



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
VICERRECTORÍA DE FORMACIÓN
DIRECCIÓN DE FORMACIÓN POSTGRADUADA

Título: Desarrollo de una interfaz de servicios basada en software libre para la manipulación de mapas digitales.

Tesis presentada en opción del grado científico de
Máster en Informática Aplicada.

Autor: Ing. Luis Lamela Fung

Tutora: Dr. Tatiana Delgado Fernández.

Ciudad de la Habana, Junio del 2007.

SÍNTESIS

En el desarrollo de aplicaciones informáticas, que requieran incorporar una componente espacial, se ve limitado el tiempo del desarrollo, debido al desconocimiento de la geomática.

En una comunidad de desarrollo donde conviven aplicaciones de diferentes características se hace necesario una especialización en cada una de las tecnologías que a pesar de ser independiente una de otra, deben llegar a interactuar de manera eficiente y coordinada para lograr los resultados deseables.

Los servidores de mapas de manera general han dado solución a la problemática de la visualización en un entorno WEB, de la información asociada a una empresa o territorio, desde el punto de vista de su distribución espacial, lo que se ha convertido en una tecnología deseable para la toma de decisiones en diferentes niveles organizativos, políticos y sociales.

Con el desarrollo de aplicaciones basadas en software libre, es posible encontrar librerías y aplicaciones orientadas al manejo de recursos geográficos de forma gratuita, sin embargo, a pesar de la solución que brindan, requieren por parte de los programadores un estudio a profundidad para poder lograr un objetivo específico.

La utilización de MapServer, proyecto desarrollado por la Universidad de Minnesota, no queda exento de este problema, a pesar de ser uno de los proyectos líderes en la publicación y manejo de información geográfica, en la WEB.

MapServer, a pesar de ser uno de los servidores de Mapas OpenSource que más se utilizan para la solución a las tareas descritas anteriormente también adolece de la problemática planteada, referida a la dificultad de su uso por otros desarrolladores no familiarizados con la Geomática.

El objetivo de este trabajo consiste en proponer una solución para proveer a MapServer de una interfaz de Servicios con vistas a su utilización en una comunidad de programadores, tomando como aplicación a la Universidad de las Ciencias Informáticas.

INDICE

| | |
|--|----|
| INTRODUCCIÓN | 1 |
| PROBLEMA DE INVESTIGACIÓN | 3 |
| HIPÓTESIS | 3 |
| OBJETIVO GENERAL | 3 |
| OBJETIVOS ESPECÍFICOS | 4 |
| OBJETO DE INVESTIGACIÓN | 4 |
| CAMPO DE INVESTIGACIÓN | 4 |
| RESULTADOS ESPERADOS | 4 |
| NOVEDAD CIENTÍFICA | 5 |
| MÉTODO CIENTÍFICO | 6 |
| CAPÍTULO 1 - FUNDAMENTACIÓN TEÓRICA | 7 |
| 1.1 INTRODUCCIÓN | 7 |
| 1.2 ESTADO DEL ARTE SERVIDORES DE MAPAS OPENSOURCE | 8 |
| 1.3 MAPSERVER: EL SERVIDOR DE MAPAS SELECCIONADO | 12 |
| 1.4 LAS BASES DE DATOS ESPACIALES | 14 |
| 1.5 DATOS ESPACIALES Y FRAMEWORK PARA PERSISTENCIA | 16 |
| 1.6 ISO TC/211 | 16 |
| 1.7 OPENGIS® | 17 |
| 1.8 WEB MAP SERVICE: EL SERVICIO BÁSICO DE LOS SERVIDORES DE MAPAS | 18 |
| 1.9 ARQUITECTURA ORIENTADA A SERVICIOS | 19 |
| 1.10 CONCLUSIONES DEL CAPITULO 1 | 19 |
| CAPÍTULO 2 – DESCRIPCIÓN DE LA PROPUESTA | 21 |
| 2.1 CARACTERIZACIÓN DEL FENÓMENO | 21 |
| 2.2 ARQUITECTURA MULTICAPAS | 21 |
| 2.3 MODELO DE REFERENCIA | 24 |
| 2.3.1 ARQUITECTURA PROPUESTA | 24 |

| | | |
|---|--|-----------|
| 2.3.1.1 | DESCRIPCIÓN DE MÉTODOS Y TÉCNICAS UTILIZADAS | 30 |
| CAPÍTULO 3 – ANÁLISIS DE LOS RESULTADOS..... | | 42 |
| 3.1 | INTRODUCCIÓN..... | 42 |
| 3.2 | MODELO DE DOMINIO | 43 |
| 3.3 | ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE..... | 43 |
| 3.3.1 | REQUISITOS FUNCIONALES | 43 |
| 3.3.2 | REQUISITOS NO FUNCIONALES..... | 44 |
| 3.4 | MODELO DE SISTEMA..... | 45 |
| 3.4.1 | DESCRIPCIÓN DE ACTORES | 45 |
| 3.4.2 | DIAGRAMA DE CASOS DE USO DEL SISTEMA | 45 |
| 3.4.3 | ESPECIFICACIÓN DE CASOS DE USO DEL SISTEMA | 46 |
| 3.5 | ANÁLISIS Y DISEÑO | 51 |
| 3.5.1 | MODELO DE CLASES DEL ANÁLISIS..... | 51 |
| 3.5.2 | MODELO DE CLASES DEL DISEÑO | 52 |
| 3.5.3 | MODELO DE DATOS | 57 |
| 3.5.4 | MODELO DE IMPLEMENTACIÓN..... | 58 |
| 3.5.4.1 | DIAGRAMA DE COMPONENTES..... | 58 |
| 3.5.4.2 | MODELO DE DESPLIEGUE | 59 |
| 3.6 | MODIFICACIONES REALIZADAS A HIBERNATE PARA INCORPORAR LOS TIPOS DE DATOS ESPACIALES AL MOTOR DE PERSISTENCIA. | 60 |
| 3.7 | CREACIÓN DE LA INTERFAZ DE SERVICIOS..... | 62 |
| 3.8 | ANÁLISIS DE APLICABILIDAD..... | 63 |
| CONCLUSIONES..... | | 67 |
| RECOMENDACIONES..... | | 69 |
| GLOSARIO DE TÉRMINOS..... | | 70 |
| BIBLIOGRAFÍA..... | | 72 |
| ANEXOS | | 76 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1– Presencia de los servidores de mapas en internet. (Fuente: 23) | 13 |
| Figura 2 – Arquitectura básica de servidores de Mapas (Fuente: [9]) | 23 |
| Figura 3 – Arquitectura interna de los servidores de Mapas OpenSource. (Fuente: [40]) | 24 |
| Figura 4 – Arquitectura para interfaz de servicios cartográficos. | 25 |
| Figura 5 – Representación características que se incorporan al servidor de mapas.... | 26 |
| Figura 6 – Estructura de paquetes de la interfaz de servicios cartográficos. | 27 |
| Figura 7– Arquitectura interna de Hibernate (Fuente: [http://hibernate.org]) | 33 |
| Figura 8– Árbol de jerarquía del MapFile de MapServer (Fuente: [http://mapserver.gis.umn.edu]) | 36 |
| Figura 9 – Diagrama de casos de usos | 45 |
| Figura 10 – Modelo de clase del análisis para el caso de uso Realizar Operaciones. . | 51 |
| Figura 11 – Modelo de clase del análisis para el caso de uso Gestionar Objeto..... | 52 |
| Figura 12 – Modelo de clase del análisis para el caso de uso Obtener Información. | 52 |
| Figura 13 – Diagrama de clases del diseño del Caso de Uso Realizar Operaciones...53 | |
| Figura 14 – Diagrama de clases del diseño del Caso de Uso Gestionar Objeto | 54 |
| Figura 15 – Diagrama de clases del diseño del Caso de Uso Obtener Información..... | 55 |
| Figura 16 – Diagrama de secuencia del Caso de Uso Realizar Operaciones (Escenario Dibujar Mapa). | 56 |
| Figura 16 – Diagrama de secuencia del Caso de Uso Realizar Operaciones (Escenario Ampliar Mapa). | 56 |
| Figura 18 – Diagrama de secuencia del Caso de Uso Gestionar Objeto (Escenario Construir Línea)..... | 57 |
| Figura 17 – Diagrama de secuencia del Caso de Uso Obtener Información. | 57 |
| Figura 17 – Modelo de datos utilizado en la interfaz de servicios cartográficos | 58 |
| Figura 18 – Modelo de componentes de la interfaz de servicios | 59 |
| Figura 19 – Diagrama de despliegue para la interfaz de servicios. | 60 |
| Figura 20 – Interfaz UserType | 61 |
| Figura 21 – Interfaz GeometryUserType | 61 |

Figura 22 – Resultado de una aplicación que utiliza la interfaz de manipulación de mapas.66

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1– Tabla comparativa entre servidores de mapas OpenSource..... | 12 |
| Tabla 2 – Característica 2-tier n-tier. (Fuente: [15]) | 22 |
| Tabla 3 – Ejemplo de configuración de a partir del MapFile..... | 37 |
| Tabla 4 – Ejemplo de configuración con conexión a un SGBD..... | 38 |
| Tabla 5 – Esquema de configuración basado en XML para la definición de los Web Services | 62 |
| Tabla 6 – Ejemplo de la utilización del servicio GetMap a través de MapServer..... | 63 |
| Tabla 7 – Ejemplo de la utilización del servicio GetMap a través de la interfaz..... | 63 |
| Tabla 8 – Ejemplo de utilización de la interfaz de manipulación de mapas utilizando PHP..... | 64 |
| Tabla 9 – Ejemplo de utilización de la interfaz de manipulación de mapas utilizando JAVA | 64 |

INTRODUCCIÓN

El desarrollo de las Tecnologías de la Información y las Comunicaciones ha conllevado a la creación de nuevas herramientas en los más diversos campos que permiten mejorar el proceso de toma de decisiones en toda la sociedad.

Los Sistemas de Información Geográfica (SIG) se basan en técnicas computacionales que permiten la adquisición, manejo, análisis y representación de la información geográfica [2].

El beneficio del uso de los SIG se aprecia en numerosas vertientes, entre las que se destacan la facilidad para la integración de los datos de diferentes fuentes, las consultas y los análisis espaciales, la mejora de la integración organizacional y el apoyo al trabajo multidisciplinario, el apoyo al proceso de toma de decisiones y la facilidad para la creación de mapas y emisión de otros reportes. [14].

La evolución de estos sistemas ha transitado desde los enfoques *stand-alone* de los años 60-80, hasta aquellos enfoque más recientes donde la información geográfica se comparte en entornos multiusuarios empresariales y sectoriales hasta incluir el acceso de toda la sociedad a servicios y datos geoespaciales, gracias a Internet.

En los SIG públicos, se hace necesario implementar Infraestructuras de Datos Espaciales (IDE) que establezcan un marco legal adecuado, y los estándares necesarios para lograr la interoperabilidad requerida.

Existe una clara necesidad, a todos los niveles, para poder acceder, integrar y utilizar datos espaciales procedentes de fuentes dispares con el fin de guiar la toma de decisiones. La habilidad entonces, para tomar decisiones acertadas colectivamente a nivel local, regional y global, depende de la implementación de una IDE que suministre una compatibilidad entre jurisdicciones, promoviendo el acceso y uso de datos. [10]

Uno de los servicios principales de estas Infraestructuras de Datos Espaciales lo constituyen los servidores de mapas, mediante los cuales el usuario puede tener acceso a los datos geoespaciales (mapas), sus informaciones asociadas y los metadatos de tales servicios de mapas. [24]

Una IDE permite interoperar y compartir datos y servicios ubicados en distintos entornos de hardware y software, de manera transparente para el usuario, que interactúa como si toda la información estuviera en un único lugar, con la simple utilización de su navegador de Internet.

Debido a lo anterior, resulta evidente que una IDE conlleva a un alto nivel organizativo y la colaboración de los diferentes actores a nivel institucional en aras de obtener todas las bondades que la misma permite; sin embargo es necesario destacar que es importante el factor humano, tecnológico y un conjunto de aplicaciones para obtener los resultados esperados.

A partir de la política trazada por el Ministerio de la Informática y las Comunicaciones de desarrollar aplicaciones basadas en software libre se realizó un estudio inicial de los servidores de Mapas para seleccionar el más adecuado a nuestras condiciones para, a partir de esta información, incorporar a las diferentes aplicaciones que se ejecutan en una comunidad de desarrollo, como el de la Universidad de las Ciencias Informáticas, de una componente geoespacial que permita describir la distribución geográfica de su información.

En una comunidad de desarrolladores de software, donde cada aplicación tiene características y usos diferentes es necesario llevar a cabo una especialización de las diferentes tecnologías y permitir además el uso adecuado e interactivo de las mismas en un entorno que facilite y haga más eficiente el desarrollo de aplicaciones integradas.

Debido a lo anterior es deseable contar con una plataforma de servicios geográficos que brinde la posibilidad de utilizar la información geoespacial sin la necesidad de realizar un estudio minucioso sobre las normas a tener en cuenta para desarrollar un Sistema de Información Geográfica interno para cada una de las aplicaciones, que además permita la integración con otras aplicaciones y que sea lo suficientemente consistente y estable, desde el punto de vista de las normas y las metodologías de trabajo propuestas por el estándar internacional de la serie ISO 19100, dedicado a la Información Geográfica y la Geomática.

En este sentido se desarrolló un SIG para la manipulación de la cartografía local de la Universidad de las Ciencias Informáticas obteniéndose resultados

satisfactorios, no obstante, dicho sistema no permite la interacción de las aplicaciones que conviven en la misma comunidad, lo que conllevó a que las aplicaciones que conviven en la comunidad no pudieran utilizar las funcionalidades del mismo para incorporar en sus resultados la integración de los datos de manera georeferenciada.

Paralelamente a la realización de este SIG se desarrolló un sistema para la generación de gráficos estadísticos, en la que se utilizó la información espacial. Esto conllevó a modificaciones en el SIG original y evidenció la necesidad de diseñar servicios que pudieran ser utilizados por los clientes de la comunidad de desarrolladores sin que esto conlleva a una dependencia de estos usuarios con la Geomática.

El presente trabajo analiza específicamente como es posible a partir de la utilización de MapServer incorporar una capa de persistencia para el manejo de los datos y una capa de servicios que permita a la comunidad de desarrolladores que la utilizarán, relacionar sus datos con la información geográfica del territorio objeto de análisis, sin necesidad de tener conocimientos de geomática, disminuyendo los tiempos de desarrollo de funcionalidades particulares para cada una de las aplicaciones de manera independiente.

PROBLEMA DE INVESTIGACIÓN.

Los servidores de Mapas OpenSource no proveen Servicios Web de cara al cliente que faciliten su explotación sin contar con conocimientos mínimos de Geomática.

HIPÓTESIS.

Utilizando tecnología ISAPI y otras, basadas en Arquitectura Orientada a Servicios (SOA) se puede aumentar el rendimiento de los Servidores de Mapas.

OBJETIVO GENERAL.

Proveer un servicio Web que permita la manipulación de la cartografía a las diferentes comunidades de desarrollo de la universidad que pueda ser utilizada sin conocimientos avanzados de geomática y las tecnologías asociadas.

OBJETIVOS ESPECÍFICOS

- Realizar un estudio de los Servidores de Mapas OpenSource.
- Selección del Servidor de Mapas más adecuado para nuestras condiciones actuales.
- Estudio de tecnologías y arquitecturas existentes para elevar el rendimiento del Servidor de Mapas seleccionado y posibilitar a la comunidad de desarrolladores de una interfaz de servicios fácil de utilizar sin necesidad de configuración previa.
- Definir una arquitectura que permita separar y detectar fácilmente las responsabilidades de cada capa de la misma, disminuyendo los tiempos de mantenimiento y modificaciones a la interfaz de servicios.

OBJETO DE INVESTIGACIÓN.

Análisis de la arquitectura de Servidores de Mapas y tecnologías asociadas.

CAMPO DE INVESTIGACIÓN.

Las tecnologías de Información Geoespacial basadas en Servicios Web.

RESULTADOS ESPERADOS.

