agregar".	4- El sistema agrega la conexión y cierra la ventana.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.1 El usuario introduce los datos incorrectos: <ol style="list-style-type: none"> 1. En caso que la conexión ya existe. El sistema muestra un ícono de error indicando el mismo y desactiva el botón "Agregar". 2. En caso de que la dirección del fichero mapfile no sea correcta, se muestra un mensaje de error.
Sección "Editar Conexión"	
Acción del Actor	Respuesta del Sistema

Capítulo 3 Presentación de la solución propuesta

1- El usuario selecciona la opción "Editar".	2- El sistema muestra una nueva ventana con la conexión seleccionada.
3- El usuario realiza los cambios deseados a la conexión seleccionada.	4- El sistema realiza los cambios y guarda la conexión.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1- El usuario introduce los datos incorrectos:	4.1- En caso que la conexión ya existe. El sistema muestra un ícono de error indicando el mismo y desactiva el botón "Agregar". 4.2- En caso de que la dirección del fichero mapfile no sea correcta, se muestra un mensaje de error.
Sección "Eliminar Conexión"	
Acción del Actor	Respuesta del Sistema
1- El usuario selecciona la conexión y elige la opción "Eliminar".	2- El sistema muestra un mensaje para verificar que se desea realizar la operación.
3- El usuario selecciona "Aceptar".	4- El sistema elimina la conexión.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1- El usuario selecciona la opción "Cancelar"	4.1- El sistema regresa a la ventana anterior.
Poscondiciones	La información referente a la conexión ha sido gestionada (adicionada, editada, eliminada)
Prototipo de Interfaz	

Capítulo 3 Presentación de la solución propuesta

TABLA 2 DESCRIPCIÓN DEL CASO DE USO GESTIONAR CONEXIÓN CON MAPSERVER.

3.7 Conclusiones

En el presente capítulo se realizó el modelo del dominio, el cual permitió comprender mejor el funcionamiento del sistema en términos de conceptos y las relaciones entre ellos. Además se definieron los actores y casos de uso del sistema, estos últimos fueron posteriormente descritos. Se realizó el diagrama de caso de uso del sistema.

Capítulo 4 Construcción de la solución propuesta

4.1 Introducción

En el presente capítulo se plantea la construcción propuesta en el capítulo anterior, en función de diagramas de clases y estándares de diseño, generalidades de la implementación, diseño de la base de datos, modelo de despliegue y modelo de implementación, finalizando con los diseños de casos de pruebas que validen la implementación.

4.2 Principios de Diseño

Los principios de diseño son una secuencia de pasos, no es una receta pues intervienen: la creatividad, experiencia y un compromiso con la calidad donde existen factores internos (son buscados por el Ingeniero de Software) y externos (propiedades que pueden ser observadas por los usuarios). **(Roger S. Pressman, 2002).**

En el libro Ingeniería de Software Un enfoque práctico de Pressman, Alan Davis define el principio de diseño como el proceso:

- ❖ Donde deben tomarse enfoques alternativos.
- ❖ Que deberá rastrearse hasta el análisis.
- ❖ Que se debe reutilizar.
- ❖ Que debe tratar de imitar el dominio del problema.
- ❖ Busca uniformidad e integración.
- ❖ Que deberá estructurarse para admitir cambios.
- ❖ Para prevenir la adaptación a circunstancias inusuales.
- ❖ Que no debe codificar.
- ❖ Que debe ser evaluado en función de calidad mientras está creciendo.
- ❖ Que minimice errores conceptuales.

Como principio de diseño del sistema se tienen:

- ❖ Una apariencia fácil de utilizar de manera que pueda ser usado por personas con conocimiento básico en SIG y MapServer.

Capítulo 4 Construcción de la solución propuesta

- ❖ Una iconografía descriptiva, es decir, que cada ícono se ajuste a la funcionalidad que represente.
- ❖ Mostrar al usuario un mensaje de confirmación de manera que pueda asegurarse que la operación realizada es correcta.
- ❖ Los nombres de los atributos comenzarán con letra minúscula y los métodos con letra mayúscula.

4.3 Diagrama de clase del diseño

El diagrama de clase es la representación principal de diseño y análisis para un sistema. En él, la estructura de clases se especifica a través de relaciones entre ellas y las distintas estructuras de herencia que se establecen. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal y, en el diseño, se parte del mismo diagrama y se modifica para satisfacer los detalles de la implementación.

A continuación se muestran los diagramas de clases del diseño por caso de uso del sistema (Ver Anexo 3), así como los diagramas de secuencia del diseño (Ver Anexo 4):

Capítulo 4 Construcción de la solución propuesta

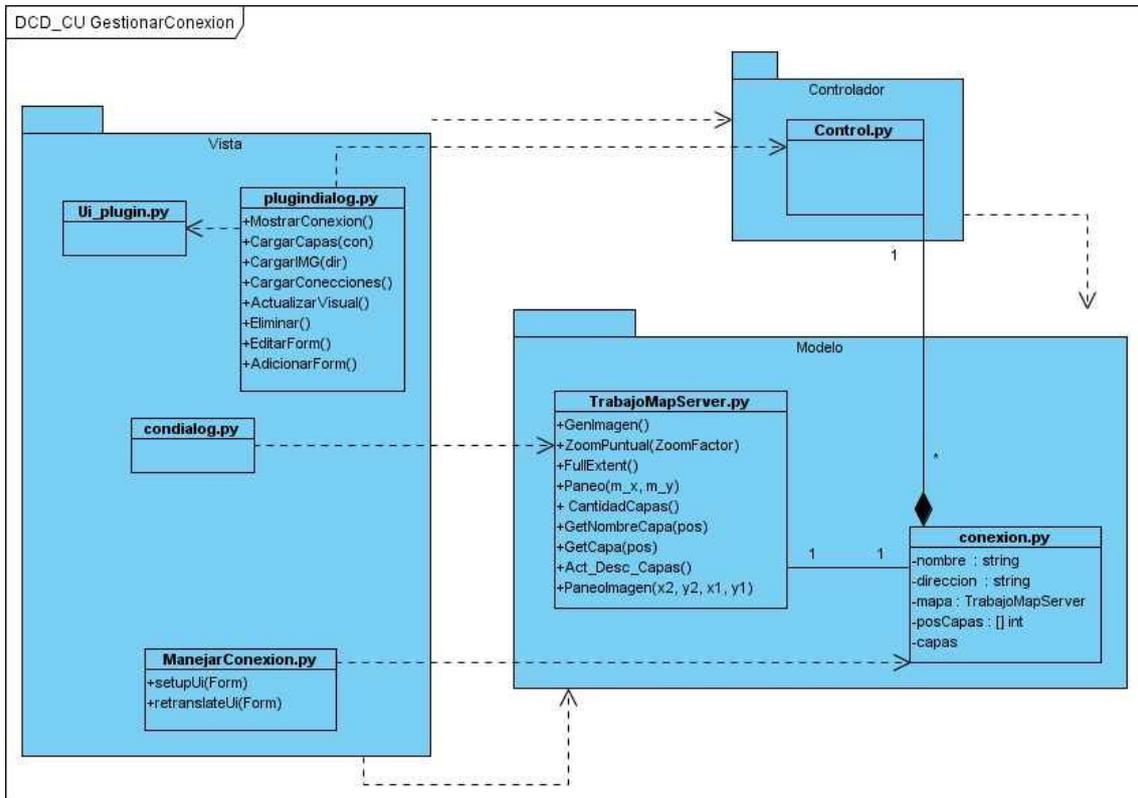


FIGURA 3 DIAGRAMA DE CLASE DEL DISEÑO CU GESTIONAR CONEXIÓN.

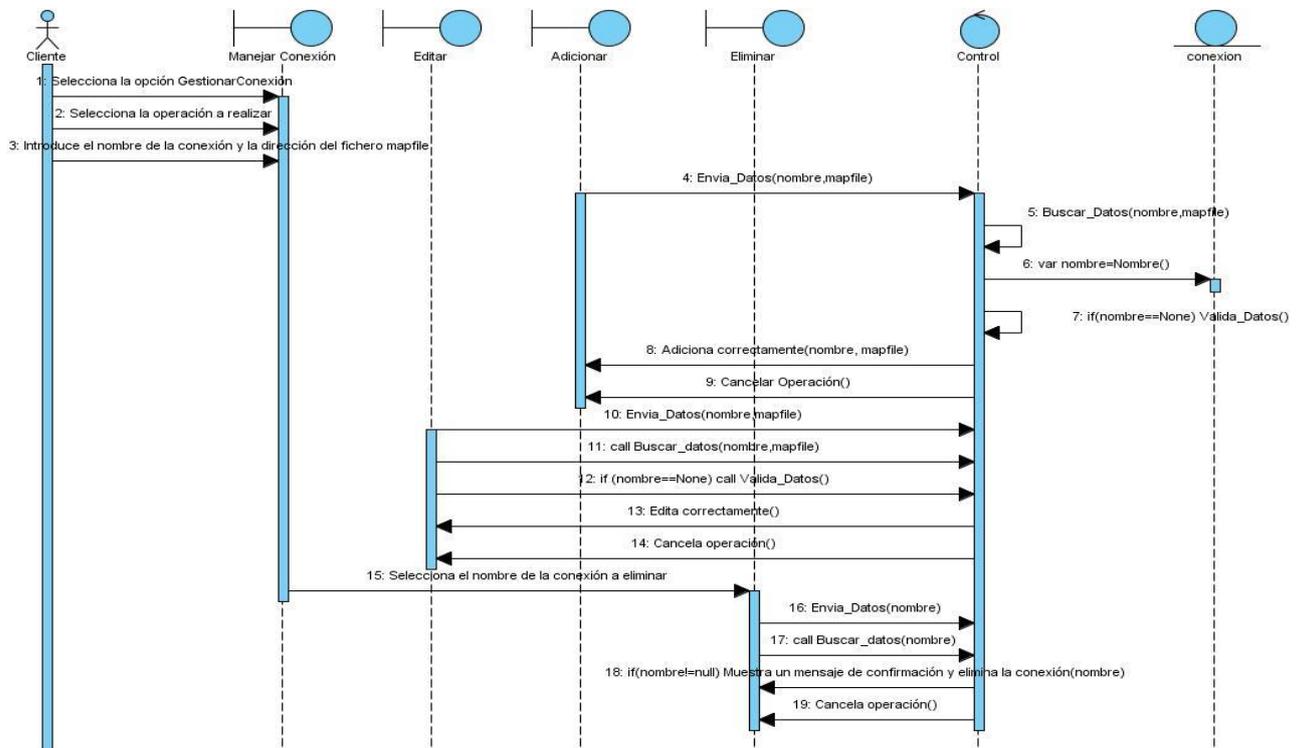


FIGURA 4 DIAGRAMA DE SECUENCIA CU GESTIONAR CONEXIÓN.