Con la publicación de los servicios Web que manipulen información cartográfica es posible obtener funcionalidades espaciales en las diferentes aplicaciones de las comunidades de desarrollo sin tener conocimiento de las tecnologías que se emplean para llevar a cabo la manipulación de la cartografía, incorporando un valor añadido a las mismas.

Para lograr los objetivos planteados anteriormente la tesis se ha estructurado en tres capítulos.

Capítulo 1: En este capítulo se realiza un análisis del estado del arte relacionado con los diferentes servidores de mapas OpenSource, donde finalmente se fundamenta la selección del Servidor de Mapa que se utilizará. Además de esto se relacionan las tecnologías asociadas y los estándares y normativas que existen a nivel mundial para llevar a cabo la implementación que se propone.

1. A pesar de utilizar MapServer para la gestión de la cartografía el cliente no depende de la configuración del mismo, solo de las funcionalidades que se brindan en la interfaz.
2. Es posible definir las diferentes capas del mapa desde el punto de vista de clases permitiendo la manipulación de la cartografía aplicando para esto el paradigma de la programación orientada a objetos.
3. Es posible definir las diferentes capas del mapa a partir de un esquema de base de datos permitiendo la manipulación de la cartografía a partir de álgebra relacional y la lógica relacional.
4. Los resultados a partir de la obtención de las capas del mapa, tanto a nivel de clases como a nivel de esquema de base de datos permite a Hibernate, como motor de persistencia, la explotación del recurso de manera eficiente, obteniéndose mejores resultados y disminuyendo los tiempos de mantenimiento de la aplicación por conceptos de incorporación de funcionalidades a la interfaz de servicios cartográficos.
5. Los resultados obtenidos a partir de la interfaz de servicios estarán en función de la calidad de la cartografía y del nivel de detalle de cada una de las capas.

MÉTODO CIENTÍFICO

Para el desarrollo de esta tesis se han empleado los siguientes métodos de investigación científica:

1. Análisis y síntesis: Este método fue utilizado para analizar la situación problemática y determinar posibles variantes de solución.
2. Hipotético – deductivo: Permitted, a partir de la hipótesis, elaborar conclusiones acerca de la factibilidad de la utilización de una interfaz de servicios para la manipulación de mapas digitales que brinde la posibilidad de integrar los datos de las diferentes aplicaciones que conviven en una comunidad de desarrollo de software, sin tener conocimientos de la Geomática y tecnologías asociadas.
3. Criterio de experto: Para asumir determinados criterios que son imprescindibles para realizar el desarrollo del sistema.

3. Es posible definir las diferentes capas del mapa a partir de un esquema de base de datos permitiendo la manipulación de la cartografía a partir de álgebra relacional y la lógica relacional.
4. Los resultados a partir de la obtención de las capas del mapa, tanto a nivel de clases como a nivel de esquema de base de datos permite a Hibernate, como motor de persistencia, la explotación del recurso de manera eficiente, obteniéndose mejores resultados y disminuyendo los tiempos de mantenimiento de la aplicación por conceptos de incorporación de funcionalidades a la interfaz de servicios cartográficos.
5. Los resultados obtenidos a partir de la interfaz de servicios estarán en función de la calidad de la cartografía y del nivel de detalle de cada una de las capas.

MÉTODO CIENTÍFICO

Para el desarrollo de esta tesis se han empleado los siguientes métodos de investigación científica:

1. Análisis y síntesis: Este método fue utilizado para analizar la situación problemática y determinar posibles variantes de solución.
2. Hipotético – deductivo: Permitted, a partir de la hipótesis, elaborar conclusiones acerca de la factibilidad de la utilización de una interfaz de servicios para la manipulación de mapas digitales que brinde la posibilidad integrar los datos de las diferentes aplicaciones que conviven en una comunidad de desarrollo de software, sin tener conocimientos de la Geomática y tecnologías asociadas.
3. Criterio de experto: Para asumir determinados criterios que son imprescindibles para realizar el desarrollo del sistema.

CAPÍTULO 1 - FUNDAMENTACIÓN TEÓRICA.

La Información Geográfica y sus tecnologías asociadas se han convertido en uno de los pilares fundamentales en las Sociedades de la Información. [9]

Una proporción significativa de toda la información que está públicamente disponible en cualquier economía está relacionada con Información Geográfica, la cual es usada en un amplio rango de actividades económicas, sociales, culturales y comerciales.

1.1 INTRODUCCIÓN.

Las iniciativas de informatización global, regional, nacional y local, aplicada a diversos escenarios tales como gobierno electrónico, manejo de desastres, cuidado de la salud de los ciudadanos, entre otros, se soportan en la actualidad sobre Infraestructuras de Datos Espaciales. [7]

A nivel internacional existe desde 1994 el Open Geospatial Consortium u Open GIS Consortium (OGC), proyecto que agrupa a más de 250 organizaciones públicas y privadas, cuya finalidad es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica. La misma persigue acuerdos entre las diferentes empresas del sector que posibiliten la interoperación de sus sistemas de geoprocésamiento y facilitar el intercambio de la información geográfica en beneficio de los usuarios.

En el marco del Programa de Informatización de la Sociedad Cubana, se ha venido trabajando desde hace algunos años por la integración de los servicios de información geográfica al resto de los servicios informáticos destinados a este fin.

Uno de los ejemplos más ilustrativos es la experiencia de un Sistema de Información Territorial desarrollado en Manicaragua que ha tenido un impacto importante en el proceso de toma de decisiones del gobierno municipal.

En la actualidad, la iniciativa cubana para la creación de la Infraestructura Nacional de Datos Espaciales, denominada Infraestructura de Datos Espaciales de la República de Cuba, según el Acuerdo del Comité Ejecutivo del Consejo de Ministros No 5535, está siendo respaldada por el Ministerio de la

Informática y las Comunicaciones como parte de su Estrategia de Informatización de la Sociedad.

El proyecto de informatización de la Universidad de las Ciencias Informáticas (UCI), que se quiere convertir en prototipo de ciudad universitaria digital y de sociedad de información en Cuba, también ha identificado la importancia de los Servicios de Información Geográfica. [7]

Los servicios de información geográfica en este sentido posibilitarán a las aplicaciones disminuir los tiempos de desarrollo, debido a que centraran todos sus recursos humanos y materiales en la solución de su problema concreto y no en aras de resolver funcionalidades de sistemas que pueden ser resueltas a partir del consumo de un servicio.

Según [11], los Web Services representan los principios sobre los cuales se construirá la próxima generación de arquitecturas de e-business, presentando una evolución desde sistemas orientados a objetos en sistemas basados en servicios.

En este sentido es importante destacar que la solidez de esta tecnología radica en la sencillez de los formatos que envían a través de la red.

Debido a lo anterior y para un desarrollo de servicios Web que manipulen la cartografía y permitan la integración de los datos de diferentes aplicaciones con la componente espacial, es necesario acudir a la explotación de dicha tecnología de una manera adecuada.

A continuación se realiza un análisis a partir de los principales servidores de mapas OpenSource existentes, donde se plantean las fortalezas y debilidades de cada uno de ellos.

1.2 ESTADO DEL ARTE SERVIDORES DE MAPAS OPENSOURCE

La gestión de la cartografía desde el punto de vista informático es una de las tareas que conlleva una organización adecuada para garantizar un rendimiento óptimo de los recursos.

Son ya numerosos los países que han emprendido iniciativas en este sentido. Las orientaciones que se han adoptado desde el punto de vista de la gestión de dichas iniciativas, son necesariamente dependientes de circunstancias tales

como la situación de partida, la organización y ordenamiento territorial de los diferentes niveles de las administraciones públicas y muchos otros que, lógicamente, son muy variables en cada caso. [7]

La extensión en el uso de los sistemas SIG en organizaciones de todo tipo, ha llevado asociada la digitalización de una buena parte de la información espacial previamente disponible en soportes convencionales. Si unimos a esto las mayores posibilidades actuales para la adquisición de información espacial de prácticamente cualquier tipo imaginable, encontramos un escenario que, desde el punto de vista de la oferta, y especialmente en lo que se refiere a su cantidad, puede calificarse como de explosivo.

Sin embargo la implantación de sistemas de información geográfica en las organizaciones sigue siendo una actividad con un coste y un riesgo de no alcanzar plenamente los objetivos comparativamente altos con los correspondientes a sistemas de información más convencionales. Existe un convencimiento generalizado de que el principal obstáculo, aunque no el único, para la plena realización de los beneficios que se esperan de los SIG, y en general del uso de la información espacial, reside en las actuales dificultades para el intercambio y circulación de la información espacial entre organizaciones. [7]

Es importante destacar que en un entorno de interoperabilidad y estándares, tampoco hay que preocuparse por el software que se esté utilizando. Es posible, también, servir mapas en Web gracias a la filosofía adoptada basada en Metadatos y Catálogos de datos geográficos. Con todas estas características presentes, la tendencia apunta a la integración de las tecnologías espaciales con el resto de las Tecnologías de la Información y las Comunicaciones, lo cual es particularmente apreciado en el manejo de recursos de empresas. [7]

En este sentido a partir del desarrollo del software libre existen diferentes iniciativas que se encargan de desarrollar sus propias soluciones, así como varias comunidades con las cuales estas aplicaciones se van perfeccionando gradualmente.

Los servidores de mapas OpenSource forman parte de esta iniciativa y debido a esto existen comunidades que se dedican al mantenimiento de los mismos logrando diferentes mejoras en este sentido las cuales se traducen en un mejor aprovechamiento de los recursos en aras de mejorar los servicios al cliente.

A continuación se describen algunos de los servidores de mapas OpenSource.

UMN MapServer [38]

Inicialmente consistía en un conjunto de *script* que se utilizaban en ArcInfo para la manipulación de la cartografía de manera dinámica.

MapServer presenta dos soluciones:

- Como ejecutable CGI (Common Gateway Interface), la cual a pesar de ser el uso más común generaba diferentes problemas de seguridad debido a la tecnología que utiliza y la imposibilidad de utilizar tecnología multihilos, mucho más eficiente que la anterior.
- Como biblioteca, la cual es utilizada para la solución de tareas de alto contenido dinámico y consultas a la medida del usuario.

Características:

- Sencillez de configuración y administración. Toda la configuración se encuentra en el MapFile.
- Multiplataforma
- Velocidad de acceso a los datos
- Gran soporte de formatos soportados, tanto vectoriales como ráster.
- Implementa diferentes especificaciones de OGC (WMS, WFS (Básico), WCS, SOS (experimental), WMContext, SLD, GML2, FE).
- Utiliza GDAL, OGR y PROJ

Deegree [33]

Características:

- Se destaca por un número elevado de especificaciones de la OGC, entre las que se destaca WCS, WFC, WCS, CSW, WPS, SOS, entre otras.

- Presenta una elevada capacidad de configuración, lo que conlleva a que la misma sea compleja y poco amistosa. Requiere la creación de los ficheros de configuración.
- Requiere para un adecuado funcionamiento de un considerable número de ficheros y tecnologías.

GeoServer [34]

Características:

- Utiliza para su implementación de la biblioteca GeoTools, destinada a la manipulación de la información espacial.
- Administración en entorno Web. Presenta una interfaz Web que auxilia al usuario en la gestión de la configuración.
- Bases de datos espaciales (PostGis, Oracle, ArcSDE, DB2 y shapefile)
- Soporta la implementación de los protocolos WMS, WFS, WCS (Beta), SLD, GML y FE.
- Permite servir mapas en formato SVG
- Para su funcionamiento requiere el JDK 1.4 o superior y el JAI

MapGuide Open Source [36]

Recientemente liberado por la empresa Autodesk. Dispone de un sistema de publicación Web lo que implica una fácil administración y configuración, sin embargo, en estos momentos no se tienen experiencias de la explotación del mismo en el ámbito del software libre

Según [23] la evolución y utilización de esta herramienta es difícil de prever a pesar de contar con el respaldo de AutoDesk.

gvSIG [35]

Características:

- Producto muy terminado y orientado al usuario final, tanto a nivel de interfaz de usuario como de funciones implementadas.
- Soporte para los formatos más populares tanto vectoriales como de imágenes.

- Funcionalidades previstas muy completas.
- No soporta edición de datos vectoriales.
- No permite enlazar tablas (JOIN).

La tabla 1 muestra algunos de los resultados comparativos entre servidores de mapas OpenSource y en el anexo 1 se presentan los resultados de la benchmark de soluciones cliente/servidor en software libre realizadas a dos de estos servidores de mapas. [13]

Tabla 1– Tabla comparativa entre servidores de mapas OpenSource.

| | UMN MapServer | Deegree | GeoServer | gvSIG |
|--|------------------|--------------------------------------|-----------|-------|
| Implementación de estándares de la OGC | Si | Si | Si | Si |
| Uso de CGI | Si | No | No | No |
| Uso de SOAP | No | Si | Si | Si |
| Nivel de Utilización | Más utilizado | Ver nivel de utilización en figura 1 | | |
| Nivel de explotación | Baja | Alta | Media | Baja |
| Dependencia de ficheros | Baja | Alta | Media | Baja |

1.3 MAPSERVER: EL SERVIDOR DE MAPAS SELECCIONADO.

Según [23], en la figura se muestra la presencia en internet de algunos de los servidores de mapas analizados; así como de las aplicaciones relacionadas con estas tecnologías.

Dicho estudio muestra como resultado la utilización del servidor de mapas MapServer, inicialmente creado por la Universidad de Minnesota y en este momento mantenido por el proyecto TerraSip, patrocinado por la NASA. [10]

MapServer, como servidor de mapas ha demostrado a partir de estadísticas y estudios realizados que el mismo ofrece mejores rendimientos y prestaciones, con respecto a otros servidores de mapas OpenSource.

Presencia en buscadores

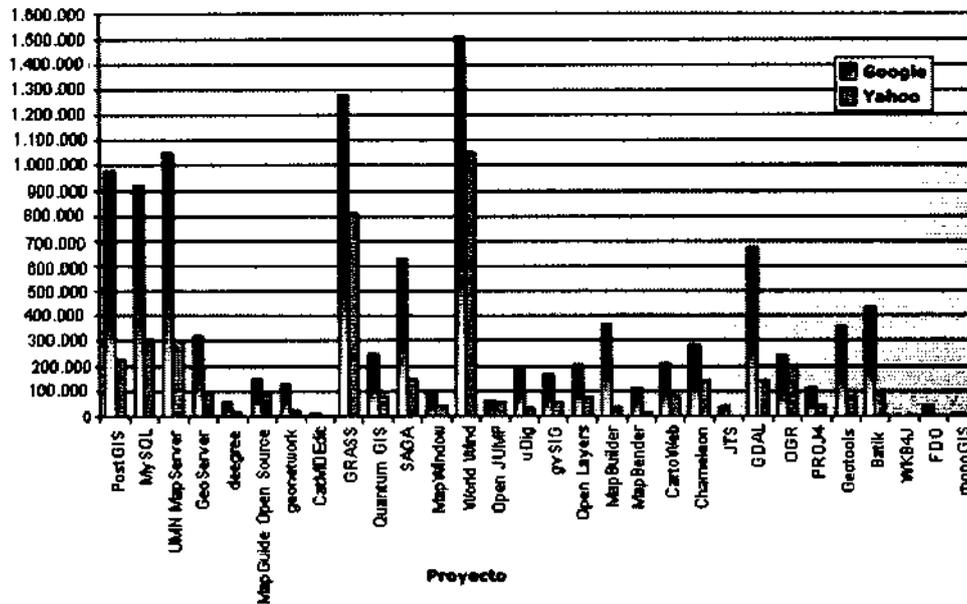


Figura 1– Presencia de los servidores de mapas en internet. (Fuente: 23)

A pesar de que este servidor de mapas provee clases para la manipulación de la cartografía existen ocasiones para la cual resulta complicada su utilización para el desarrollo.

De manera básica y para una implementación rápida MapServer generalmente utiliza para la interoperabilidad los estándares definidos por OGC. Específicamente MapServer utiliza el servicio de mapas (WMS), y el destinado a la interoperabilidad entre interfaces (WFS), sin embargo, la implementación de estos estándares se basan en tecnología CGI, que tiene el inconveniente de no tener un rendimiento adecuado en comparación con otras tecnologías para concurrencias de servicios.

Básicamente la tarea fundamental de la gestión de la cartografía consiste en proveer a los sistemas informáticos de algoritmos y métodos capaces de mostrar la cartografía de un territorio de forma total o parcial y que permita

interactuar de forma dinámica con la misma, incorporando para esto los datos asociados a cada elemento catastral.

MapServer, como servidor de mapas provee funcionalidades para la manipulación de la cartografía, sin embargo la configuración de la misma y la funcionalidades que brinda, en ocasiones, asume un conocimiento para aquellos usuarios que utilicen este servicio que en muchas ocasiones conlleva a no obtener los resultados esperados.

La posibilidad de manipular una cartografía a partir de las clases es otra posibilidad que provee MapServer, pero esta solución elimina la posibilidad de la interoperabilidad de mapas.

MapServer inicialmente puede utilizarse para manipular la cartografía de manera secuencial, directamente con archivos en formato de ESRI (shape), sin embargo, se obtiene un rendimiento significativamente superior cuando se utiliza en combinación con gestores de bases de datos, los cuales en algunos casos requieren de asimilar tipos de datos espaciales como es el caso de PostgreSQL/POSTGIS.

1.4 LAS BASES DE DATOS ESPACIALES

Un aspecto clave en una Infraestructura de Datos Espaciales es proveer mecanismos que permitan buscar, recuperar, compartir e integrar datos espaciales. [18]

La construcción de una base de datos geográfica implica un proceso de abstracción para pasar de la complejidad del mundo real a una representación simplificada que pueda ser procesada por el lenguaje de las computadoras actuales. Este proceso de abstracción tiene diversos niveles y normalmente comienza con la concepción de la estructura de la base de datos, generalmente en capas; en esta fase, y dependiendo de la utilidad que se vaya a dar a la información a compilar, se seleccionan las capas temáticas a incluir. [19]

En este sentido, las bases de datos se han convertido en el instrumento estratégico al constituir la base de los sistemas de información y soportar la gestión y la toma de decisiones a nivel mundial.

Según CATTELL (1991), las bases de datos deben caracterizar por “proporcionar capacidades de gestión de datos, permitiendo que grandes cantidades de datos persistentes sean compartidos por muchos usuarios.

Desde este punto de vista las mismas proporcionan mecanismos para la gestión de objetos, permitiendo tipos de datos mucho más complejos, objetos multimedia, datos derivados, encapsulamiento de la semántica de los datos, así como otras nuevas capacidades.

Algunos proporcionan incluso gestión de conocimiento, soportando un gran número de reglas complejas para inferencia automática de información y también para mantener las restricciones de integridad entre datos.

En la mayoría de los casos los Sistemas de Gestión de Base de Datos (SGBD), solucionan los problemas a los cuales nos enfrentamos, sin embargo, en un inicio, los mismos descuidaban la información espacial. A raíz de esto se ha incorporado hoy en día interfaces que proveen una solución de lo descrito anteriormente, permitiendo incorporar de manera sencilla la componente espacial dentro de la toma de decisiones a cualquier nivel organizacional.

Oracle, líder en este campo, es un ejemplo de lo anteriormente descrito. Dicho SGBD, provee internamente a sus bases de datos de la información espacial.

Por otro lado, PostgreSQL, el cual va en ascenso, es el Sistema Administrador de bases de Datos de código abierto que posibilitó el desarrollo de soluciones corporativas con una mejor relación costo/beneficios. Un punto fuerte de este SGBD es su capacidad de tratar grandes volúmenes de datos con escalabilidad, o sea, su arquitectura puede ser continuamente ampliada de acuerdo con la demanda de los usuarios.

De acuerdo con [39], exactamente en este contexto, entran las aplicaciones en el área de Geotecnologías que necesitan de una infraestructura robusta y en continua expansión.

Retomando la política de software libre, definida por el Ministerio de la Informática y las Comunicaciones, PostGIS incorpora a PostgreSQL la definición de tipos de datos geométricos lo que conlleva evidentemente a brindar al usuario la posibilidad de gestionar información relacionada con la geometría de estos objetos.

1.5 DATOS ESPACIALES Y FRAMEWORK PARA PERSISTENCIA.

En consecuencia con [32], una base de datos geográfica *“representa el reemplazo digital para los mapas y reportes”* y además *“pueden ser mucho más útiles que sus predecesores (papel) por todas las ventajas que una base de datos digital posee en sus diversos campos de aplicación”*.

A partir de las nuevas arquitecturas de desarrollo, para obtener mejores rendimientos, se plantea la separación de las funcionalidades de las aplicaciones de los datos asociados a la misma. Para esto debe utilizarse un framework de persistencia que se encargue de toda la gestión de los datos relacionado con la explotación de los sistemas informáticos.

Existen diversos frameworks o capas para permitir un cómodo desarrollo en cualquier tipo de aplicaciones, de manera que pueda determinarse las diferentes responsabilidades de cada una de ellas.

El desarrollo de aplicaciones utilizando una arquitectura de n capas se posibilita un fácil y adecuado mantenimiento de las aplicaciones. De todas las capas de desarrollo que se implementan en una aplicación, la capa de persistencia es la que tiene una relación más directa con los datos, ya que es la encargada del almacenamiento, recuperación y actualización de los mismos.

Los datos espaciales posibilitan incorporar a los gestores de bases de datos de información geoespacial, la cual está estrechamente relacionada con la cartografía. Estos datos espaciales son la clave para permitir a los sistemas de información geográfica proveer la información relacionada con una empresa, región o país de una componente geoespacial, facilitando mejorar los criterios para la toma de decisiones a cualquier nivel de dirección.

1.6 ISO TC/211

El Comité Técnico 211 de la ISO (ISO/TC 211) es el encargado de desarrollar estándares globales en el campo de la información geográfica y la Geomática. En la actualidad ISO TC/211 está trabajando en más de 30 estándares, algunos de los cuales ya se encuentran disponibles [30].

La importancia de la definición de estos estándares radica en obtener mejores resultados, basado en la organización de la información a nivel internacional de

manera adecuada, que posibilite la incorporación de aplicaciones como es el caso de la Interfaz de servicios para la solución a la consulta. Utilización y el almacenamiento de los datos correspondientes a las diferentes temáticas de una región determinada.

De acuerdo a [6], la misión principal de la ISO/TC 211 es establecer un conjunto de estándares para la adquisición, manipulación, administración, análisis, acceso, presentación y transferencia de la información entre diferentes usuarios y sistemas sin que esto dependa de su ubicación geográfica.

Sin embargo [5] plantea que *“no bastan los estándares concertados, se requieren herramientas para instrumentarlos y sobretodo educación y entrenamiento en la colección, administración y uso de los datos espaciales para remover las barreras”*. En este sentido es importante destacar que la adecuada explotación de los recursos materiales y humanos juega un papel fundamental en los resultados que se obtendrán.

1.7 OPENGIS®

OGC es una asociación no lucrativa para promover nuevos enfoques técnicos y comerciales para geoprocesamiento interoperable [27]. Su visión se puede enunciar como una infraestructura nacional y global en la cual los datos geoespaciales y los recursos de geoprocesamiento se mueven libremente, totalmente integrados con las últimas tecnologías de computación distribuida y accesible a todo el mundo.

Si bien ISO/TC 211 se orienta hacia estándares abstractos de información geográfica, OGC aporta especificaciones de implementación, lo cual constituye una de las claves de su popularidad. Por otra parte, está integrado por los líderes mundiales de tecnología de información y de las tecnologías geoespaciales, lo cual asegura su implementación en la industria. [8]

Algunos de los principales estándares relevantes al tema de las Infraestructuras de Datos Espaciales que se han liberado por OpenGIS son los siguientes:

1. OpenGIS® Especificaciones de Interface de Servidor Web de Mapas WMS [25].

2. OpenGIS® Especificaciones del Servicio de Catálogo [26].
3. OpenGIS® Especificaciones de Servicio Web de Coberturas WCS [28].
4. OpenGIS® Especificaciones de Servicio Web de Elementos WFS [29].

Es este sentido la abstracción de los estándares que se menciona anteriormente, los cuales, tienen una alta relación con las especificaciones de implementación, en ocasiones, puede repercutir negativamente en el desarrollo de un proyecto en el cual no exista un conocimiento de las tecnologías relacionadas con las temáticas relacionadas con la Geoinformática.

Debido a lo anterior se observa la necesidad de trabajar desde el punto de vista de la especialización de tecnologías, logrando la colaboración y explotación adecuada de cada una de ellas sin la necesidad actual de utilizar recursos de bajo rendimiento o detener el flujo de desarrollo de un proyecto, conllevando a la reorientación de recursos humanos y materiales en función de aspectos secundarios del proyecto que pueden ser perfectamente resueltos por terceros proyectos haciendo uso de los servicios informáticos y la interoperabilidad entre sistemas.

1.8 WEB MAP SERVICE: EL SERVICIO BÁSICO DE LOS SERVIDORES DE MAPAS.

La especificación Web Map Server (**WMS**) o Servicios de Mapas Web es un estándar definido por el Open Geospatial Consortium que brinda una interfaz común para consultar y acceder a capas de mapas en un Servidor de Mapas. Los clientes y servidores que se adhieren al estándar OGC/WMS pueden comunicarse entre sí, sin importar la arquitectura o implementación particular de cada uno.

WMS ha sido desarrollado **Open Geospatial Consortium (OGC)**. El software cliente obtiene del servidor imágenes y código XML de georeferenciación, como resultado de los criterios enviados al servidor en las peticiones de acuerdo al estándar.

Un servicio **WMS** brinda los siguientes servicios:

1. *GetCapabilities* (Información del servidor)
2. *GetMap* (Generación de imágenes de mapa)

3. *GetFeatureInfo* (Consulta de atributos, text/plain y text/html únicamente)

1.9 ARQUITECTURA ORIENTADA A SERVICIOS

Las tecnologías analizadas anteriormente requieren trabajar coordinadamente; con el objetivo de alcanzar los resultados esperados. En este sentido es posible la utilización de una arquitectura orientada a servicios (SOA). Según [1], "SOA no es una nueva tecnología sino la evolución de conceptos y tecnologías existentes, soportadas por el uso de estándares abiertos", cuyo objetivo principal radica en "la reutilización de los activos y servicios existentes, y un enfoque basado en estándares e interoperabilidad".

La cartografía en WEB (WebMapping), sienta las bases necesarias para el establecimiento efectivo de un entorno en el que es factible el intercambio, de información geográfica y servicios de geoprosesos, muy en la línea de las visiones actuales sobre el mercado de la información y el desarrollo del sector de las industrias de la información. [12]

El uso de estándares abiertos basados en SOA, garantizará la comunicación eficiente entre los diferentes sistemas de la comunidad sin que esto conlleve a la dependencia entre sistemas y solamente estará en función de los servicios que cada aplicación sea capaz de proveer, utilizando las tecnologías adecuadas para esto.

Desde el punto de vista práctico de la utilización de una arquitectura orientada a servicios en entornos Geoespaciales, la principal ventaja radica en la evidente independencia de los sistemas a utilizar los servicios expuestos sin que esto conlleve a analizar especificidades internas de las tecnologías de manipulación de mapas existentes o utilizadas en el desarrollo de la interfaz, permitiendo incorporar un valor añadido a las aplicaciones que utilicen estos servicios.

1.10 CONCLUSIONES DEL CAPITULO 1

Los servidores de mapas estudiados son todos compatibles con OGC los que permiten la interoperabilidad

Desde el punto de vista de la configuración existen servidores donde la complejidad de la configuración al respecto de su implantación, así como, otros

servidores donde la configuración es más simple o tienen una interfaz que auxilia a la configuración del mismo.

Muchos de ellos a pesar de ser aplicaciones OpenSource, no soportan una interfaz que, de manera cómoda, permita publicar servicios de manipulación de la cartografía a otras aplicaciones para la integración de los datos y los mapas que se manipulan en una región determinada y que no dependan de la configuración del servidor por parte del cliente que los utilice.

Para el caso de las aplicaciones que para la comunicación utilizan la tecnología CGI, el rendimiento de estas disminuye considerablemente. En este caso, ante la posibilidad de peticiones concurrentes, el rendimiento de los servidores disminuiría debido a que la tecnología utilizada para ellos no brinda la posibilidad del trabajo con multihilos y cada petición la gestionaría sin utilizar las peticiones del mismo tipo que se encuentran corriendo en tiempo real.

A partir de la literatura consultada se observa la difusión que tiene MapServer con respecto al resto de los servidores de mapas, la cual está dada por el rendimiento y flexibilidad que el mismo presenta al poder utilizar librerías para el desarrollo de aplicaciones a la medida.

A pesar de su éxito, es posible elevar el rendimiento de MapServer, utilizando como variante la biblioteca que el mismo brinda, construyendo servicios basados en tecnología ISAPI y utilizando una arquitectura orientada a servicios, mejorando los tiempos de respuesta y elevando el rendimiento de dicho servidor de mapas.

CAPÍTULO 2 – DESCRIPCIÓN DE LA PROPUESTA

2.1 CARACTERIZACIÓN DEL FENÓMENO

En el capítulo 1 se abordaron los servidores de mapas, los contextos donde se aplican y la necesidad de aplicaciones que permitan integrar los datos con la información cartográfica de una región específica de manera tal que las comunidades de desarrolladores puedan utilizar estos servicios sin la necesidad de tener un conocimiento adecuado de la geomática.

Desde este punto de vista y para el desarrollo de lo anterior, es necesario remitirse a la arquitectura propuesta por los estándares OpenSource que se dedican a esta tarea, así como de las arquitecturas y tecnologías existentes a nivel mundial.

2.2 ARQUITECTURA MULTICAPAS

La programación en múltiples capas es la técnica más efectiva en aplicaciones empresariales, debido a la fácil administración que implica el dividir los componentes de la aplicación en capas y la rapidez que esto implica en programas orientados a Cliente-Servidor. Esta arquitectura consiste en dividir los componentes primarios de la aplicación.

De esta manera es posible separar las diferentes responsabilidades de una aplicación, obteniendo mejores resultados, tanto de rendimiento de la aplicación, como de mantenimiento de la misma.

El paradigma de desarrollo de aplicaciones en capas, se ha desarrollado con el transcurso de los años, permitiendo en la actualidad lograr una escalabilidad en aplicaciones que permiten la explotación de los sistemas en diferentes dispositivos de salida.

A continuación se muestra una tabla comparativa donde se evidencia las ventajas que conlleva el adecuado uso de esta arquitectura.

Tabla 2 – Característica 2-tier n-tier. (Fuente: [15])

| Característica | 2 capas | n capas |
|--------------------------------------|--|---|
| Administración de sistemas | Mayor complejidad por componente | Menor complejidad por componente, mayor complejidad de conjunto (necesidad de Middleware) |
| Seguridad | Enfocado al dato | Enfocado al servicio u objeto |
| Encapsulamiento de datos | Tablas expuestas, código incrustado | Invocación de servicios u objetos |
| Rendimiento | Pobre: Utilización de SQL sobre la red | Óptimo: el middleware optimiza el tráfico de red, balanceo de carga u uso racional de los recursos del servidor y los datos |
| Escalabilidad | Número de clientes limitados | Múltiples llamadas, distribución de carga y servicios. Posibilidad la utilización de múltiples servidores. |
| Reutilización de aplicaciones | Aplicaciones monolíticas | Servicios y objetos. |

De manera general las arquitecturas multicapas, proporcionan una gran cantidad de beneficios para las empresas que necesitan soluciones flexibles y fiables para resolver complejos problemas inmersos en cambios constantes permitiendo a todas las aplicaciones basadas en esta tecnología la posibilidad de trabajar con clientes ligeros, tal como navegadores de Internet, WebTV, Teléfonos Inteligentes, PDAs (Personal Digital Assistants o Asistentes Personales Digitales) y muchos otros dispositivos preparados para conectarse a Internet.

La arquitectura de los servidores de mapas es de tipo cliente – servidor. En este caso el cliente solicita los recursos del servidor y el mismo gestiona todas las peticiones y responde de manera ordenada a éstas.