4.4 Generalidades de la implementación

Al concluir la fase de Construcción, se terminan de analizar y diseñador todos los Casos de Usos del sistema, refinando de forma general los artefactos construidos en los flujos de Análisis y Diseño.

El modelo de implementación consiste en una visión general de lo que tiene que ser implementado, y un apartado para cada iteración con sus respectivos componentes y subsistemas, así como de los resultados del software que se han de obtener y el testeo que se ha de realizar sobre ellos (Colectivo de Autores ISW II); dicho artefacto está compuesto por el diagrama de despliegue y el diagrama de componente.

4.4.1 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 de cómputo, es decir establece una correspondencia entre las arquitecturas de software y la del sistema. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar. Los nodos poseen relaciones que representan medios de comunicación entre ellos.

A continuación se muestra el modelo de despliegue del sistema, el cual está compuesto por dos nodos de procesamiento (PC _Cliente y Servidor) que estarán vinculados a través del protocolo de comunicación TCP/IP.

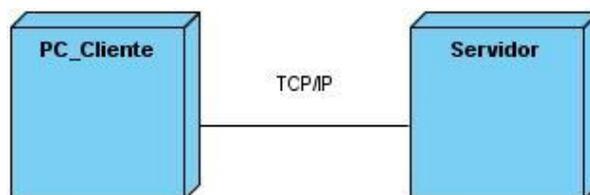


FIGURA 5 DIAGRAMA DE DESPLIEGUE.

4.4.2 Modelo de Componente

Un componente es un elemento de funcionalidad del sistema reutilizable que proporciona y utiliza el comportamiento a través de las interfaces. Los diagramas de componentes proveen una vista arquitectónica de alto nivel del sistema; posibilitando visualizar el camino para la implementación.

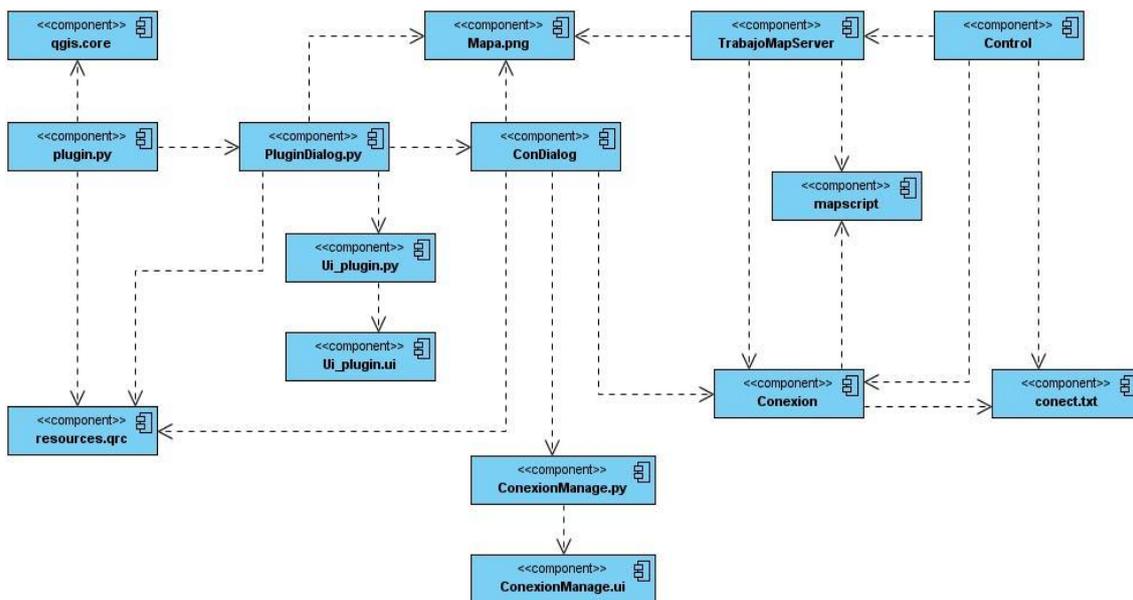


FIGURA 6 DIAGRAMA MODELO DE COMPONENTES.

4.5 Prueba del sistema propuesto

El proceso de desarrollo de software no está exento de errores; es por eso que es necesaria la realización de pruebas, también conocidas como test. Estas pruebas son actividades en las cuales un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas; los resultados son observados y registrados para ser evaluados. El éxito de una prueba es el descubrimiento de un error; por tal motivo esto contribuye a mejorar la calidad del software.

Entre los diferentes tipos de pruebas se pueden destacar:

Capítulo 4 Construcción de la solución propuesta

Pruebas de Validación: Son las pruebas realizadas sobre un software completamente integrado para evaluar el cumplimiento con los especificados (Ulibarri, 2004).

Pruebas de Sistema: El software ya validado se integra con el resto del sistema donde algunos tipos de pruebas a considerar son: rendimiento, robustez, seguridad, usabilidad, instalación, entre otras (Ulibarri, 2004).

Pruebas de Caja Blanca: Se detectan determinados errores que con otros tipos de pruebas resultaría muy difícil posibilitando que se genere un código de mayor calidad que cumpla con los estándares (Ulibarri, 2004).

Pruebas de Aceptación: Son las pruebas que realiza el cliente, determinando que el sistema cumple o no con lo especificado en los funcionales y no funcionales (Ulibarri, 2004).

Pruebas de Caja Negra: Permiten obtener conjuntos de condiciones de entrada que ejecuten todos los funcionales de un programa. Este tipo de prueba no debe interpretarse como una alternativa a las pruebas de caja blanca sino que debe interpretarse como un enfoque diferente (Ulibarri, 2004).

4.5.1 Prueba de Caja Negra

Para validar los componentes del sistema y asegurar el cumplimiento de las funcionalidades del mismo se decide realizar un conjunto de pruebas de caja negra, las cuales se centran en lo que se espera de un módulo, es decir intentan encontrar casos en donde no se cumplen determinadas especificaciones; por ello se denominan también pruebas funcionales donde el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo realmente el módulo por dentro. Las herramientas básicas para realizarlas son: observar la funcionalidad y contrastar con la especificación.

Los casos de pruebas de la caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.

Capítulo 4 Construcción de la solución propuesta

- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Tipos de errores que pretende encontrar:

- Funciones incorrectas o ausentes.
- Errores en la interfaz.
- Errores rendimiento.
- Errores de inicialización y terminación.
- Errores en estructuras de datos o en acceso a base de datos externas.

Clasificación de las técnicas de caja negra:

A continuación se detallan las técnicas de pruebas de caja negra existentes:

Prueba de análisis de valores límites: Es una técnica de diseño de casos de prueba que completa a la partición equivalente. No selecciona cualquier elemento de una clase de equivalencia, sino lleva a la elección de casos de pruebas en sus extremos. Además no se centra solamente en las condiciones de entrada, también analiza los campos de salida (**Prado, 2007**).

Como un ejemplo típico de pruebas inducidas por los datos se puede mencionar la comprobación de valores límites, la verificación de la integridad de los datos en la base de datos, las situaciones de excepciones que puedan ocurrir y el rendimiento del sistema cuando cada características del mismo se explota al límite de su capacidad. Tiene una limitación: es prácticamente imposible reproducir todo el espectro por la innumerable cantidad de combinaciones de entradas posibles, agravadas por el desconocimiento de la lógica interna (**Prado, 2007**).

Para la validación del sistema se seleccionó *la técnica partición equivalente*. Este método de prueba de caja negra divide el dominio de entrada de un programa en clases de datos, a partir de las cuales deriva los casos de prueba donde cada una de estas clases de equivalencia representa a un conjunto de estados válidos o inválidos para las condiciones de entrada. (**Prado, 2007**)

Capítulo 4 Construcción de la solución propuesta

A continuación se muestra la técnica partición equivalente aplicada a los Casos de Uso (Ver Anexo 5).

4.5.2 Caso de Uso Desactivar capa MapServer

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
Desactivar capa MapServer.	EC1.1: Desactivar capa correctamente.	Esta funcionalidad permite mostrar sólo las capas que no han sido desactivadas o activas.	<ul style="list-style-type: none"> El sistema muestra una ventana con las capas existentes. El usuario selecciona la capa que desea desactivar. El sistema desactiva las capas seleccionadas.
	EC 1.2: No desactivar capa por no haber sido seleccionada	El sistema no muestra los resultados ya que el usuario no seleccionó una capa.	<ul style="list-style-type: none"> El sistema muestra el mensaje "Debe seleccionar al menos una capa".

TABLA 3: SECCIONES A PROBAR CU DESACTIVAR CAPA

Descripción de las variables

No	Nom_Capa	Clasificación	Valor Nulo	Descripción
1	Nombre de la capa	Campo de selección.	No	
2	Nombre de la conexión	Campo de texto.	No	

ID del escenario	Escenario	Nom_Capa	Nom_conexión	Respuesta del Sistema	Resultado de la Prueba

Capítulo 4 Construcción de la solución propuesta

ID del escenario	Escenario	Nom_Capa	Nom_conexión	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Desactivar capa correctamente.	V/ "Viales"	V/ "Ejemplo"	Se desactivan las capas seleccionadas.	Satisfactorio.
EC 1.2	No desactivar capa por no haber sido seleccionada.	I/ " "	V/ "Ejemplo"	Muestra un mensaje de error informando que se selecciona al menos una capa.	Satisfactorio.

TABLA 4: MATRIZ DE DATOS CU DESACTIVAR CAPA MAPSERVER

4.6 Conclusiones

En este capítulo se desarrollaron los principales artefactos para llevar a cabo el proceso de implementación de vital importancia en la construcción del sistema. Mediante la realización del Modelo de Despliegue, se describió cómo está distribuida física y lógicamente la arquitectura del sistema. La realización del Modelo de Implementación, permitió detallar los componentes creados para el desarrollo de la aplicación y la relación entre ellos. Se llevó a cabo la elaboración de los diseños de pruebas de Caja Negra, con el que se pretende probar cada uno de los elementos que componen la interfaz, permitiendo así validar la solución propuesta.