En la figura 2 se muestra la arquitectura de servidores mapas. En este caso se puede observar que es una arquitectura cliente – servidor, a la cual se le incorpora un componente adicional que es el encargado de devolver al cliente una imagen georeferenciada de una solicitud hecha a una cartografía, la cual puede tener incorporado los datos de diferentes temáticas, relacionados con la región en cuestión. En nuestro caso específico se utilizará MapServer para realizar dicho trabajo, debido al rendimiento y características que el mismo brinda.

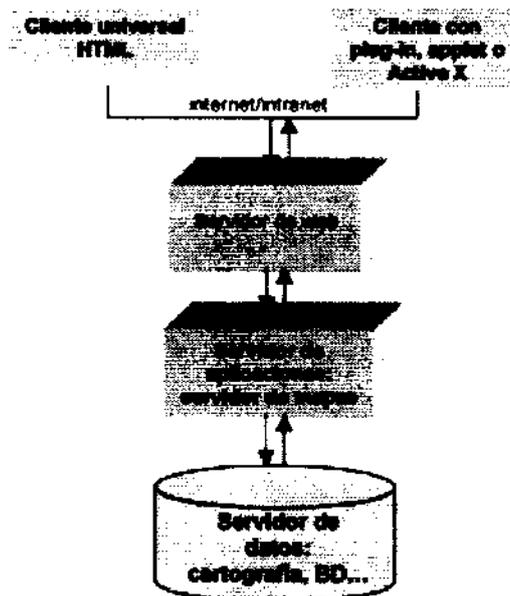


Figura 2 – Arquitectura básica de servidores de Mapas (Fuente: [9])

Para realizar alguna modificación es necesario es necesario analizar la arquitectura interna un Servidor de Mapas. En este caso se requiere describir dicha arquitectura a partir de MapServer.

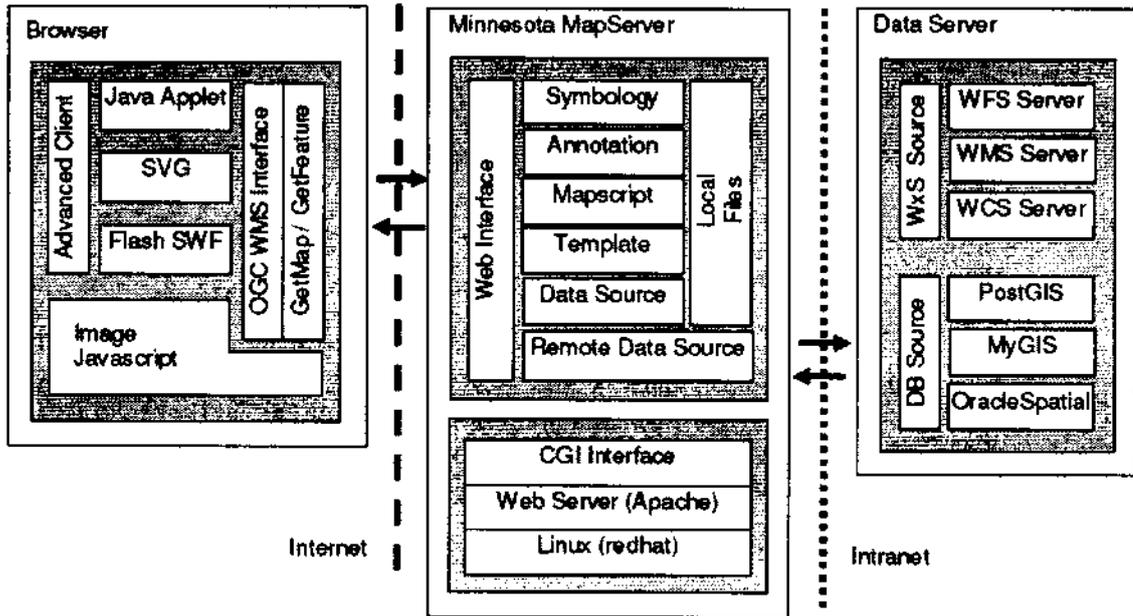


Figura 3 – Arquitectura interna de los servidores de Mapas OpenSource. (Fuente: [40])

Como se observa en la figura que caracteriza la arquitectura interna de los servidores OpenSource, los estándares mínimos que se incorporan son los relacionados con el WMS, WFS y WCS debido a que son los más utilizados para las tareas más cotidianas.

En el anexo se muestra los componentes de UMN MapServer, como servidor seleccionado para el desarrollo de la interfaz.

2.3 MODELO DE REFERENCIA

2.3.1 ARQUITECTURA PROPUESTA

Dicha arquitectura se encuentra separada por capas, de manera tal que permite realizar modificaciones de manera paralela sin la necesidad de realizar grandes modificaciones desde el punto de vista de la implementación de los programadores.

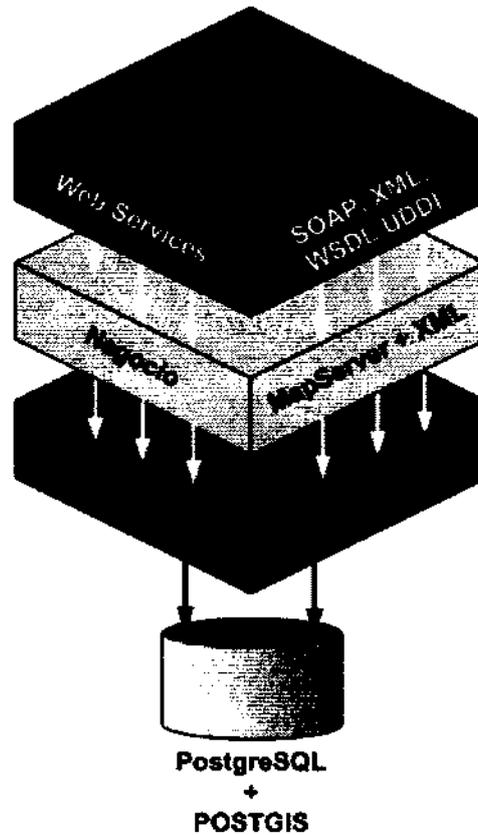


Figura 4 – Arquitectura para interfaz de servicios cartográficos.

Como se observa en la figura anterior, se presenta una división entre capas, en la cual existe una definición muy clara de las responsabilidades a nivel de software de cada una de ellas, permitiendo el mantenimiento del código y la incorporación de nuevas funcionalidades, sin necesidad de que para esto sea necesario realizar modificaciones en todo el programa.

De esta manera estas modificaciones o actualizaciones de código se encuentran bien localizadas permitiendo ganar en eficiencia desde el punto de vista de la producción debido a que es posible realizar modificaciones de forma paralela disminuyendo significativamente los tiempos, lo que significa un mayor tiempo de prestación de servicios a los clientes que utilicen la plataforma.

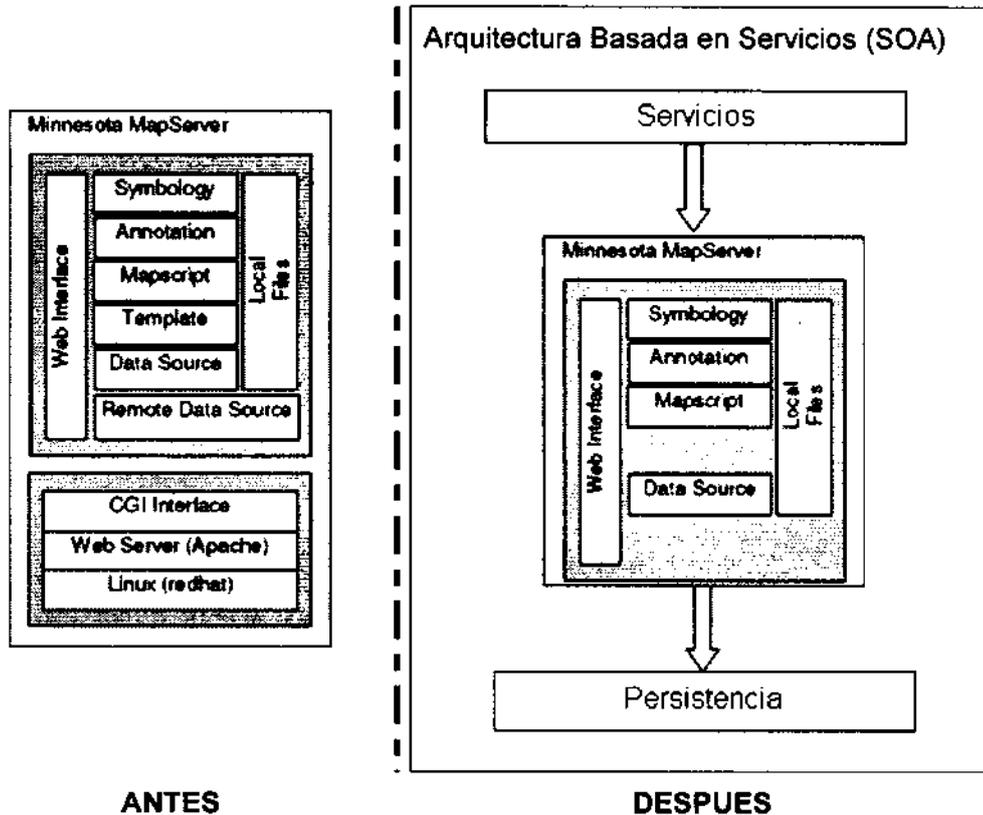


Figura 5 – Representación características que se incorporan al servidor de mapas.

Las incorporaciones que se le realizan a MapServer implican que es posible realizar diferentes operaciones aplicando técnicas más eficientes, sin la necesidad de sobrecargar de responsabilidades al mismo.

De esta manera MapServer solo tendría la responsabilidad de representar de manera georeferenciada la cartografía de manera total o parcial con la información relevante que desee el usuario que utilice el servicio.

Desde el punto de vista de la utilización de servicios esta tecnología está comenzando a utilizarse con mayor auge a partir de utilizar recursos de terceros para dar solución a los problemas concretos del desarrollo.

Dentro de la arquitectura orientada a servicios, la capa de servicios será aquella que tendrá interacción con las diferentes aplicaciones de la comunidad de desarrolladores. A partir de los servicios publicados en esta capa los clientes ejecutan una petición la cual es analizada en la capa de negocio de la misma. Según sea el caso del servicio que se solicite, la capa de persistencia puede colaborar con MapServer, de manera tal que ya el servidor de mapas no

tiene porque preocuparse por las conexiones, consultas y persistencia de los datos de manera general. Este enfoque, además permite actualizaciones claramente localizadas conllevando a disminuir significativamente los tiempos fuera de servicio de la interfaz.

La incorporación de la capa de persistencia, permite además la migración de las bases de datos sin que esto conlleve a problemas de explotación por parte de las comunidades que utilicen la interfaz.

A continuación se muestra la estructura de paquetes que dio solución a la propuesta planteada anteriormente.

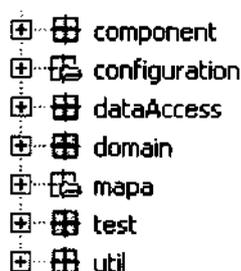


Figura 6 – Estructura de paquetes de la interfaz de servicios cartográficos.

A continuación se describen brevemente las responsabilidades de cada uno de los paquetes que se muestran en la figura anterior.

Paquete component

Este paquete brinda las funcionalidades que utilizará el usuario a través de Web Services. Provee todas las funciones que utilizará el cliente, desde el punto de vista de explotación. Dicho paquete esta compuesto por las siguientes clases:

- GISComponent.java: Provee todas las funcionalidades públicas que se brindan a partir de servicios Webs.
- GeomFunctions.java: Interfaz que provee las funciones geométricas que brinda POSTGIS
- PostgisFuntionsHibernate.java: Extiende las funcionalidades que brinda GeomFunctions.java. Configura y provee acceso a las sesiones que maneja Hibernate, así como a las funciones básicas de POSTGIS.

Paquete configuration

Este paquete contiene el fichero de configuración donde se encuentra información relevante que es utilizada por la interfaz de servicios.

Paquete dataAccess

Este paquete contiene las interfaces de cada capa de la base de datos. En este caso específico se caracterizan las capas del mapa de la UCI. Dicho paquete esta compuesto por los siguientes ficheros:

- **AreasDAO.java**: Clase Interfaz que define las operaciones que se realizarán a la tabla **Areas** de la base de datos espacial.
- **ContornosDAO.java**: Clase Interfaz que define las operaciones que se realizarán a la tabla **Contornos** de la base de datos espacial.
- **EdificiosDAO.java**: Clase Interfaz que define las operaciones que se realizarán a la tabla **Edificios** de la base de datos espacial.
- **ManzanasDAO.java**: Clase Interfaz que define las operaciones que se realizarán a la tabla **Manzanas** de la base de datos espacial.
- **SeparadoresDAO.java**: Clase Interfaz que define las operaciones que se realizarán a la tabla **Separadores** de la base de datos espacial.
- **VialesDAO.java**: Clase Interfaz que define las operaciones que se realizarán a la tabla **Viales** de la base de datos espacial.

Paquete DataAcces.impl

Este paquete contiene la implementación de las interfaces de cada capa de la base de datos. En este caso específico se caracterizan las capas del mapa de la UCI. Dicho paquete esta compuesto por los siguientes ficheros:

- **AreasDAOImpl.java**: Implementación de la interfaz **AreasDAO.java**
- **ContornosDAOImpl.java**: Implementación de la interfaz **ContornosDAO.java**
- **EdificiosDAOImpl.java**: Implementación de la interfaz **EdificiosDAO.java**
- **ManzanasDAOImpl.java**: Implementación de la interfaz **ManzanasDAO.java**

- **SeparadoresDAOImpl.java:** Implementación de la interfaz **SeparadoresDAOImpl.java**
- **VialesDAOImpl.java:** Implementación de la interfaz **VialesDAO.java**

Paquete **DataAcces.impl.map**

En este paquete se describen las diferentes capas de la cartografía a nivel de base de datos. De esta manera es posible describir la conformación de cada tabla de la base cartográfica, los atributos que tiene y la relación que existe entre las diferentes tablas de la base de datos geográfica.

- **Areas.hbm.xml:** XML que describe la composición de cada tipo de campo de la tabla **Areas**.
- **Contornos.hbm.xml:** XML que describe la composición de cada tipo de campo de la tabla **Contornos**.
- **Edificios.hbm.xml:** XML que describe la composición de cada tipo de campo de la tabla **Edificios**.
- **Manzanas.hbm.xml:** XML que describe la composición de cada tipo de campo de la tabla **Manzanas**.
- **Separadores.hbm.xml:** XML que describe la composición de cada tipo de campo de la tabla **Separadores**.
- **Viales.hbm.xml:** XML que describe la composición de cada tipo de campo de la tabla **Viales**.

Paquete **domain**

Este paquete contiene las clases de dominio que permite caracterizar cada una de las entidades generales del mapa. Dicho paquete esta compuesto por las siguientes clases:

- **Areas.java:** Clase que encapsula las características y funcionalidades de la capa **Areas** de la cartografía de la UCI.
- **Contornos.java:** Clase que encapsula las características y funcionalidades de la capa **Contornos** de la cartografía de la UCI.
- **Edificios.java:** Clase que encapsula las características y funcionalidades de la capa **Edificios** de la cartografía de la UCI.

- **Manzanas.java:** Clase que encapsula las características y funcionalidades de la capa **Manzanas** de la cartografía de la UCI.
- **Separadores.java:** Clase que encapsula las características y funcionalidades de la capa **Separadores** de la cartografía de la UCI.
- **Viales.java:** Clase que encapsula las características y funcionalidades de la capa **Viales** de la cartografía de la UCI.

Paquete mapa

Este paquete contiene el fichero de configuración de MapServer.

Paquete test

Paquete para la realización de pruebas para la comprobación de los resultados que se están obteniendo.

Paquete util

Este es un paquete de utilidades que permite leer los diferentes ficheros de configuración y además resuelve el problema de Hibernate para la persistencia de datos espaciales. Dicho paquete esta compuesto por las siguientes clases:

- **GeometryUserType.java:** Implementa la interfaz de Hibernate **UserType.java**, permitiendo que el mismo admita los tipos datos espaciales que brinda POSTGIS. Hibernate, como motor de persistencia, no conoce estos tipos de datos geométricos, por lo que no brinda un tipo de atributo para estos tipos de datos
- **ReadGeoConfig.java:** Permite la lectura de los ficheros de configuración.