Después del total cumplimiento de las tareas trazadas durante la investigación se arribó a un conjunto de conclusiones que a continuación se relacionan:

- ❖ La solución propuesta constituye la primera aplicación capaz de integrar el SIG de escritorio GeoQ con el servidor de mapas MapServer.
- ❖ Cada uno de los requisitos funcionales y no funcionales fueron debidamente implementados.
- ❖ Las herramientas y tecnologías utilizadas para la construcción de la solución propuesta son libres y multiplataformas.
- ❖ Con la solución propuesta es posible obtener la misma visualización que en la Plataforma GENESIG.
- ❖ El diseño de las pruebas de cajas negras permitió la validación del sistema propuesto para verificar el cumplimiento de los requisitos funcionales capturados.

Los autores de la presente investigación recomiendan:

- ❖ Continuar diseñando funcionalidades que exploten al máximo la vinculación del SIG GeoQ con el servidor de mapas MapServer.
- ❖ Trabajar en mejoras en cuanto al diseño del sistema, teniendo en cuenta la opinión de las personas expertas en el tema.
- ❖ Realizarles otros tipos de pruebas al software construido.

BIBLIOGRAFÍA CITADA

- ❖ AGUINIS, MARCOS. LA GESTA DEL MARRANO.
- ❖ alemanes, Colectivo de autores. 1966. Didáctica 2da. Parte. . 1966.
- ❖ Cabanes, Nacho. Diccionario básico de la informática.
- ❖ Carrillo Pérez, Isaías, Pérez González, Rodrigo and Rodríguez Martín, Aureliano David. 2008. Metodologías de Desarrollo del Software. 2008.
- ❖ Ceria, Santiago. Casos de Uso: Un Método Práctico para Explorar Requisitos.
- ❖ Corporation, Rational Software. 1995. Rational RequisitePro. 1995.
- ❖ Cubanos, Colectivo de autores. 1984. 1984.
- ❖ Davis, Roy Eugene. El libro del la Vida .
- ❖ Larman, Craig. Applying UML and Patterns.
- ❖ 2008. Metodología de Desarrollo de software. 2008.
- ❖ PRESSMAN. 2005. PRESSMAN. 2005.
- ❖ Rumbaugh, James, Jacobson, Ivar and Booch, Grady. El lenguaje unificado de modelado.
- ❖ Sanz Núñez, Alfonso C. 1992. El Atlas Nacional de España. 1992.
- ❖ Zayas, Carlos Álvarez de. 1988. Fundamentos teóricos de la dirección del proceso de formación del profesional de perfil amplio. 1988.

BIBLIOGRAFÍA CONSULTADA

- ❖ Online] [Cited: marzo 16, 2011.]
http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/Pruebas.pdf.
- ❖ dsi.uclm. [Online] [Cited: marzo 12, 2011.]
<http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
- ❖ eisc.univalle. [Online]
http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/Pruebas.pdf.
- ❖ es.scribd. es.scribd. [Online] [Cited: marzo 12, 2011.]
<http://es.scribd.com/doc/49708406/19/Modelo-de-despliegue>.
- ❖ <http://eva.uci.cu>. eva. [Online] [Cited: abril 6, 2011.] <http://eva.uci.cu>.
- ❖ ibiblio. [Online] [Cited: marzo 14, 2011.]
<http://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-modelado-sistemasUML/multiple-html/x219.html>.
- ❖ indalog. [Online] [Cited: marzo 18, 2011.]
<http://indalog.ual.es/mtorres/LP/Prueba.pdf>.
- ❖ ingenierogestion. [Online] [Cited: marzo 18, 2011.]
<http://ingenierogestion.blogspot.com/2009/06/pruebas-de-caja-negra-y-caja-blanca.html>.
- ❖ lad.dit. [Online] [Cited: marzo 18, 2011.]
<http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.
- ❖ Prado, Elena Raja. 2007. Actas de Talleres de Ingeniería del Software. 2007. 1.
- ❖ 2007. sistedes. [Online] 2007. <http://www.sistedes.es/TJISBD/Vol-1/No-4/articulos/pris-07-raja-ctps.pdf>.
- ❖ Reynoso, Carlos Billy. 2004. Introducción a la Arquitectura de Software. 2004.

Bibliografía Consultada

- ❖ Roger S. Pressman. 2002. Ingeniería de Software. Un enfoque práctico. 2002.
- ❖ Ulibarri. 2004. Pruebas del Sistema. 2004.
- ❖ [Online] [Cited: marzo 12, 2011.]
<http://www.uv.mx/personal/asumano/files/2010/07/Principios-de-Dise%C3%B1o.pdf>.
- ❖ <http://hub.qgis.org/projects/plugin-builder>

Glosario de Términos

- D -

Datos ráster

Es una malla regular formada por celdas, o en el caso de imágenes, píxeles. Tienen un número fijo de filas y columnas. Cada celda tiene un valor numérico y cierto tamaño geográfico. (E. Sherman, 2004)

Datos vectoriales

Son datos que pueden ser almacenados de tres formas: puntos, líneas y polígonos. (E. Sherman, 2004)

- C -

Calidad cartográfica

La calidad de una cartografía, entendida en su acepción más amplia, limita la forma en que puede y debe ser usada y analizada la información que soporta. (López, 1998)

Cartografía

Es la ciencia que se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales y demás. También se denomina cartografía a un conjunto de documentos territoriales referidos a un ámbito concreto de estudio. (brice, 2010)

- I -

Imagen rasterizada (también llamada mapa de bits, imagen matricial o bitmap)

Es una estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color, denominada ráster, que se puede visualizar en un monitor de ordenador, papel u otro dispositivo de representación. (E. Sherman, 2004)

- M -

Mapa

Es una representación gráfica y métrica de una porción de territorio generalmente sobre una superficie bidimensional pero que puede ser también esférica como ocurre en los globos terráqueos. Incluye una síntesis de conjunto al igual que un detalle analítico que permita una lectura más profunda. El nivel en que se cumplan estas condiciones será igualmente el nivel de calidad cartográfica. (Florencia, 2009)

- O -

OSGeo

La Fundación para el Código Abierto Geoespacial, u OSgeo, es una organización sin ánimo de lucro cuyo objetivo principal es apoyar y promocionar el desarrollo abierto y colaborativo de los datos y las tecnologías geoespaciales. Proporciona apoyo económico, organizativo y legal a la comunidad del código abierto geoespacial. Es una entidad legal independiente a la que los miembros de la comunidad pueden aportar código fuente, donativos y otros recursos. OSGeo sirve además como una organización de apoyo a la comunidad del código abierto geoespacial, proporcionando un foro e infraestructura compartido para mejorar la colaboración horizontal entre proyectos. (Adair, Birch, Burk, & otros, 2006)

- V -

Vector

Son una vía para describir una localización utilizando un conjunto de coordenadas. Cada coordenada hace referencia a una localización geográfica mediante un sistema de valores **x** e **y**. (E. Sherman, 2004)

Anexo 1 Arquitectura Modelo Vista Controlador

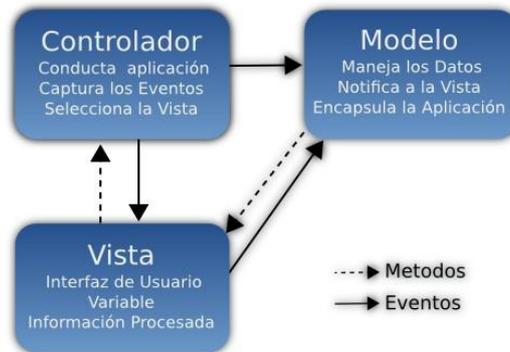


FIGURA 7 PATRÓN MODELO – VISTA – CONTROLADOR.

Anexo 2 Descripción Textual de los casos de uso

Caso de Uso:	Añadir capa MapServer
Actores:	Usuario
Resumen:	El caso de uso inicia cuando el usuario selecciona el botón o la acción de Agregar capa MapServer. El caso de uso finaliza cuando el usuario completa las operaciones (realizar conexión, extraer capas) con la capa MapServer.
Precondiciones:	El nombre de la conexión no debe existir en el sistema.
Referencias	<i>RF 1,RF 1.1,RF 1.2</i>
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El actor selecciona la opción de “Agregar capa MapServer”.	2- El sistema muestra una interfaz que le permitirá al usuario crear una Nueva conexión con el servidor de mapas MapServer.
3- El actor selecciona la opción de “Nueva Capa”.	4- El sistema le muestra una nueva ventana pidiéndole al actor que introduzca los datos para poder realizar la conexión. Entre los datos se encuentran: el nombre de la conexión y la dirección del fichero mapfile.
5- El actor procede a llenar el campo de información para la conexión.	6- El sistema brinda la opción de “Probar conexión”.
7- El actor selecciona la opción de “Realizar conexión”, “Extraer capas” o “Cancelar”, según lo que es deseado.	8- En dependencia a la opción que haya seleccionado el usuario, el sistema ejecuta una operación: 2. En caso de que la opción escogida por el usuario haya sido Realizar conexión: Ver

	<p style="text-align: center;">sección <i>“Realizar conexión con MapServer”</i>.</p> <p>3. En caso de que la opción escogida por el usuario haya sido Extraer capas: Ver sección <i>“Extraer capas de MapServer que se van a mostrar”</i>.</p> <p>4. En caso contrario si el usuario escogió la opción de “Cancelar” el sistema cierra la ventana automáticamente.</p>
--	--

Sección *“Realizar conexión con MapServer”*

Acción del Actor	Respuesta del Sistema
1- El usuario selecciona la opción Realizar conexión.	2- El sistema le muestra una nueva ventana pidiéndole al actor que introduzca el nombre de la conexión y la dirección de un fichero mapfile.
3- El usuario introduce los datos.	4- El sistema establece la conexión y muestra una nueva ventana con todas las capas existentes.

Flujo Alternativo

Acción del Actor	Respuesta del Sistema
	3.1 El usuario no introduce el nombre o la dirección del fichero mapfile. El sistema desactiva el botón “Agregar”.

Sección *“Extraer capas de MapServer que se van a mostrar”*.

Acción del Actor	Respuesta del Sistema
1- El usuario selecciona la conexión a mostrar.	2- El sistema le muestra una vista previa de la conexión seleccionada.
3- El usuario selecciona la opción “Cargar en GeoQ”.	4- El sistema extrae la conexión como una capa hacia GeoQ.

Flujo Alternativo

Acción del Actor		Respuesta del Sistema
Poscondiciones	La información referente a la capa MapServer ha sido añadida.	
Prototipo de Interfaz		

TABLA 5 DESCRIPCIÓN DEL CASO DE USO AÑADIR CAPA MAPSERVER.

Caso de Uso:	Mostrar Capa MapServer.	
Actores:	Usuario	
Resumen:	El caso de uso inicia cuando el usuario necesita mostrar capas MapServer.	
Precondiciones:	Debe existir al menos una capa MapServer. Se debe seleccionar al menos una capa.	
Referencias	RF 2	
Prioridad	Crítico.	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1- El usuario selecciona la capa que desea visualizar.		2- El sistema verifica que existe al menos una capa.
		3- El sistema visualiza la capa seleccionada.
Flujo Alternativo		
Acción del Actor		Respuesta del Sistema
		2.1 El sistema muestra un mensaje de error "Debe seleccionar al menos una capa".

	3.1 El usuario cancela la operación. El sistema cancela automáticamente la operación.
Poscondiciones	Se muestra la capa MapServer con todos sus atributos.
Prototipo de Interfaz	

TABLA 6 DESCRIPCIÓN DEL CASO DE USO MOSTRAR CAPA

Caso de Uso:	Realizar paneo a la capa MapServer.	
Actores:	Usuario	
Resumen:	El caso de uso inicia cuando el usuario necesita hacerle paneo a una región de una de las capas MapServer.	
Precondiciones:		
Referencias	RF 3	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1- El usuario selecciona la opción de Paneo. 2- El usuario arrastra el mapa hacia la región que quiere visualizar.	3- El sistema activa la opción de "Paneo". 4- El sistema desplaza el mapa a la posición seleccionada.	
Flujo Alterno		
Acción del Actor	Respuesta del Sistema	
Poscondiciones	Se muestra la región de la capa a la que se le hizo el paneo.	

Prototipo de Interfaz

TABLA 7 DESCRIPCIÓN DEL CASO DE USO REALIZAR PANEO A LA CAPA MAPSERVER.

Caso de Uso:	Realizar zoom	
Actores:	Usuario	
Resumen:	El caso de uso inicia cuando el usuario selecciona, el botón o la acción de Realizar Zoom y según el zoom seleccionado es que procede a realizar su función.	
Precondiciones:	Para lograr una navegación tiene que tener cargada o creada una capa. Además debe de tener seleccionada la capa para poder realizar las diferentes opciones del Zoom.	
Referencias	RF 4,RF 4.1,RF 4.2, RF 4.3	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1- El usuario selecciona una de las tres operaciones presentes: "Realizar Zoom+", "Realizar Zoom-" y "Realizar Zoom General".	2- Si el usuario escogió Realizar Zoom+, ver sección "Realizar zoom + a la capa MapServer". Si el usuario escogió Realizar Zoom-, ver sección "Realizar zoom - a la capa MapServer". Si el usuario escogió Realizar Zoom General, ver sección "Realizar zoom general a la capa MapServer"	
Sección "Realizar zoom + a la capa MapServer"		

Acción del Actor	Respuesta del Sistema
1- El usuario da clic en el área de trabajo donde se encuentra el mapa en el que se está trabajando y especificando mediante el clic dado, el lugar al que desea visualizar.	2- El sistema captura la posición mediante el clic seleccionado por el usuario, permitiendo llevar hacia adelante el mapa, mostrando una mejor visualización de la posición capturada.
Sección “Realizar zoom - a la capa MapServer”	
Acción del Actor	Respuesta del Sistema
1- El usuario da clic en el área de trabajo donde se encuentra el mapa en el que se está trabajando y especificando mediante el clic dado, el lugar al que desea visualizar.	2- El sistema captura la posición mediante el clic seleccionado por el usuario, permitiendo llevar hacia atrás el mapa, mostrando una menor visualización de la posición capturada.
Poscondiciones	Se observa la capa con el nuevo tamaño (más grande o más pequeña).
Sección “Realizar zoom general a la capa MapServer”	
Acción del Actor	Respuesta del Sistema
1- El usuario selecciona la opción “Zoom General”.	2- El sistema muestra la imagen en su tamaño y posición original.
Poscondiciones	Se observa la capa con el nuevo tamaño (más grande, más pequeña o con su tamaño original).
Prototipo de Interfaz	

TABLA 8 DESCRIPCIÓN DEL CASO DE USO REALIZAR ZOOM.

Caso de Uso:	Añadir transparencia a la capa MapServer.
Actores:	Usuario

Resumen:	El caso de uso inicia cuando el usuario necesita añadir transparencia a una capa MapServer.	
Precondiciones:	Debe existir al menos una capa.	
Referencias	RF 5	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1- El usuario selecciona las propiedades de la capa y selecciona "Transparencia".	2- El sistema muestra una nueva ventana con la transparencia actual con un valor de 0 a 100.	
3- El usuario establece la transparencia deseada.	4- El sistema aplica transparencia a la capa seleccionada.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
Poscondiciones	Se muestra la capa con la nueva transparencia.	
Prototipo de Interfaz		

TABLA 9 DESCRIPCIÓN DEL CASO DE USO AÑADIR TRANSPARENCIA A LA CAPA MAPSERVER.

Caso de Uso:	Desactivar capa MapServer.
Actores:	Usuario
Resumen:	El caso de uso inicia cuando el usuario necesita desactivar capas MapServer.

Precondiciones:	Debe existir al menos una capa. Debe seleccionarse previamente una capa.
Referencias	RF 6
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra una ventana con las capas existentes.
2- El usuario selecciona la capa que desea desactivar.	3- El sistema desactiva las capas seleccionadas.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	1.1 El usuario no selecciona la capa. El sistema muestra el mensaje "Debe seleccionar al menos una capa". 3.1 El usuario cancela la operación. El sistema cierra automáticamente la ventana.
Poscondiciones	Se muestran solamente las capas que no han sido desactivadas.
Prototipo de Interfaz	

TABLA 10 DESCRIPCIÓN DEL CASO DE USO DESACTIVAR CAPA MAPSERVER.

Caso de Uso:	Listar ficheros mapfile.
Actores:	Usuario
Resumen:	El caso de uso inicia cuando el usuario desea establecer nuevas conexiones o editar las existentes; para ello debe seleccionar el fichero mapfile, de una lista de ficheros con tan sólo dar clic sobre uno de ellos.
Precondiciones:	Debe existir al menos un fichero mapfile.
Referencias	RF 8
Prioridad	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El usuario selecciona la opción “Adicionar” o “Editar”.	2- El sistema muestra un listado con los ficheros mapfile existentes.
3- El usuario da clic sobre un fichero mapfile.	4- El sistema muestra de forma automática los datos del mapfile.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Poscondiciones	Se establece una nueva conexión con el mapfile seleccionado o se edita una existente.
Prototipo de Interfaz	
z	

TABLA 11 DESCRIPCIÓN DEL CASO DE USO LISTAR FICHEROS MAPFILE.

ANEXO 3 DIAGRAMA DE CLASES DEL DISEÑO

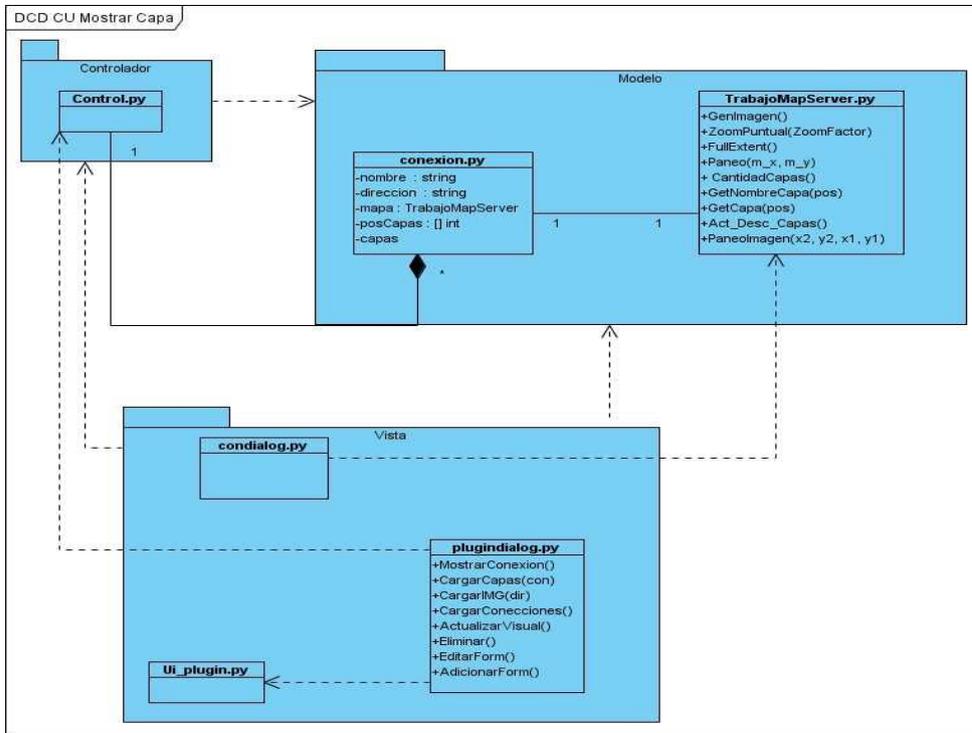


FIGURA 8 DIAGRAMA DE CLASE DEL DISEÑO CU MOSTRAR CAPA.