2.3.1.1 DESCRIPCIÓN DE MÉTODOS Y TÉCNICAS UTILIZADAS

A continuación se describe las herramientas que se seleccionaron para dar solución al problema planteado; así como una fundamentación que respalda la selección realizada.

SERVIDOR DE BASE DE DATOS

A partir de la política trazada por el Ministerio de Informática y Comunicaciones (MIC), acerca de la utilización del software libre y a partir de los requerimientos originados por nuestro problema en cuestión y los resultados alcanzados por el

gestor de base de datos PostgreSQL, se decidió optar por la misma para el almacenamiento de la información cartográfica, para esto fue necesario incorporar al gestor de base de datos de soporte para estructura de datos espaciales, utilizando para esto POSTGIS.

POSTGRESQL Y POSTGIS

La integración de PostgreSQL y POSTGIS, incorpora a la base de datos la posibilidad de la definición de tipos de datos espaciales, permitiendo encriptar la información cartográfica de una región específica y además vinculando la información asociada con esta región de manera cómoda. Esta solución es adecuada para el trabajo que se propone, cumpliendo con las reglamentaciones del Ministerio de Informática y las Comunicaciones acerca de la política de utilización del Software Libre.

PostGIS es una extensión al sistema de base de datos objeto-relacional PostgreSQL. Los objetos que el mismo soporta se muestran a continuación:

- POINT(0 0 0)
- LINESTRING(0 0,1 1,1 2)
- POLYGON(((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0)),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))
- MULTIPOINT(0 0 0,1 2 1)
- MULTILINESTRING(((0 0 0,1 1 0,1 2 1)),(2 3 1,3 2 1,5 4 1))
- MULTIPOLYGON((((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0)),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0)),((-1 -1 0,-1 -2 0,-2 -2 0,-2 -1 0,-1 -1 0)))
- GEOMETRYCOLLECTION(POINT(2 3 9),LINESTRING((2 3 4,3 4 5))

A continuación se muestra un ejemplo de como insertar registros en una tabla en la cual se incorporan datos de GIS.

```
INSERT INTO CALLES_GEOM(ID,GEOM,NAME) VALUES (1,
GeometryFromText
('LINESTRING(191232 243118,191108 243242)',-1),'Jeff Rd');
```

Es importante destacar que para el caso de la capa de datos se utilizará, Hibernate. Esto permitirá comunicarse con la base de datos cartográfica

utilizando para esto conexiones persistentes. Con relación a lo anterior es necesario precisar que Hibernate carece de la definición de datos espaciales asignándole a los datos espaciales el tipo de datos **Serializable** lo que introduce errores en el mapeo objeto relacional. Para darle solución a lo anterior fue necesario extender las funcionalidades de la interfaz **UserType** que el mismo brinda. En el capítulo 3 se abordará con más detalle la solución que se implementó para esto.

CAPA DE DATOS

Se encarga de la manipulación de los datos y de toda la información cartográfica almacenada en la base de datos geográfica, es la responsable de la integridad y veracidad de los datos que se provee al cliente. En este caso estos clientes finales realizarán las peticiones necesarias a partir de **Web Services**.

Para la implementación de la capa de datos se utilizó Hibernate, como motor de persistencia, debido a que el mismo permite el almacenamiento, mantenimiento y eliminación de objetos en la base de datos.

Hibernate utiliza ficheros XML para la configuración de conexión, descripción y relación de los datos de cada una de las tablas de la base de datos.

En este caso se definieron clases que representan la estructura física de cada tabla, así como el esquema físico de cada tabla. El objetivo de lo anterior radica en realizar el mapeo objeto relacional que brinda Hibernate.

La ventaja de lo anterior posibilita utilizar las funcionalidades que se generan en cada clase permitiendo la manipulación con sesiones y la persistencia de nuevos datos, sin la necesidad de introducir trazas de SQL.

HIBERNATE

Hibernate es un entorno de trabajo que tiene como objetivo facilitar la persistencia de objetos en bases de datos relacionales y al mismo tiempo la consulta de estas bases de datos para obtener objetos. Para facilitar el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, se utilizan archivos declarativos de configuración basados en (XML) que permiten establecer estas relaciones.

Hibernate se adapta al proceso de desarrollo de software, sin importar si se parte de una base de datos en blanco o de una ya existente.

Características de Hibernate:

- Realiza el mapeo de clases en Java a tablas de una base de datos (y de tipos de datos de Java hacia tipos de datos de SQL)
- Genera también consulta de datos y facilidades de recuperación.
- Genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias,
- Mantiene la portabilidad entre todas las bases de datos con un ligero incremento en el tiempo de ejecución.
- Es una herramienta de Mapeo objeto-relacional para la plataforma Java (y disponible también para .NET)

Hibernate proporciona interfaces al programador que le permiten abstraerse de las tareas repetitivas como puede ser el abrir o cerrar una sesión para persistir un objeto en la base de datos y en consecuencia, reducir notablemente el código encargado de la persistencia.

A continuación se muestra la arquitectura interna de Hibernate:

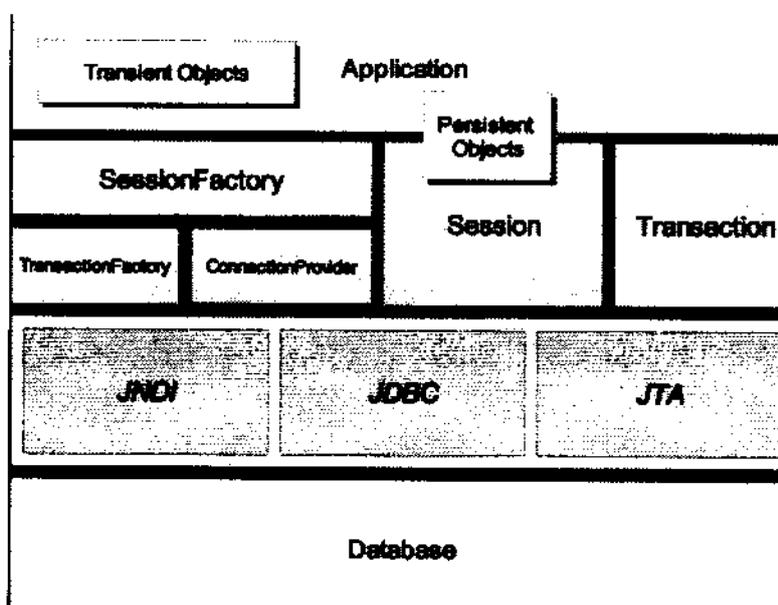


Figura 7– Arquitectura interna de Hibernate (Fuente: <http://hibernate.org>)

Como se muestra en el gráfico Hibernate se integra dentro de los servicios de una plataforma J2EE siendo capaz de obtener conexiones a través de objetos DataSource vía JNDI, ejecutar sus transacciones dentro de un entorno JTA, o permitiendo conexiones por JDBC.

CAPA DE LÓGICA DE NEGOCIO

En la misma se realiza toda la implementación de la lógica necesaria para los diferentes servicios que se proveerán de cara al cliente o aplicaciones clientes.

En este caso en cada una de las funcionalidades del servicio se define cuales son las tareas que necesita realizar MapServer como servidor de mapas y en cuales de esas operaciones el motor de persistencia Hibernate puede colaborar en función de balancear las responsabilidades de las tecnologías que conviven en la interfaz de servicios.

Para la implementación realizada en la capa de lógica de negocio se utilizó Java nativo, debido a que el mismo provee un conjunto de clases, las cuales pueden ser reutilizables para obtener los requerimientos que se solicitan.

LIBRERÍAS DE MAPSERVER PARA LA MANIPULACION DE MAPAS

La presentación de mapas en Internet, permite no sólo mostrar las características propias de un determinado fenómeno de manera estática, sino que además facilita la incorporación de variables dinámicas las cuales le asignan nuevas capacidades de comunicación de información a los mapas, permitiendo que la observación y análisis de los fenómenos sea más claro.

Existen estándares que definen las funcionalidades que deben tener los servidores de mapas de manera general.

Un servidor de Mapas puede hacer tres cosas:

- Producir un mapa (una serie de elementos gráficos o como un conjunto empaquetado de datos de características geográficas).
- Responder a preguntas básicas sobre el contenido del mapa
- Decirle a otros programas qué mapas puede producir y cuáles de ellos pueden ser cuestionados adicionalmente.

Como se analizó en el capítulo 1, MapServer brinda dos variantes para la manipulación de la cartografía. En este caso la utilización de las librerías que el mismo provee tiene una mayor utilidad y rendimiento, sobre todo cuando se requiere realizar aplicaciones a la medida para un número de usuarios considerables y con diferentes problemáticas.

MAPSERVER. EJEMPLO DE SERVIDOR DE MAPAS OPESOURCE.

MapServer, a pesar de ser una aplicación Common Gateway Interface (CGI), provee un conjunto de librerías que permite a los desarrolladores la manipulación de la cartografía. No obstante a lo anterior, esto requiere por parte de los programadores la especialización de estos conocimientos para la realización de tareas de este tipo.

En proyectos donde se requiere la utilización de servicios basados en cartografía, sería deseable poder contar con los mismos, sin que esto conlleve a la orientación de recursos para el desarrollo de lo anterior.

MapServer gestiona toda la configuración de la cartografía a través de un fichero de configuración (MapFile), propio de su arquitectura, el cual define, todos los protocolos de comunicación y de formato de salida que tendrá la cartografía, de cara al cliente.

Debido a lo anterior y haciendo uso de las librerías que provee MapServer, y a partir del estudio del rendimiento que se obtiene con el mismo, se decidió seleccionar al mismo para darle solución a los problemas planteados anteriormente.

La diferencia de esta librería con respecto a la versión CGI de MapServer, radica en que con la primera no se garantiza interoperabilidad, lo que conlleva a aplicar una arquitectura orientada a servicios para resolver este problema en cuestión.

ARCHIVO DE CONFIGURACIÓN EN MAPSERVER

El MapFile, es el archivo de configuración de MapServer, en el mismo se describe como será la salida de los mapas, y la conformación de la cartografía.

La estructura jerárquica del mismo se muestra a continuación:

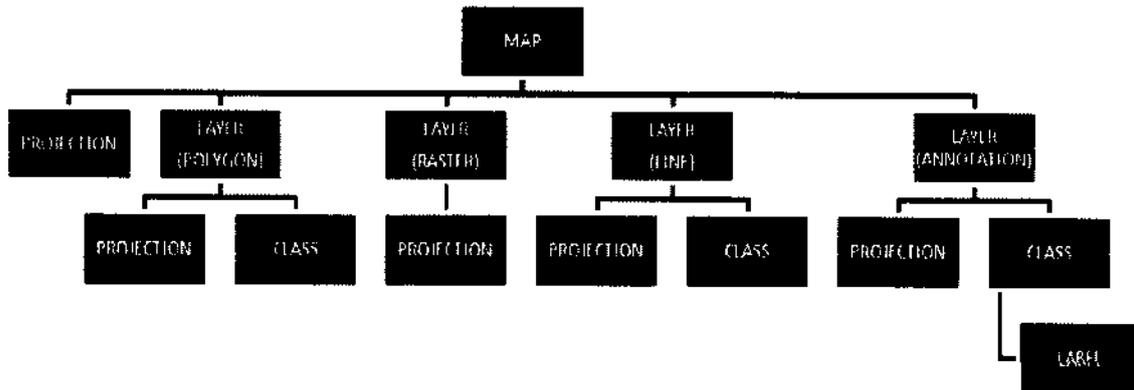


Figura 8– Árbol de jerarquía del MapFile de MapServer (Fuente: <http://mapserver.gis.umn.edu>)

Cada uno de los niveles de jerarquía del árbol anterior tiene especificaciones diferentes, por lo que resultaría desventajoso para un proyecto de desarrollo específico, detenerse a revisar la configuración del **MapFile**. Debido a lo anterior, esta configuración queda oculta a los clientes que interactúan con la interfaz de servicios permitiendo solamente el uso y explotación de las funcionalidades que sean públicas para todos los usuarios.

Es importante destacar el término **CLASS** dentro del nivel **LAYER**. En este caso esto permitirá una abstracción para cada capa de la cartografía, lo que desde el punto de vista de la programación permitirá la manipulación de las capas del mapa de forma independiente, de la manera tradicional establecida por el paradigma de la programación orientada a objetos.

A continuación se muestra un fragmento de configuración del **MapFile** de MapServer para la configuración de la cartografía, tanto desde el punto de vista organizativo como desde el punto de los parámetros de salida de cara al cliente.

Tabla 3 – Ejemplo de configuración de a partir del MapFile.

| | |
|---|--|
| MAP | # Especificación del objeto MAP |
| NAME loja | # Nombre del objeto MAP |
| IMAGETYPE PNG | # Definición del tipo de imagen |
| EXTENT 476041.48 9438571.50 1186767.15 | # Extensión georeferenciada del mapa |
| 10161286.94 | # Tamaño de presentación del mapa |
| SIZE 400 300 | # Fuente de origen de los datos |
| SHAPEPATH "data" | # Definición de archive template |
| IMAGECOLOR 255 255 255 | # Color de relleno de la imagen |
| TEMPLATEPATTERN "loja" | |
| UNITS METERS | # Definición de unidades de medida |
| WEB | # Definición del objeto WEB |
| TEMPLATE 'ejemplo2.1.html' | |
| IMAGEPATH 'c:/practica1/temp/' | |
| IMAGEURL 'c:/maps' | |
| END | |
| PROJECTION | # definición de proyección para MAP |
| "proj=laea" | |
| "ellps=clrk66" | |
| "lat_0=45" | |
| "lon_0=-100" | |
| END | |
| LAYER | # Inicio de la definición del layer# |
| NAME states | Definición Layer Mapa Ecuador |
| DATA cuador2 | |
| STATUS DEFAULT | |
| TYPE POLYGON | |
| PROJECTION | # Definición de proyección para LAYER |
| "init=epsg:2163" | |
| END | |
| CLASS | |
| COLOR 232 232 232 | |
| OUTLINECOLOR 32 32 32 | |
| END | |
| END | # Fin de la definición de LAYER |
| END | # Fin del archivo MAP |

Como se observa en el ejemplo existen parámetros para los cuales un usuario que no tenga conocimiento de la Geomática, no obtendría los resultados deseados. Además en el ejemplo mostrado se utiliza la configuración a partir de ficheros secuenciales los cuales no son la mejor variante para obtener

mejoras de rendimiento. En la tabla 4 se muestra un fragmento de configuración relacionada con el gestor de base de datos PostgreSQL.

Tabla 4 – Ejemplo de configuración con conexión a un SGBD

```
LAYER
NAME "Contornos"
STATUS ON
CONNECTIONTYPE POSTGIS
CONNECTION "user=postgres password=postgres dbname=MapaUCI
host=localhost"
CLASSITEM "the_geom"
DATA "the_geom from contornos"
TYPE POLYGON
```

CAPA DE WEB SERVICES

Capa de interacción de los clientes y aplicaciones clientes. Esta capa será la interfaz de los clientes o diferentes aplicaciones del entorno universitario para utilizar los servicios.

Para esto se utilizó Apache Axis2 debido a que los servicios Web son sistemas software diseñado para soportar interoperabilidad entre diferentes máquinas en una red.

Para conseguir esta interoperabilidad los servicios Web están basados en tecnologías estándares: XML, WSDL (Web Service Description Language), SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration).

SOAP (SIMPLE OBJECT ACCESS PROTOCOL)

SOAP es un protocolo de mensajes entre computadoras que especifica el formato de mensaje que accede e invoca a los objetos. El mismo es un protocolo simple de comunicación para enviar mensajes del tipo RPC.

SOAP trabaja con la infraestructura de Internet. No hay que hacer configuraciones especiales en routers, firewall, proxy servers, para que SOAP trabaje. Usando HTTP se vuelve muy fácil de desarrollar.

Varios proveedores y organizaciones han soportado la implementación de SOAP. W3C ha definido a SOAP como una especificación, y ninguna organización o un simple proveedor tendrán el control sobre SOAP, por lo tanto

se ha convertido en otro estándar de Internet, tales como HTML, XML, HTTP y SMTP.

2.3.1.1.1.1 WSDL

WSDL es un formato XML que describe los servicios que se exponen en la interfaz de servicios. Este formato permitirá a las aplicaciones clientes de la comunidad de desarrolladores comunicarse eficientemente con el servicio.

Entre los parámetros que se establecen en el WSDL se encuentra los siguientes parámetros:

- URL del servicio
- Nombre del servicio
- Nombre de la funcionalidad
- Parámetros que requiere
- Parámetro que devuelve
- Puerto de comunicación

De esta forma un desarrollador de aplicaciones a partir de la descripción anterior puede utilizar un servicio sin necesidad de saber como esta implementado internamente.

APACHE AXIS2

Los Web Services son componentes que pueden ser utilizados sin preocuparse de como fueron implementados, y ser accedidos utilizando XML (SOAP), generalmente sobre HTTP. En este sentido los Web Service son aplicaciones modulares, que pueden ser publicadas, localizadas, e invocadas desde cualquier lugar de la WEB, combinando los mejores aspectos de los componentes de programación, y programación Web, y son un paquete en forma de módulos, que pueden ser reutilizados, sin preocuparnos acerca de cómo el servicio esta implementado, o aún en que lenguaje, sistema operativo, o que modelos de componentes fueron usados para crearlos, es decir, la conexión al servicio, el mapeo de XML y otros aspectos son ignorados por el desarrollador encargado de alimentarse de estos servicios.

Los requerimientos a la hora de desarrollar o consumir un Web Services son:

- Representación de datos: XML
- Selección de un formato común y extensible de mensaje: SOAP
- Selección de un lenguaje común y extensible para describir los servicios: WSDL (basado en XML).

Apache Axis2 es una arquitectura pensada para soportar la creación, mantenimiento y publicación de Web Services basados en tecnología SOAP que cumple con los protocolos y características vistas anteriormente.

Debido a esto se propone una interfaz de servicios cartográficos de cara al cliente donde la configuración de la cartografía sea independiente de las funcionalidades que se brinden, logrando que los usuarios o desarrolladores dependan solamente de las funcionalidades y no de la configuración.

A partir del análisis de las diferentes tecnologías analizadas en este capítulo y en función del objetivo de este trabajo, quedan definidas las tecnologías a utilizar para el desarrollo de la interfaz de servicios cartográficos basado en una plataforma de desarrollo corporativa en J2EE.

2.4 CONCLUSIONES DEL CAPITULO 2

En el capítulo se ha presentado las principales modificaciones que se le han realizado a MapServer y se han analizado las tecnologías que se utilizarán para llevar a cabo la implementación de lo antes planteado.

La incorporación de la capa de persistencia a MapServer permitirá disminuir las responsabilidades del mismo y posibilitará ante migraciones de bases de datos lograr una transparencia en el servicio sin la necesidad de cambios de implementación. En este caso los cambios relacionados con la migración estarán asociados a configuraciones del **MapFile** y los ficheros de configuración de **Hibernate**.

La incorporación de la capa de servicios permitirá a los usuarios la explotación de los mismos sin la necesidad de contar en sus puestos de trabajo con las librerías que se encargan tradicionalmente de realizar estas funcionalidades.

Debido a esto con la tecnología utilizada, basada es ISAPI el rendimiento de MapServer aumenta ya que la misma utiliza la tecnología multihilos para resolver las solicitudes realizadas a los clientes, por lo que se mejoran los tiempos de respuesta y se eleva el rendimiento del servidor de mapas.

La utilización de SOAP, permitirá una comunicación e interoperabilidad entre los sistemas a partir del intercambio de información basada en XMLs., garantizando la interoperabilidad entre los diferentes sistemas.

CAPÍTULO 3 – ANÁLISIS DE LOS RESULTADOS.

3.1 INTRODUCCIÓN

En el capítulo anterior se definieron las tecnologías que se proponen para un adecuado desarrollo de la interfaz de servicios cartográficos. En este caso se fundamentaron las causas que conllevaron a la selección de cada una de ellas en aras de darle solución a los objetivos propuestos.

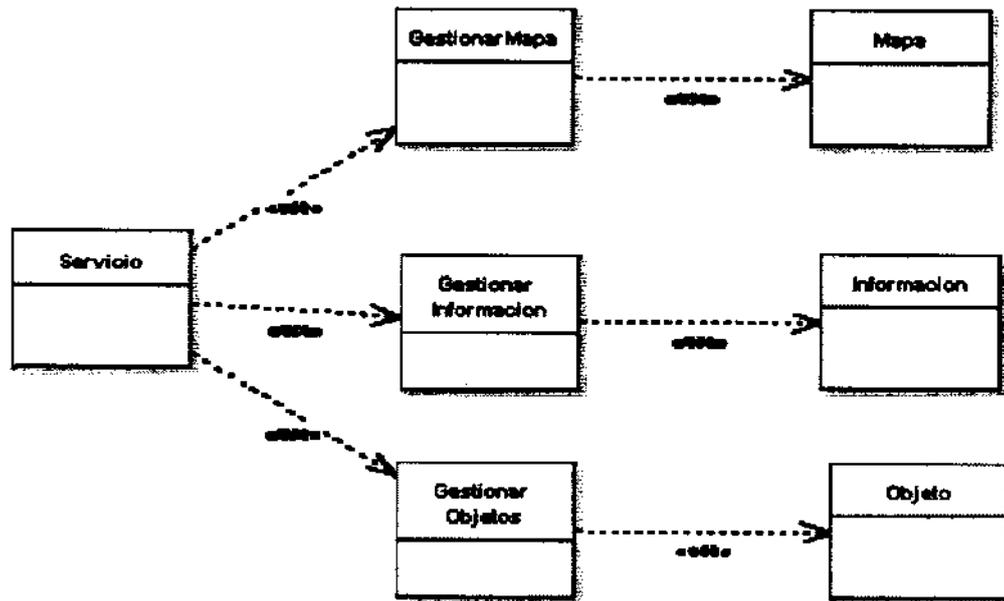
Para el desarrollo un producto de software es necesario realizar la modelación de la propuesta de solución, creando los artefactos que ayuden a la construcción del producto y utilizando técnicas que el cumplimiento de las expectativas.

Para realizar la modelación de la propuesta de solución se realizó la modelación de los siguientes artefactos:

- Modelo de dominio
- Especificación de requerimientos de software
- Modelo de clases del análisis.
- Modelo de clases del diseño.
- Modelo de implementación.

Para la modelación de los artefactos anteriores se utilizó la herramienta Enterprise Architect, la cual permite un desarrollo rápido, además de incorporar el lenguaje de modelado UML 2.0 y permitir la integración de plataformas de desarrollo con vistas a la generación de código.

3.2 MODELO DE DOMINIO



3.3 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

3.3.1 REQUISITOS FUNCIONALES

3.3.1.1 Realizar operaciones que permitan la manipulación de la cartografía.

3.3.1.1.1 Permitir realizar ampliaciones.

3.3.1.1.2 Permitir realizar reducciones.

3.3.1.1.3 Permitir realizar paneos.

3.3.1.1.4 Permitir realizar calcular distancias.

3.3.1.2 Permitir brindar información de cualquier objeto del mapa. (Edificaciones, Áreas Deportivas, etc.)

3.3.1.3 Realizar la gestión de objetos

3.3.1.3.1 Permitir insertar nuevos objetos en el mapa.

3.3.1.3.2 Permitir insertar textos en el mapa.

3.3.2 REQUISITOS NO FUNCIONALES

Requerimientos de diseño

3.3.2.1 La arquitectura que se utilizará para el desarrollo de la interfaz de servicios, será de n capas:

3.3.2.1.1 Capa de servicios.

3.3.2.1.2 Capa de Lógica de Negocio.

3.3.2.1.3 Capa de Acceso a Datos.

3.3.2.2 El lenguaje de programación que se usará será Java.

3.3.2.3 Para el análisis y el diseño del sistema se utilizará la metodología RUP, usando el lenguaje de modelación UML y como herramienta para llevarlo a cabo el Enterprise Architect.

Requerimientos de Seguridad

3.3.2.4 La interfaz debe realizar un seguimiento de trazas.

Requisitos de Rendimiento

3.3.2.5 La interfaz requiere de un buen rendimiento en máquinas de pocos recursos de Hardware.

3.2.2.6 La interfaz debe soportar transacciones simultáneas.

Requisitos de confiabilidad

3.3.2.7 La interfaz deberá tener un 100% de disponibilidad por lo que podrá ser usado las 24 horas del día.

3.3.2.8 El tiempo medio de reparación debe ser menor de 1 día para que la aplicación esté el mayor tiempo posible brindando servicios.

3.3.2.11 Todas las salidas del sistema tienen que tener el 100% de veracidad y precisión de acuerdo con las informaciones que están en la Base de Datos.

Requerimientos de Interfaz de usuarios

3.3.2.12 Interfaz de software

3.3.2.12.1 La interfaz debe prestar servicios a todos los usuarios desde cualquier ordenador conectado a la red.

3.3.2.13 Interfaz de usuario

3.3.2.13.1 La interfaz permitirá a las aplicaciones utilizar los servicios básicos sin necesidad de autenticación.

3.4 MODELO DE SISTEMA

3.4.1 DESCRIPCIÓN DE ACTORES

| | |
|---------|---|
| Cliente | Otro sistema que interactúa con la "Interfaz de Servicios". |
|---------|---|

3.4.2 DIAGRAMA DE CASOS DE USO DEL SISTEMA

A partir de la definición de estas tecnologías se propone el siguiente modelo de casos de uso en relación con la arquitectura propuesta.

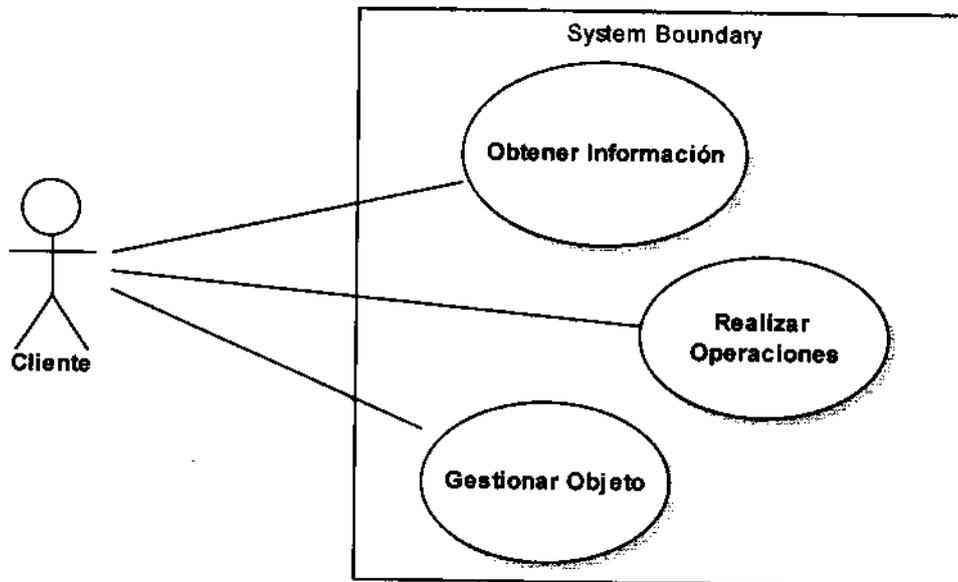


Figura 9 – Diagrama de casos de usos

3.4.3 ESPECIFICACIÓN DE CASOS DE USO DEL SISTEMA

Caso de uso Realizar Operaciones

| Especificación de Casos de Uso del Sistema | |
|--|--|
| Identificador | CUS_01 |
| Nombre | Realizar Operaciones |
| Versión | 01 |
| Descripción | El Caso de Uso se inicia cuando el Cliente realiza la petición de operación con el mapa. El sistema debe permitir realizar ampliaciones, reducciones y paneos en un área geográfica determinada. Además, debe permitir calcular la distancia entre dos puntos y obtener la ampliación de una zona geográfica a partir de la selección de un rectángulo y la escala de ampliación. Cada vez que termina una de estas acciones, finaliza el Caso de Uso. |
| Actor primario | Cliente |
| Actores secundarios | NA |
| Evento de inicio del caso de uso | NA |
| Precondiciones | NA |
| Poscondiciones | El cliente obtiene diferentes perspectivas del área señalada en el mapa, así como la distancia entre dos puntos. |
| Relaciones | <i>include</i> NA |
| | <i>extend</i> NA |
| | <i>uses</i> NA |
| Prioridad de Implantación | Alta. |
| Temas abiertos | NA |
| Requisitos Funcionales | R3.3.1.1.1, R3.3.1.1.2, R3.3.1.1.3, R3.3.1.1.4 |
| Requisitos Especiales | R3.3.2.5, R3.3.2.6, R3.3.2.7, R3.3.2.8, R3.3.2.9, R3.3.2.10, R3.3.2.11, R3.3.2.12 |
| Flujo de Eventos | |

| Acción del Actor | Respuesta del Sistema |
|---|--|
| <p>1. El caso de uso comienza cuando el Cliente realiza una petición para pintar el mapa.</p> | <p>2. Devuelve la URL donde está el resultado de la petición realizada y las opciones "Zoom", "Pan", "PanDrag", "ZoomRectangle", "Text", "MakeLine" y "Distancia2ptos".</p> |
| <p>3. Escoge la opción que considere más conveniente.</p> | <p>4. Ejecuta una de las siguientes acciones:</p> <ul style="list-style-type: none"> a) Si la opción seleccionada es "Zoom", el sistema devuelve la nueva URL con la ampliación del mapa y finaliza el caso de uso. b) Si la opción seleccionada es "ZoomRectangle", el sistema devuelve la nueva URL con el área geográfica que pertenece al rectángulo señalado en el mapa y finaliza el caso de uso. c) Si la opción seleccionada es "Text", el sistema devuelve la nueva URL con un campo de texto para que el cliente pueda escribir una cadena de caracteres y finaliza el caso de uso. d) Si la opción seleccionada es "MakeLine", el sistema devuelve la nueva URL que muestra la línea pintada en el mapa y finaliza el caso de uso. e) Si la opción seleccionada es "Distancia2ptos", el sistema devuelve la nueva URL que muestra la distancia entre los puntos señalados y finaliza el caso de uso. f) Si la opción seleccionada es "Pan", el sistema devuelve la nueva URL que muestra la nueva perspectiva del mapa y finaliza el caso de uso. g) Si la opción seleccionada es "PanDrag", el sistema devuelve la nueva URL que muestra la nueva perspectiva del mapa y finaliza el caso de uso. |

| Notas | |
|-------|---|
| N1 | En el caso de la función Zoom, se requiere introducir un punto y el factor de escala del mismo, a partir de esta información se realiza la solicitud del servicio obteniéndose la Url de la ampliación del mapa. |
| N2 | Para el caso de las reducciones el factor de escala debe ser inferior al último factor de escala utilizado. |
| N3 | En el caso de la función "ZoomRectangle", se requiere introducir dos puntos y un factor de escala, a partir de esta información se realiza la solicitud del servicio obteniéndose la URL de la ampliación del mapa. |

Caso de uso Obtener Información

| Especificación de Caso de Uso de Sistema | | |
|--|--|----|
| Identificador | CUS_02 | |
| Nombre | Obtener Información | |
| Versión | 01 | |
| Descripción | El Caso de Uso se inicia cuando el Cliente realiza la petición para obtener información de cualquier punto del mapa. El sistema devuelve la información solicitada (Nombre, Área y Centroide). Cada vez que termina esta acción finaliza el Caso de Uso. | |
| Actor primario | Cliente | |
| Actores secundarios | NA | |
| Evento de inicio del caso de uso | NA | |
| Precondiciones | NA | |
| Poscondiciones | El sistema devuelve la URL con información del punto señalado en el mapa. | |
| Relaciones | <i>include</i> | NA |
| | <i>extend</i> | NA |
| | <i>uses</i> | NA |

| Especificación de Caso de Uso de Sistema (Continuación) | |
|--|--|
| Prioridad de Implantación | Alta. |
| Temas abiertos | NA |
| Requisitos Funcionales | R3.3.1.2 |
| Requisitos Especiales | R3.3.2.5, R3.3.2.6, R3.3.2.7, R3.3.2.8, R3.3.2.9, R3.3.2.10, R3.3.2.11, R3.3.2.12 |
| Flujo de Eventos | |
| Flujo Básico | |
| Acción del Actor | Respuesta del Sistema |
| 1. El caso de uso comienza cuando el Cliente realiza una petición para obtener información de un punto del mapa. | 2. Ejecuta una de las siguientes acciones: a) Si la petición realizada es una consulta, el sistema devuelve el id, nombre, área y centroide del punto seleccionado. b) Si la petición realizada es la de área, el sistema devuelve el área del objeto señalado en el mapa. c) Si la petición realizada es la de centroide, el sistema devuelve el centroide del objeto indicado en el mapa. |
| Notas | |
| N1 | En el caso de que la consulta se realice sobre un vial o un separador, el sistema la URL con el id y el nombre de ellos. |