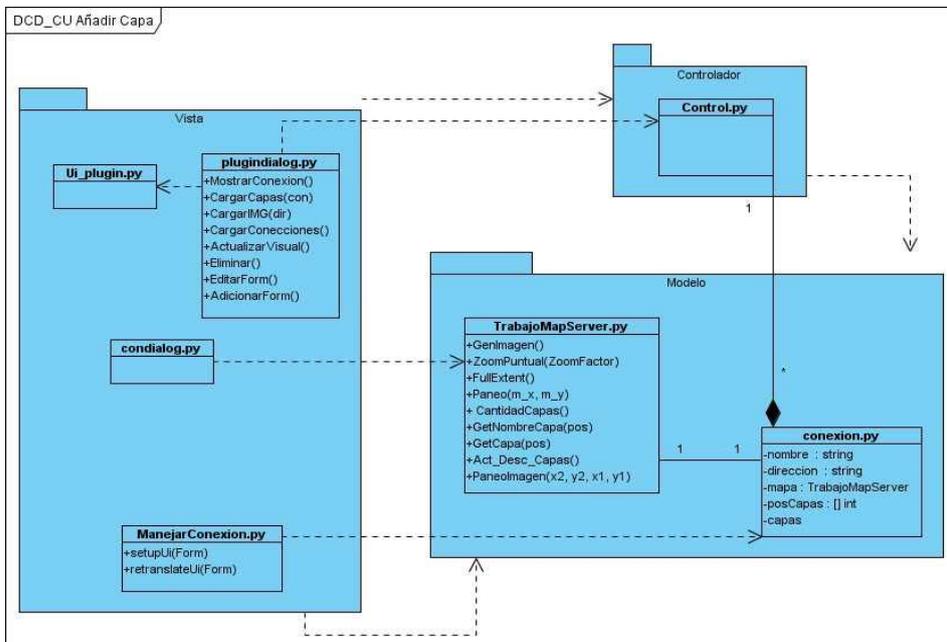


FIGURA 9 DIAGRAMA DE CLASE DEL DISEÑO CU AÑADIR CAPA.

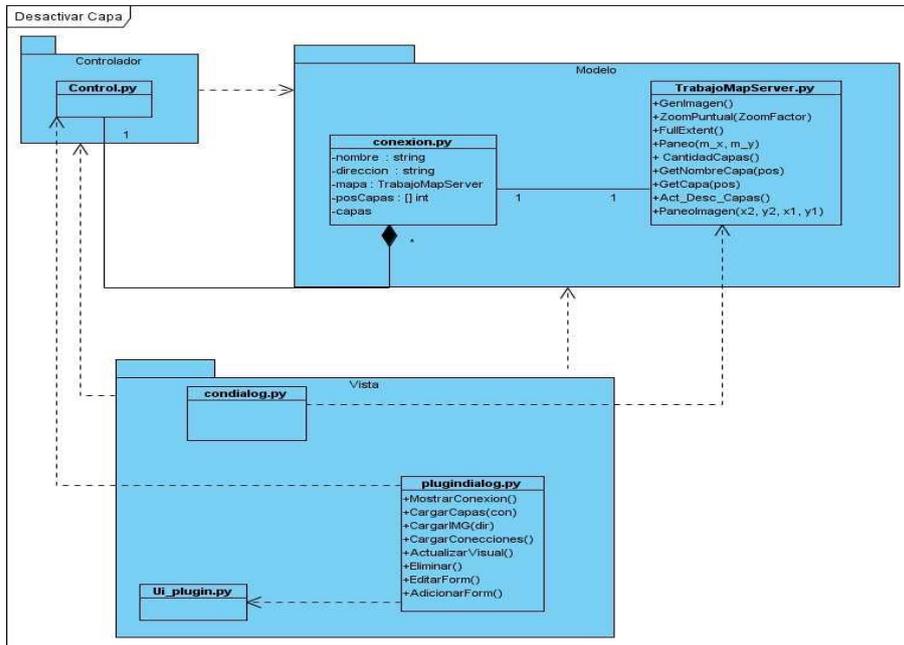


FIGURA 10 DIAGRAMA DE CLASE DEL DISEÑO CU DESACTIVAR CAPAS.

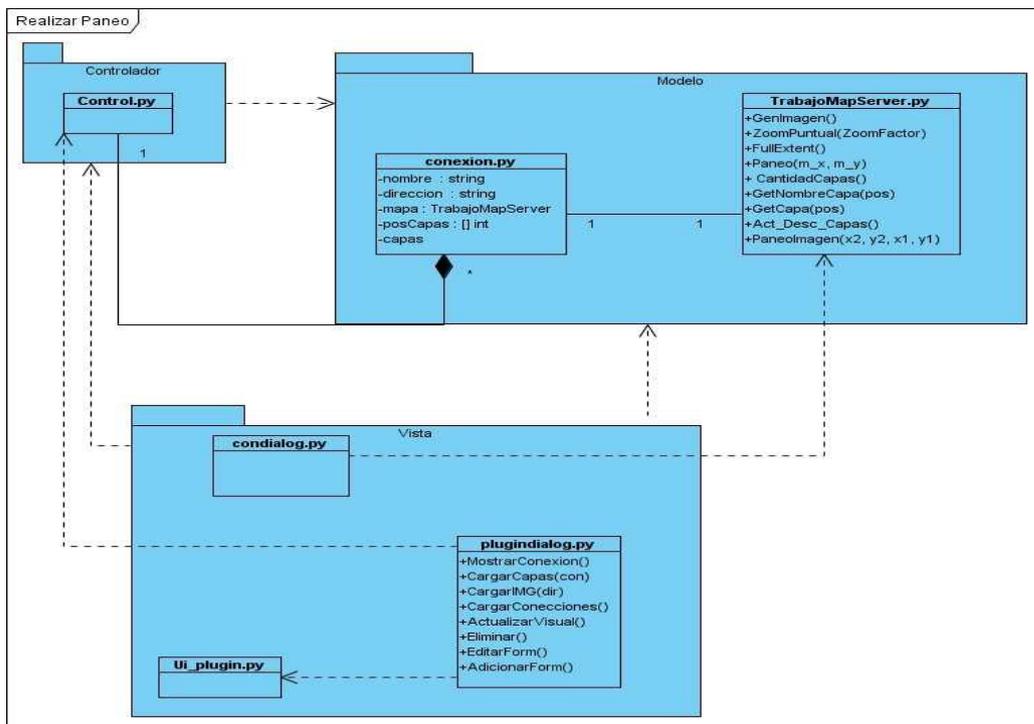


FIGURA 11 DIAGRAMA DE CLASE DEL DISEÑO CU REALIZAR PANEOS.

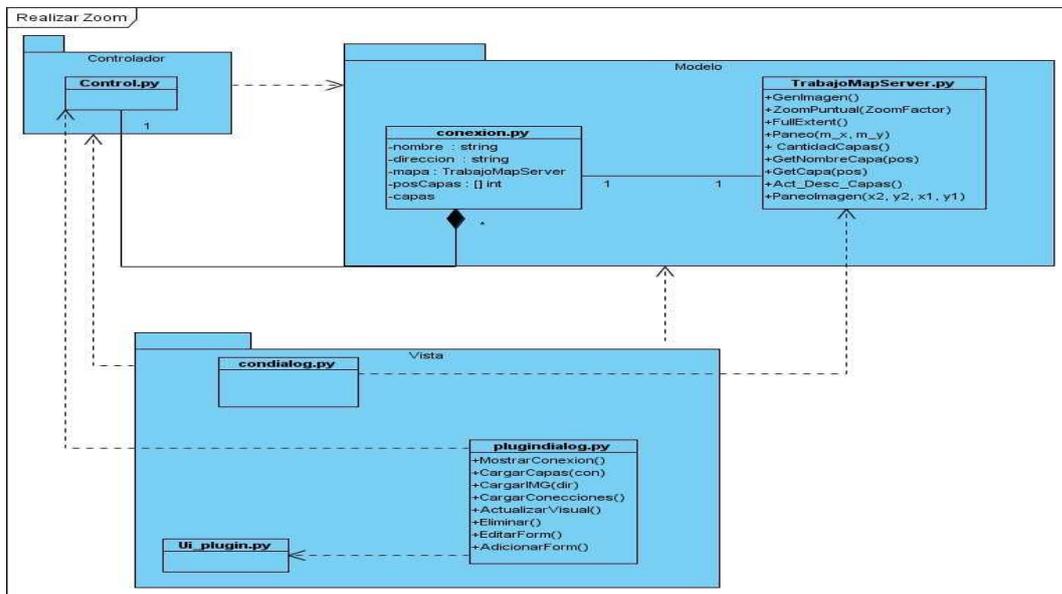


FIGURA 12 DIAGRAMA DE CLASE DEL DISEÑO CU REALIZAR ZOOM.

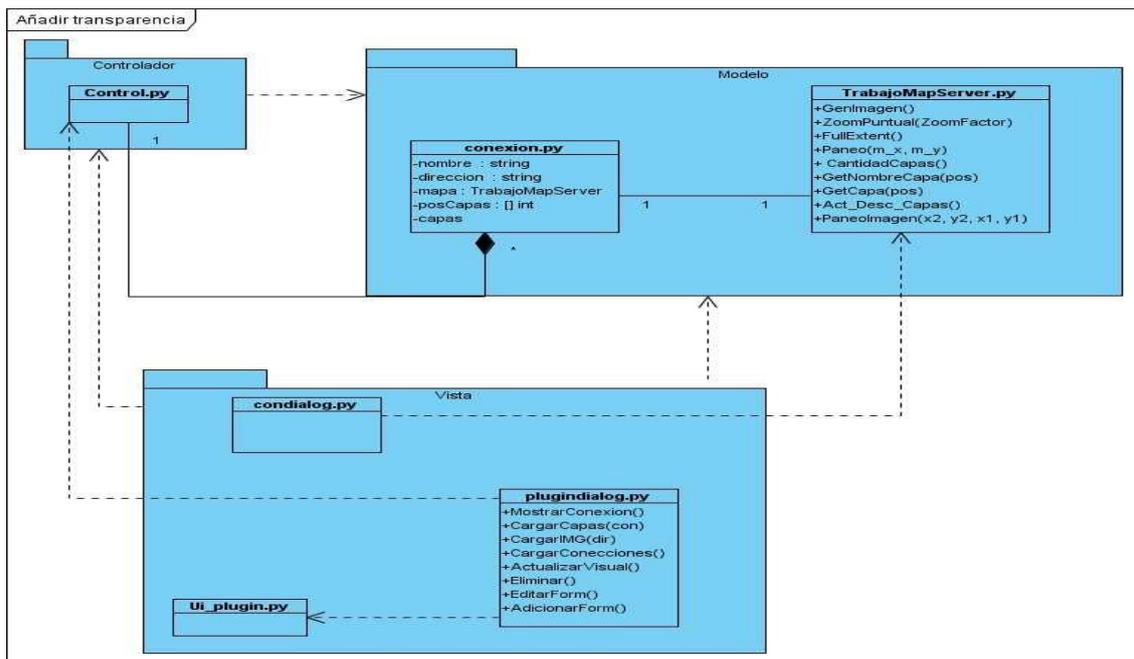


FIGURA 13 DIAGRAMA DE CLASE DEL DISEÑO CU AÑADIR TRANSPARENCIA.