Caso de Uso Gestionar Objeto

| Especificación de Caso de Uso de Sistema | |
|--|--|
| Identificador | CUS_03 |
| Nombre | Gestionar Objeto |
| Versión | 01 |
| Descripción | El Caso de Uso se inicia cuando el Cliente realiza la petición incorporar nuevos objetos al mapa. El sistema debe devolver la nueva URL con la información solicitada. Cada vez que termina esta acción finaliza el Caso de Uso. |
| Actor primario | Cliente |
| Actores secundarios | NA |
| Evento de inicio del caso de uso | NA |
| Precondiciones | NA |
| Poscondiciones | El sistema devuelve la URL donde se encuentra el nuevo mapa. |
| Relaciones | <i>include</i> NA |
| | <i>extend</i> NA |
| | <i>uses</i> NA |
| Prioridad de Implantación | Alta. |
| Temas abiertos | NA |
| Requisitos Funcionales | R3.3.1.3.1, R3.3.1.3.2 |
| Requisitos Especiales | R3.3.2.4, R3.3.2.5, R3.3.2.6, R3.3.2.7, R3.3.2.8, R3.3.2.9, R3.3.2.10, R3.3.2.11, R3.3.2.12 |

| Especificación de caso de uso de Sistema (Continuación) | |
|--|--|
| Flujo de Eventos | |
| Flujo Básico | |
| Acción del Actor | Respuesta del Sistema |
| 1. El caso de uso comienza cuando el Cliente realiza una petición para incorporar objetos al mapa. | 2. Ejecuta una de las siguientes acciones: a) Se determina el usuario que esta realizando la petición. b) Se incorpora dicha entidad en una tabla temporal, donde se registra además el usuario de sesión que gestionó la entidad correspondiente. |
| Notas | |
| N1 | Las entidades que se incorporan en la tabla temporal se eliminan cuando la misma cierre. |

3.5 ANÁLISIS Y DISEÑO

3.5.1 MODELO DE CLASES DEL ANÁLISIS

El modelo de clases del análisis se desarrolló con el objetivo de tener una idea general de los requisitos que se concibieron de manera que posibilite el mantenimiento de la plataforma en iteraciones posteriores.

Caso de uso Realizar Operaciones

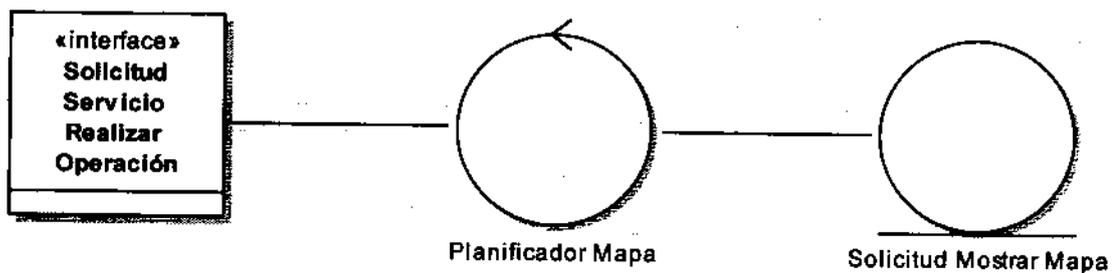


Figura 10 – Modelo de clase del análisis para el caso de uso Realizar Operaciones.

Caso de uso Gestionar Objeto

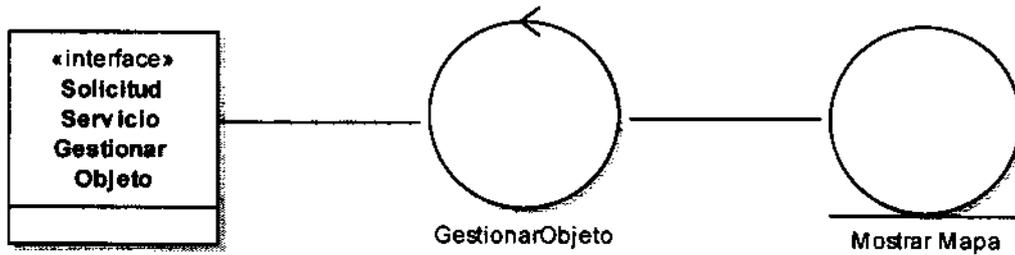


Figura 11 – Modelo de clase del análisis para el caso de uso Gestionar Objeto.

Caso de Uso Obtener Información



Figura 12 – Modelo de clase del análisis para el caso de uso Obtener Información.

3.5.2 MODELO DE CLASES DEL DISEÑO

El modelo de clases del diseño se desarrolló con el objetivo de describir la relación que existe entre cada una de las clases que participan en los diferentes casos de uso. En este caso se desarrolló el diagrama de clases y el diagrama de secuencia.

Caso de uso Realizar Operaciones

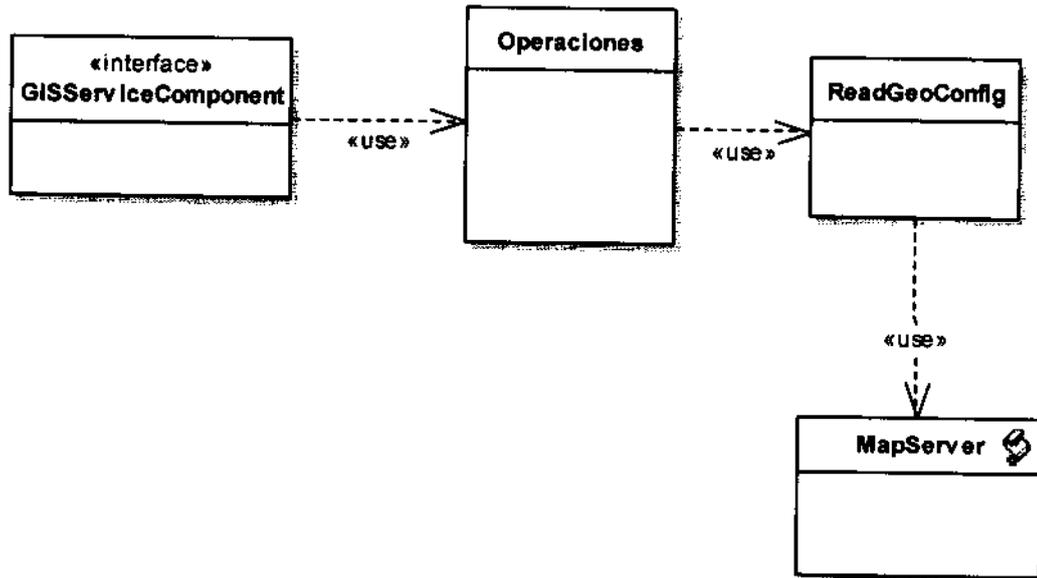


Figura 13 – Diagrama de clases del diseño del Caso de Uso Realizar Operaciones.

Caso de uso Gestionar Objeto

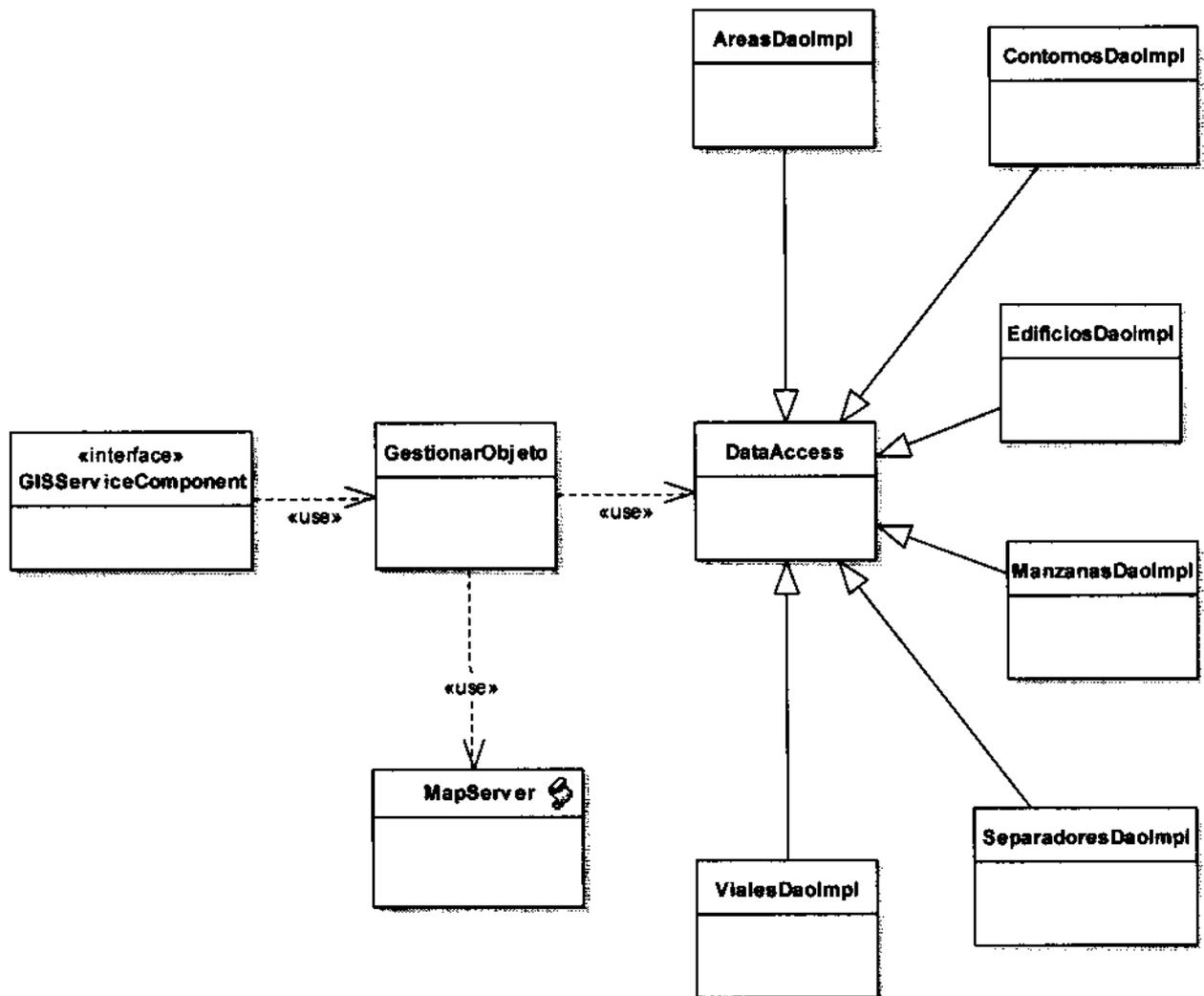


Figura 14 – Diagrama de clases del diseño del Caso de Uso Gestionar Objeto

Caso de Uso Obtener Información

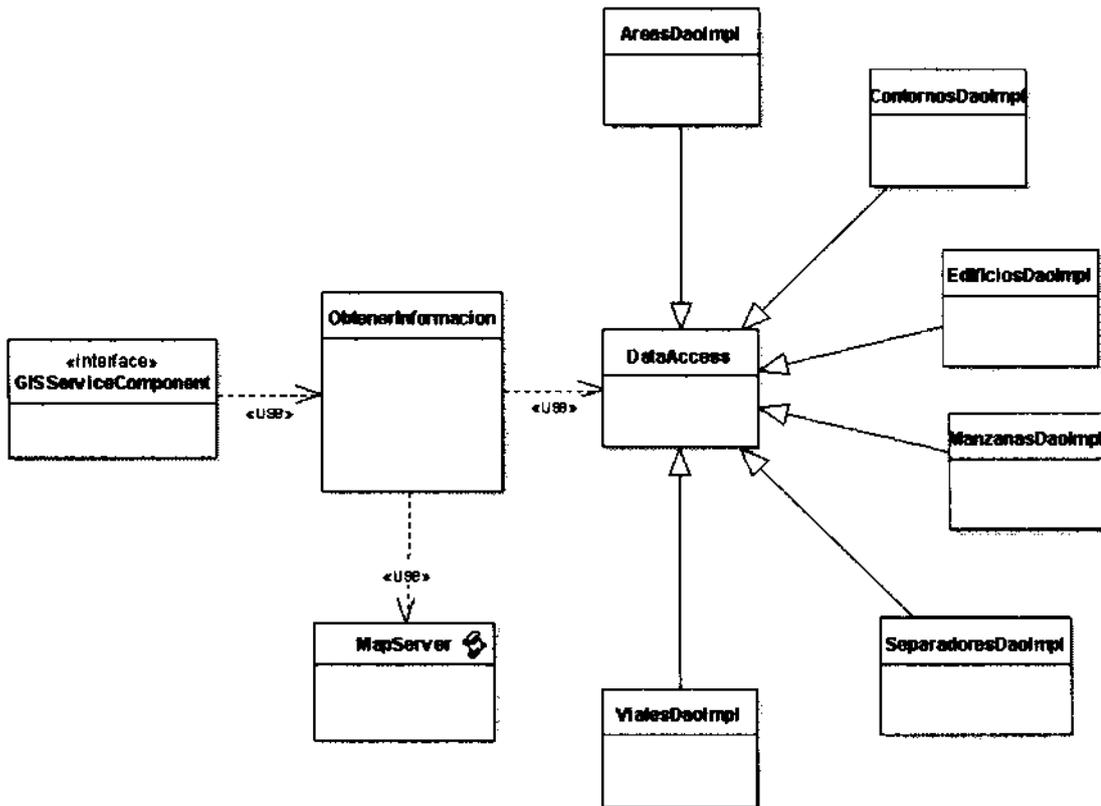


Figura 15 – Diagrama de clases del diseño del Caso de Uso Obtener Información

A continuación se muestra los diagramas de secuencia de cada uno de los casos de uso.

Caso de uso Realizar Operaciones

Este caso de uso presenta diferentes escenarios. A continuación se muestran algunos de ellos.

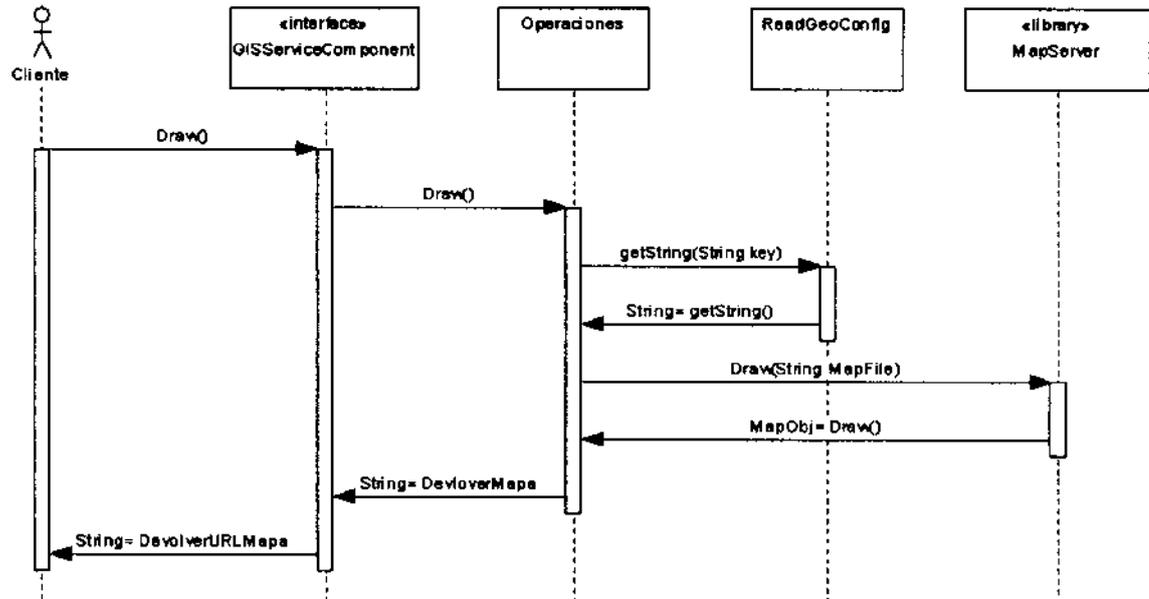


Figura 16 – Diagrama de secuencia del Caso de Uso Realizar Operaciones (Escenario Dibujar Mapa).