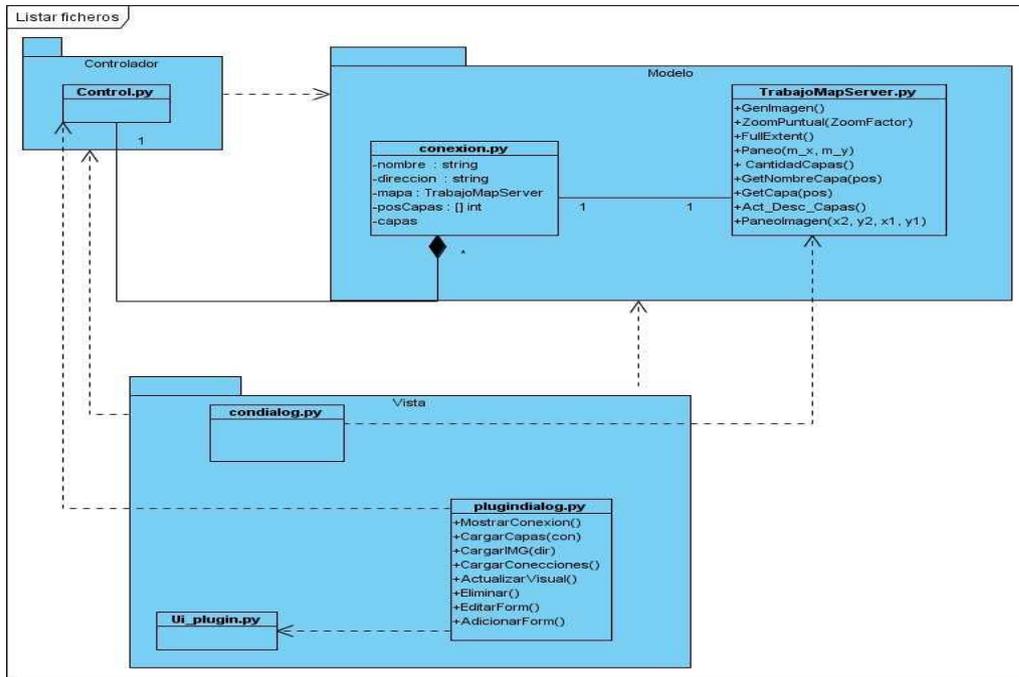


FIGURA 14 DIAGRAMA DE CLASE DEL DISEÑO CU LISTAR FICHEROS MAPFILE.

ANEXO 4 DIAGRAMAS DE SECUENCIA DEL DISEÑO

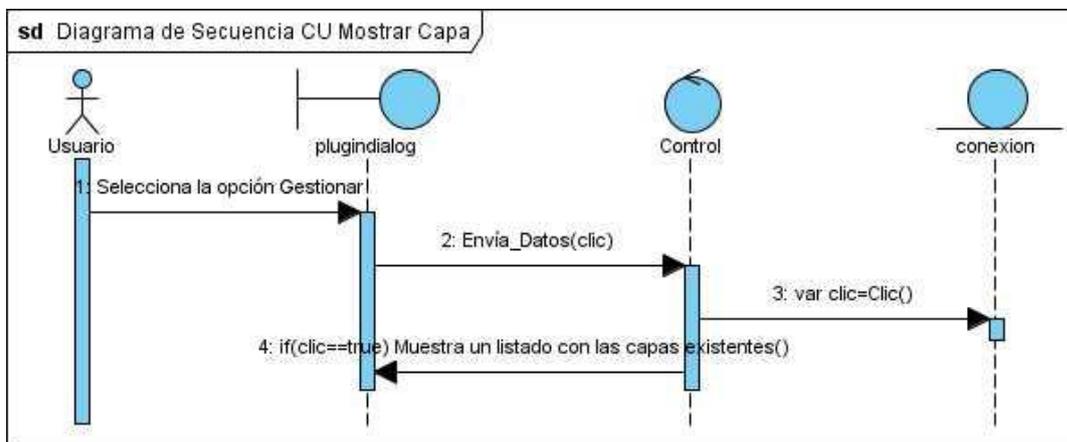


FIGURA 15 DIAGRAMA DE SECUENCIA CU MOSTRAR CAPA.

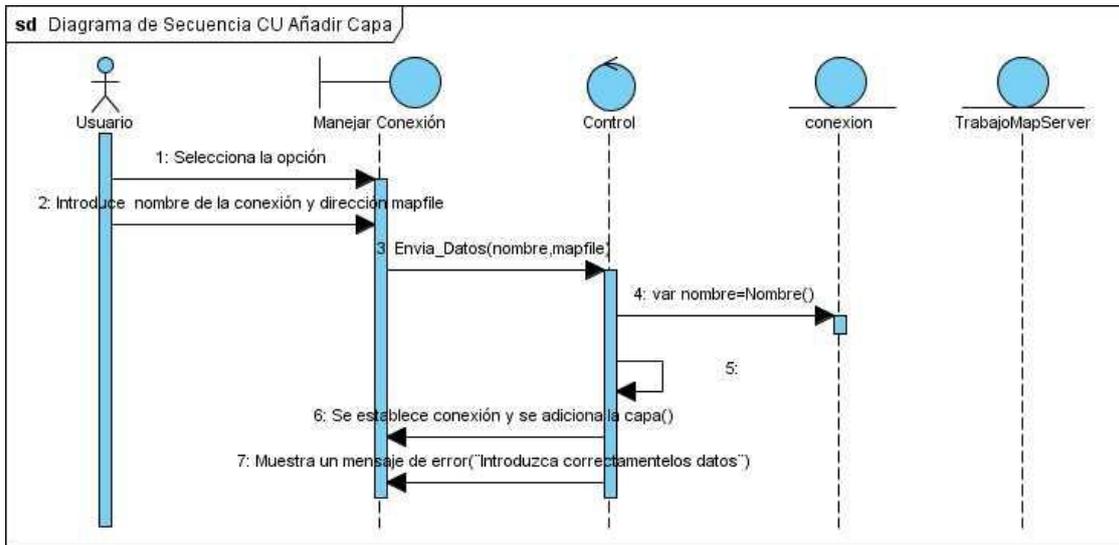


FIGURA 16 DIAGRAMA DE SECUENCIA CU AÑADIR CAPA.

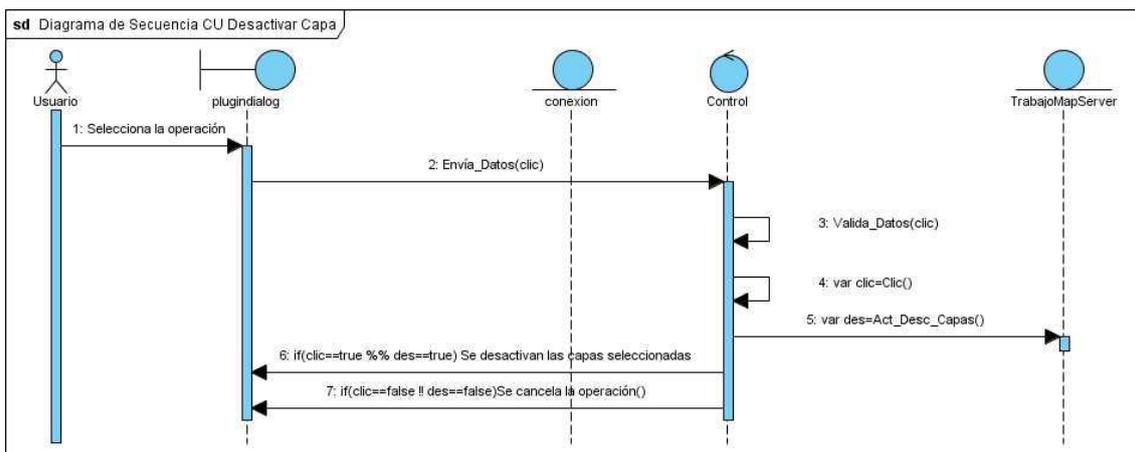


FIGURA 17 DIAGRAMA DE SECUENCIA CU DESACTIVAR CAPA.

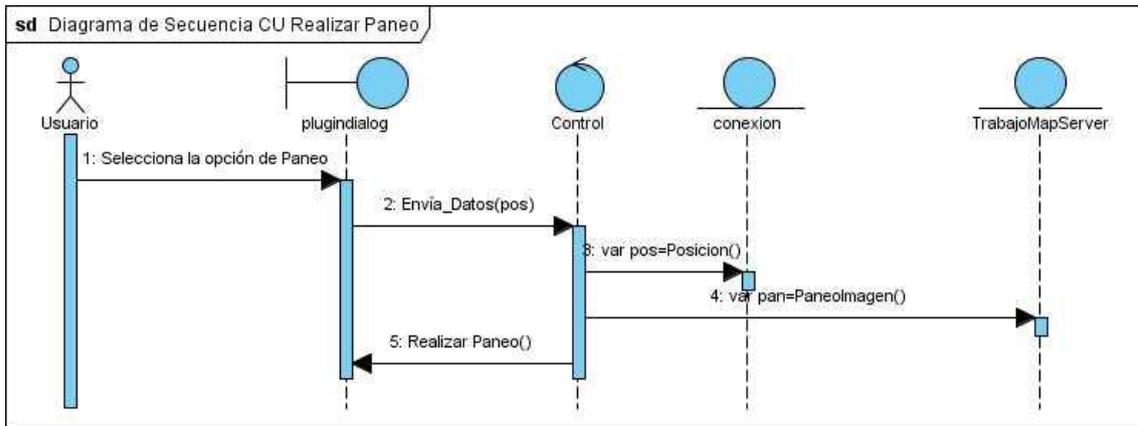


FIGURA 18 DIAGRAMA DE SECUENCIA CU REALIZAR PANEO.

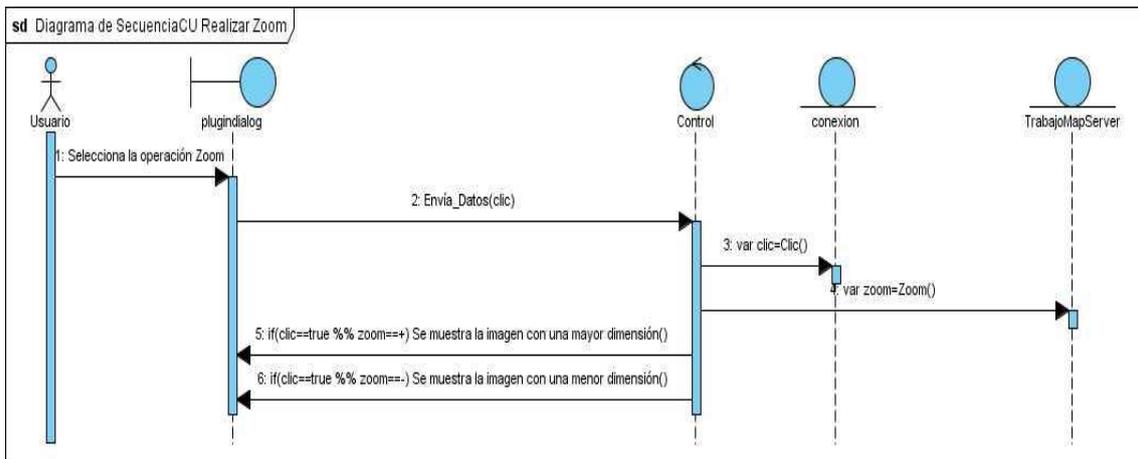


FIGURA 19 DIAGRAMA DE SECUENCIA CU REALIZAR ZOOM.

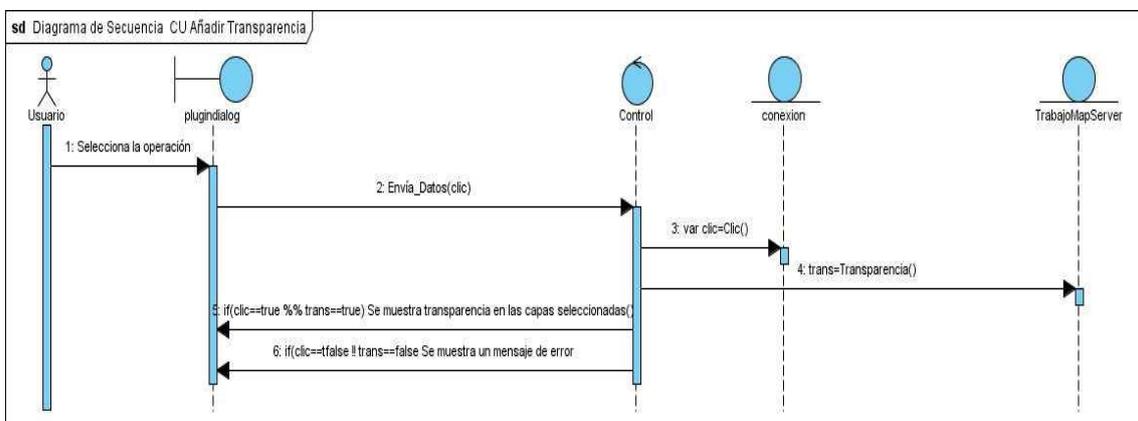


FIGURA 20 DIAGRAMA DE SECUENCIA CU AÑADIR TRANSPARENCIA.

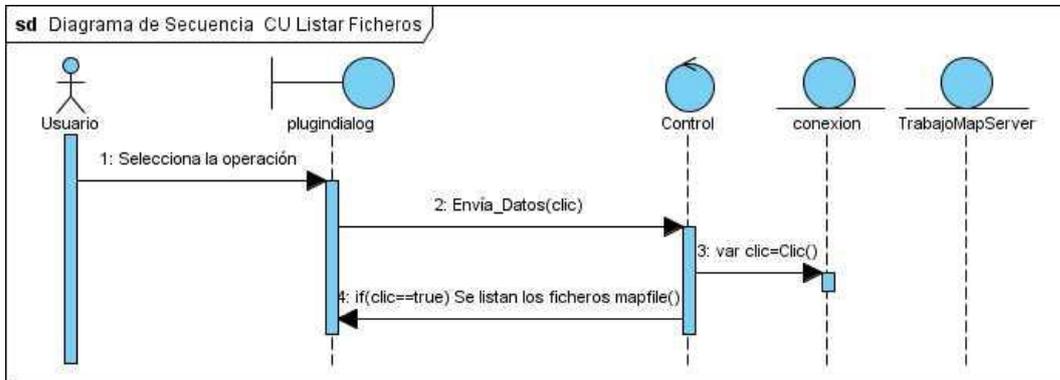


FIGURA 21 DIAGRAMA DE SECUENCIA CU LISTAR FICHEROS MAPFILE.

Anexo 5 Casos de Prueba

CU Añadir capa MapServer

Descripción General.

El caso de uso inicia cuando el usuario selecciona, el botón o la acción de Agregar capa MapServer. El caso de uso finaliza cuando el usuario completa las operaciones (realizar conexión, extraer capas) con la capa MapServer.

Condiciones de Ejecución.

El nombre de la conexión no debe existir en el sistema.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
Añadir Capa MapServer.	EC 1.1: Añadir capa correctamente.	Esta funcionalidad le permite al usuario adicionar una nueva capa.	<ul style="list-style-type: none"> El actor selecciona la opción de “Agregar capa MapServer”... El sistema muestra una interfaz que le permitirá al usuario crear una Nueva conexión con el servidor de mapas MapServer. El actor selecciona la opción de “Nueva Capa”. El sistema le muestra una nueva ventana pidiéndole al actor que introduzca los datos para poder realizar la conexión. Entre los datos se encuentran: el nombre de la conexión y la dirección del fichero
	EC 1.2: Realizar conexión con MapServer	Esta funcionalidad permite al usuario establecer una	<ul style="list-style-type: none"> El usuario selecciona la opción: Realizar conexión. El sistema muestra una interfaz que le permitirá al usuario: introducir la dirección

	<p>EC 1.3: No realizar la conexión con MapServer por no introducir el nombre de la conexión ni la dirección del fichero mapfile.</p>	<p>El sistema no muestra los resultados esperados ya que el usuario no introdujo los datos.</p>	<ul style="list-style-type: none"> • El usuario no introduce el nombre o la dirección del fichero mapfile. • El sistema desactiva el botón “Agregar”.
	<p>EC 1.4: Extraer capas de MapServer que se van a mostrar correctamente.</p>	<p>Esta funcionalidad permite al usuario seleccionar sólo las capas que desea mostrar.</p>	<ul style="list-style-type: none"> • El usuario selecciona la conexión a mostrar. • El sistema muestra una pre-visualización de la misma. • El usuario selecciona la opción “Cargar en GeoQ”. • El sistema extrae la conexión como una capa hacia GeoQ.

TABLA 12 SECCIONES A PROBAR CU AÑADIR CAPA MAPSERVER

Descripción de las variables

No	Nombre	Clasificación	Valor Nulo	Descripción
1	Nombre de la conexión	Campo de texto	No	
2	Mapfile	Campo de Selección	No	

3	Nombre de la capa	Campo de Selección	No	
---	-------------------	--------------------	----	--

Matriz de Datos CU Añadir capa MapServer

ID del escenario	Escenario	Nom_conexión	Dir_Mapfile	Nom_Capa	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Añadir capa correctamente	V/ "Ejemplo"	V/ "D:\MY\cartografia"	V/ "Viales"	Se añade correctamente la capa.	Satisfactorio.
EC 1.2	Realizar conexión con MapServer correctamente	V/ "Ejemplo"	V/ "D:\MY\cartografia"	V/ "Viales"	Se realiza correctamente la conexión.	Satisfactorio.
EC 1.3	No realizar la conexión con MapServer por no introducir la dirección del fichero	V/ "Ejemplo"	I/ " "	n/a	No se realiza la conexión.	Satisfactorio.
EC 1.4	No realizar la conexión con MapServer por no introducir el nombre de la conexión	I/ " "	V/ "D:\MY\cartografia"	V/ "Viales"	No se realiza la conexión.	Satisfactorio.
EC 1.5	Extraer capas de MapServer que se van a mostrar correctamente	V/ "Ejemplo"	V/ "D:\MY\cartografia"	V/ "Viales"	Se extrae la capa.	Satisfactorio.

TABLA 13 SC1 ADICIONAR CAPA MAPSERVER

CU Realizar zoom a la capa MapServer

Descripción General.

El caso de uso inicia cuando el usuario selecciona, el botón o la acción de Realizar Zoom y según el zoom seleccionado es que procede a realizar su función.

Condiciones de Ejecución.

Para lograr una navegación tiene que tener cargada o creada una capa. Además debe de tener seleccionada la capa para poder realizar las diferentes opciones del Zoom.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
Realizar zoom	EC 1.1: Realizar zoom correctamente	Esta funcionalidad permite al usuario acercar o alejar el	<ul style="list-style-type: none"> El usuario selecciona una de las dos operaciones presentes: "Realizar
	EC 1.2: Realizar zoom +	Esta funcionalidad permite mostrar al usuario el mapa con un tamaño mayor que el que tenía	<ul style="list-style-type: none"> El usuario da clic en el área de trabajo donde se encuentra el mapa en el que se está trabajando y especificando mediante el clic dado, el lugar al que desea visualizar. El sistema captura la posición
	EC 1.3: Realizar zoom-	Esta funcionalidad permite mostrar al usuario el mapa con un tamaño menor que el que tenía anteriormente	<ul style="list-style-type: none"> El usuario da clic en el área de trabajo donde se encuentra el mapa en el que se está trabajando y especificando mediante el clic dado, el lugar al que desea visualizar. El sistema captura la posición mediante el clic seleccionado por el usuario, permitiendo llevar hacia detrás el mapa, mostrando una menor

			visualización de la posición capturada.
	EC 1.4: Realizar zoom general	Esta funcionalidad permite mostrar al usuario el mapa con su tamaño y posición original.	<ul style="list-style-type: none"> • El usuario selecciona la opción “Zoom General” • El sistema muestra la imagen en su tamaño y posición original.

TABLA 14 SECCIÓN A PROBAR CU REALIZAR ZOOM

Descripción de las variables

No	Nombre	Clasificación	Valor Nulo	Descripción
1	<i>Nombre de la capa</i>	<i>Campo de selección</i>	<i>No</i>	

Matriz de Datos CU Realizar zoom

ID del escenario	Escenario	Nom_Capa	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Realizar zoom correctamente	V/ “Viales”	Se muestra correctamente el	Satisfactorio.
EC 1.2	Realizar zoom +	V/ “Viales”	Se muestra correctamente el mapa con una mayor	Satisfactorio.

ID del escenario	Escenario	Nom_Capa	Respuesta del Sistema	Resultado de la Prueba
EC 1.3	Realizar zoom -	V/ "Viales"	Se muestra correctamente el mapa con una menor	Satisfactorio.
EC 1.4	Realizar zoom general.	V/ "Viales"	Se muestra correctamente el mapa con su	Satisfactorio.

TABLA 15 SC1 REALIZAR ZOOM

CU Añadir transparencia a la capa MapServer

Descripción General

El caso de uso inicia cuando el usuario necesita añadir transparencia a una capa MapServer.

Condiciones de Ejecución

Debe existir al menos una capa.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central

<p>Añadir transparencia a la capa MapServer.</p>	<p>EC 1.1: Añadir transparencia correctamente.</p>	<p>Esta funcionalidad permite al usuario cambiar la nitidez a la capa una vez que haya sido seleccionada.</p>	<ul style="list-style-type: none"> • El usuario selecciona las propiedades de la capa y selecciona "Transparencia". • El sistema muestra una nueva ventana con la transparencia actual con un valor de 0 a 100. • El usuario establece la transparencia deseada. • El sistema aplica transparencia a la capa seleccionada.
--	--	---	--

TABLA 16 SECCIÓN A PROBAR CU AÑADIR TRANSPARENCIA

Descripción de las variables

No	Nombre	Clasificación	Valor Nulo	Descripción
1	Nombre de la capa.	Campo de selección	No	
2	Nombre de la conexión	Campo de texto	No	

Matriz de Datos CU Añadir transparencia

ID del escenario	Escenario	Nom_Capa	Nom_Conexión	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Añadir transparencia correctamente	V/ "Viales"	V/ "Ejemplo"	Se muestra la capa con una nueva transparencia.	Satisfactorio.

TABLA 17 SC 1 AÑADIR TRANSPARENCIA

CU Listar ficheros mapfile

Descripción General.

El caso de uso inicia cuando el usuario desea establecer nuevas conexiones para ello debe seleccionar el fichero mapfile; para ello tendrá la opción de seleccionar la dirección de una lista de ficheros con tan sólo dar clic sobre uno de ellos.

Condiciones de Ejecución.

Debe existir al menos un fichero mapfile.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
Listar ficheros mapfile.	EC 1.1: Listar ficheros correctamente .	Esta funcionalidad permite listar los ficheros mapfile que existen.	<ul style="list-style-type: none"> • El usuario selecciona la opción “Adicionar” o “Editar”. • El sistema muestra un listado con los ficheros mapfile existentes. • El usuario da clic sobre un fichero mapfile. • El sistema muestra de forma automática los datos del mapfile.

TABLA 18 SECCIÓN A PROBAR CU LISTAR FICHEROS MAPFILE

Descripción de las variables

No	Nombre	Clasificación	Valor Nulo	Descripción
1	Mapfile	Campo de selección.	No	

Matriz de Datos CU Listar ficheros mapfile.

ID del escenario	Escenario	Dir_Mapfile	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Listar ficheros correctamente.	V/ "D:\MY\cartografía"	Muestra e listado de ficheros mapfile.	Satisfactorio.

TABLA 19 SC 1 LISTAR FICHEROS MAPFILE

CU Realizar paneo a la capa MapServer.

Descripción General.

El caso de uso inicia cuando el usuario necesita hacerle paneo a una región de una de las capas MapServer.

Condiciones de Ejecución.

No presenta

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
Realizar paneo a la capa MapServer.	EC 1.1: Realizar paneo correctamente.	Esta funcionalidad permite al usuario desplazar el mapa hacia una determinada región.	<ul style="list-style-type: none"> • El usuario selecciona la opción de Paneo y da clic en la región que quiere visualizar. • El sistema desplaza el mapa a la posición seleccionada. • El sistema activa la opción de "Paneo".

TABLA 20 SECCIÓN A PROBAR CU REALIZAR PANE0

Descripción de las variables

No	Nombre	Clasificación	Valor	Descripción
----	--------	---------------	-------	-------------

			Nulo	
1	Nombre de la ..	Campo de texto	No	
2	Nombre de la capa	Campo de selección	No	

Matriz de Datos CU Realizar paneo

ID del escenario	Escenario	Nom_conexión	Nom_Capa	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Realizar paneo correctamente.	V/ "Ejemplo"	V/ "Viales"	Desplaza la capa.	Satisfactorio.

TABLA 21 SC1 REALIZAR PANEEO

CU Mostrar capa MapServer

Descripción General.

El caso de uso inicia cuando el usuario necesita mostrar capas MapServer.

Condiciones de Ejecución.

Se debe seleccionar al menos una capa.

Debe existir al menos una capa MapServer.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
----------------------	--------------------------	---------------------------------	---------------

Mostrar capa MapServer.	EC 1.1: Mostrar capa correctamente.	Esta funcionalidad permite al usuario seleccionar las capas que desea	<ul style="list-style-type: none"> El usuario selecciona la capa que desea visualizar. El sistema verifica que existe al menos una capa.
	EC 1.2: No mostrar capa porque no existe al menos una.	El sistema no muestra los resultados ya no existen capas.	<ul style="list-style-type: none"> El sistema muestra un mensaje de error "Debe seleccionar al menos una capa."

TABLA 22 SECCIÓN A PROBAR CU MOSTRAR CAPA

Descripción de las variables

No	Nombre	Clasificación	Valor Nulo	Descripción
1	Nombre de la capa	Campo de selección	No	
2	Nombre de la conexión	Campo de texto	No	

Matriz de Datos CU Mostrar capa

ID del escenario	Escenario	Nom_Conexión	Nom_Capa	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Mostrar capa correctamente	V/ "Ejemplo"	V/ "Viales"	Muestra la capa seleccionada.	Satisfactorio.
EC 1.2	No mostrar capa porque existe al menos una.	V/ "Ejemplo"	I/ " "	No muestra la capa.	Satisfactorio.

TABLA 23 SC 1 MOSTRAR CAPA

CU Gestionar Conexión

Descripción General.

El caso de uso comienza cuando el usuario necesita gestionar la conexión con MapServer. El caso de uso termina cuando el usuario completa las operaciones a realizar (adicionar, mostrar, eliminar) con la conexión.

Condiciones de Ejecución.

Debe estar creada la conexión

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
Gestionar conexión	EC 1.1: Gestionar conexión correctamente.	Esta funcionalidad permite adicionar, mostrar, eliminar una conexión.	<ul style="list-style-type: none"> • El usuario selecciona la opción Gestionar Conexión con MapServer. • El sistema muestra todas las conexiones existentes.
	EC 1.2: Adicionar conexión correctamente.	Esta funcionalidad permite al usuario adicionar una nueva conexión.	<ul style="list-style-type: none"> • El usuario selecciona la opción "Adicionar". • El sistema muestra una ventana para introducir los datos de la conexión a adicionar: nombre y
	EC 1.3: No adicionar conexión por introducir una conexión existente.	El sistema muestra un ícono de error.	El sistema muestra un ícono de error indicando el mismo y desactiva el botón "Agregar".

	<p>EC 1.4: No adicionar conexión por no introducir la dirección mapfile</p>	<p>El sistema no muestra los resultados esperados ya que el usuario no introdujo correctamente la dirección mapfile.</p>	<p>En caso de que la dirección del fichero mapfile no sea correcta, se muestra un mensaje de error.</p>
	<p>EC 1.5 Editar conexión correctamente.</p>	<p>Esta funcionalidad permite editar la conexión seleccionada por el usuario.</p>	<ul style="list-style-type: none"> • El usuario selecciona la opción “Editar”. • El sistema muestra una nueva ventana con la conexión seleccionada • El usuario realiza los cambios deseados a la conexión seleccionada. • El sistema realiza los cambios y guarda la conexión.
	<p>EC 1.6 No editar conexión por no introducir correctamente el nombre de la conexión</p>	<p>El sistema no muestra los resultados esperados ya que el usuario no introdujo correctamente el nombre de la conexión.</p>	<ul style="list-style-type: none"> • En caso que la conexión ya existe. El sistema muestra un ícono de error indicando el mismo y desactiva el botón “Editar”.
	<p>EC 1.7 No editar conexión por no</p>	<p>El sistema no muestra los resultados</p>	<ul style="list-style-type: none"> • En caso de que la dirección del fichero mapfile no sea correcta, se

	introducir correctamente la dirección del fichero mapfile	esperados ya que el usuario no introdujo correctamente la dirección mapfile.	muestra un mensaje de error
	EC 1.8 Eliminar conexión correctamente.	Esta funcionalidad permite eliminar la conexión seleccionada por el usuario.	<ul style="list-style-type: none"> • El usuario selecciona la conexión y elige la opción “Eliminar”. • El sistema muestra un mensaje para verificar que se desea realizar la operación. • El usuario selecciona “Aceptar”. • El sistema elimina la conexión.

TABLA 24 SECCIÓN A PROBAR CU GESTIONAR CONEXIÓN

Descripción de las variables

No	Nombre	Clasificación	Valor Nulo	Descripción
1	Nombre de la conexión	Campo de texto	No	
2	Mapfile	Campo de selección	No	
3	Nombre de la capa	Campo de selección	No	

Matriz de datos CU Gestionar conexión

ID del escenario	Escenario	Nom_conexión	Nom_Capa	Dir_Mapfile	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Gestionar conexión correctamente.	V/ "Ejemplo"	V/ "Viales"	V/ "D:\MY\cartografia"		Satisfactorio.
EC 1.2	Adicionar conexión correctamente.	V/ "Ejemplo"	V/ "Viales"	V/ "D:\MY\cartografia"		Satisfactorio
EC 1.3	No adicionar conexión por introducir una	I/ "Ejemplo"	n/a	n/a		Satisfactorio
EC 1.4	No adicionar conexión por no introducir la	V/ "Ejemplo"	n/a	I/ ""		Satisfactorio
EC 1.5	Eliminar conexión correctamente.	V/ "Ejemplo"	n/a	n/a		Satisfactorio

TABLA 25 SC 1 GESTIONAR CONEXIÓN