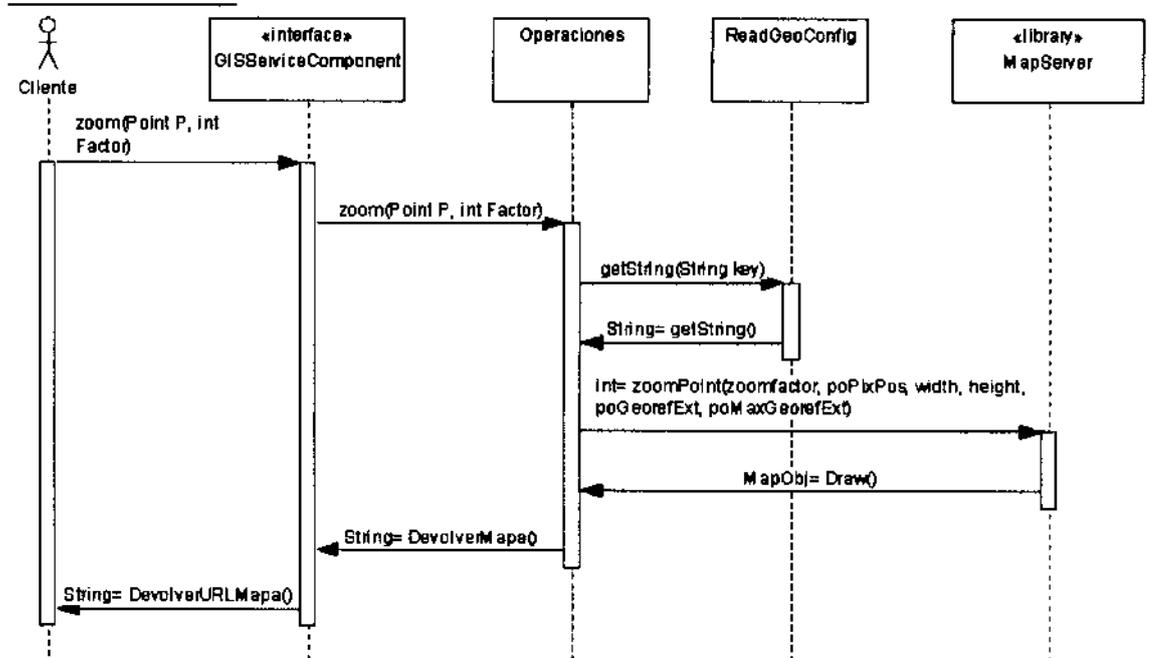


Figura 17 – Diagrama de secuencia del Caso de Uso Realizar Operaciones (Escenario Ampliar Mapa).

Caso de uso Gestionar Objeto

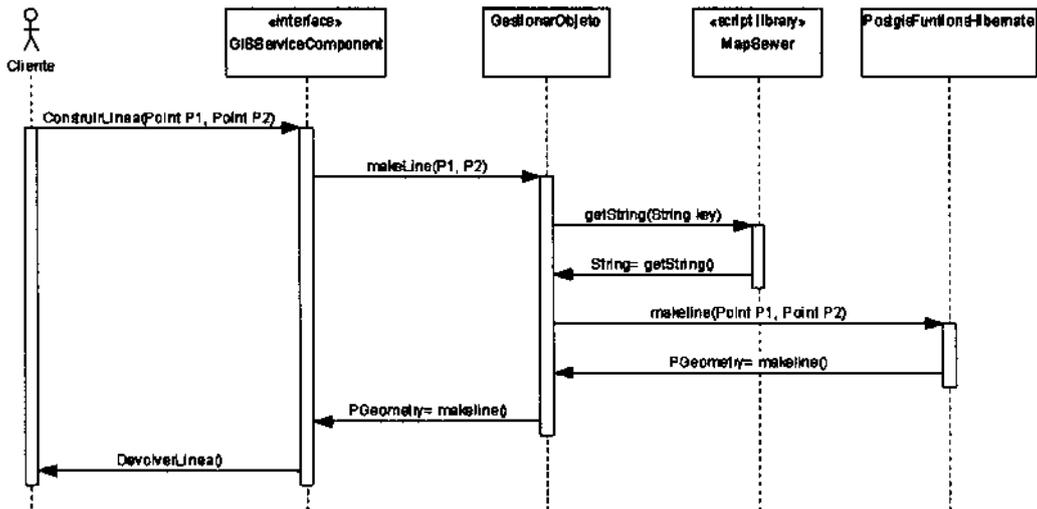


Figura 18 – Diagrama de secuencia del Caso de Uso Gestionar Objeto (Escenario Construir Línea).

Caso de Uso Obtener Información

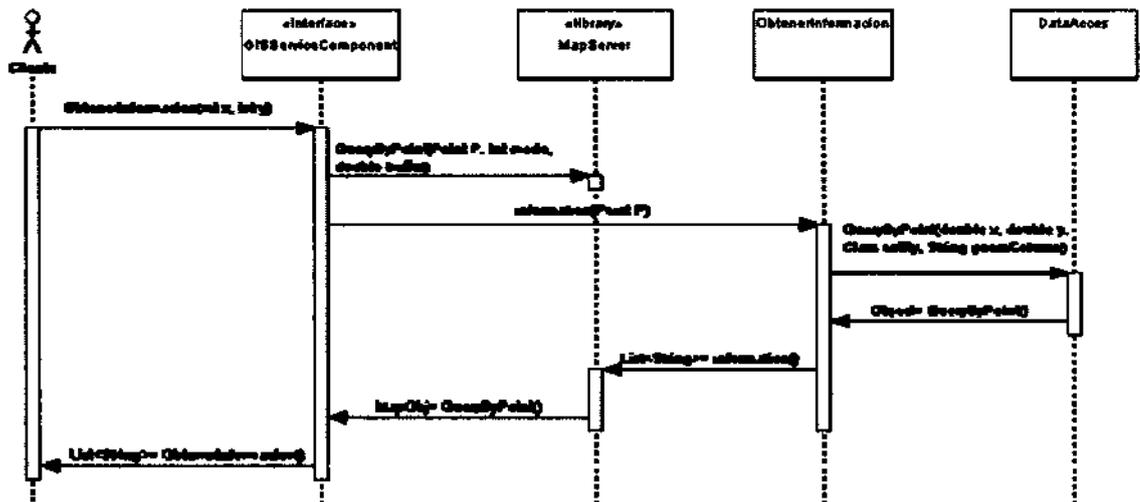


Figura 19 – Diagrama de secuencia del Caso de Uso Obtener Información.

3.5.3 MODELO DE DATOS

Para el modelo de datos se utilizó la base de datos cartográfica de la Universidad de las Ciencias Informáticas, debido a que en el momento de las pruebas que se les desarrollaron a la interfaz era esta la Base de Datos con la que se contaba. Como se explicó anteriormente, se definió la misma desde el punto de vista de esquema de base de datos y desde el punto de vista de

clases a través del paradigma de la programación orientada a objetos. A continuación se muestra el modelo de datos de la misma.

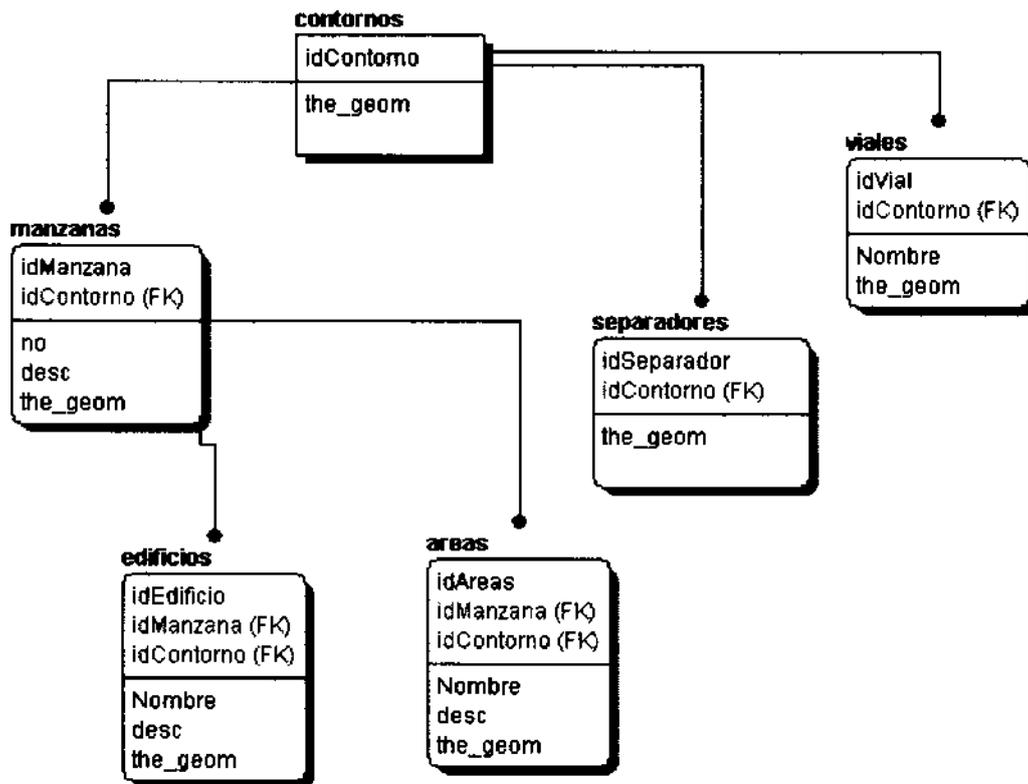


Figura 20 – Modelo de datos utilizado en la interfaz de servicios cartográficos

3.5.4 MODELO DE IMPLEMENTACIÓN

En el modelo de implementación se muestra el modelo de componentes y el modelo de despliegue para la instalación y puesta en marcha de la interfaz de servicios.

3.5.4.1 DIAGRAMA DE COMPONENTES

Para los diagramas de componentes se realizó un diagrama de paquetes que representa la arquitectura utilizada, se representan las capas (servicios, negocio y persistencia) en paquetes, dentro de los cuales están los subpaquetes que responden a los casos de uso y los que responden a los diagramas de cada uno de ellos.

A continuación se muestra el diagrama de paquetes.

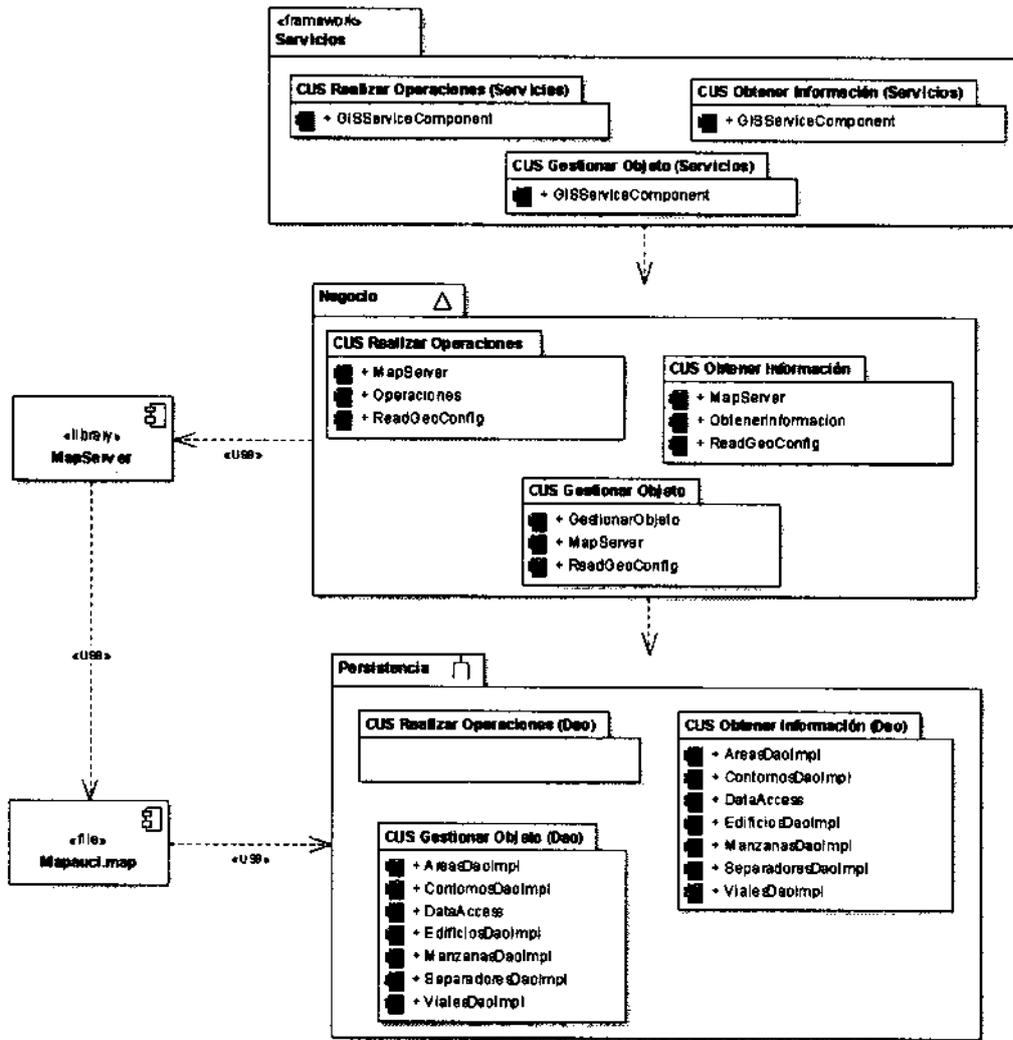


Figura 21 – Modelo de componentes de la interfaz de servicios

3.5.4.2 MODELO DE DESPLIEGUE

Para el modelo de despliegue se utilizó una arquitectura del tipo cliente servidor en aras de permitir la comunicación de los clientes con la interfaz de servicios planteada. En este caso se separó dicha interfaz de la base de datos, la cual se propone que se hospede en un servidor destinado para estas tareas.

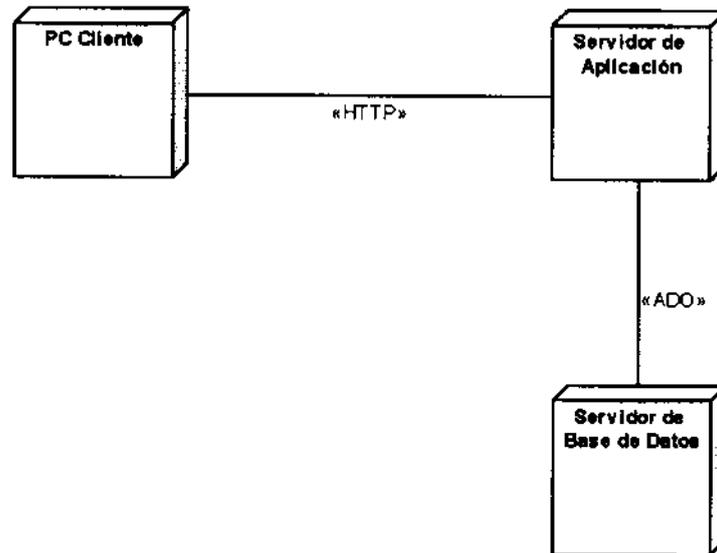


Figura 22 – Diagrama de despliegue para la interfaz de servicios.

3.6 MODIFICACIONES REALIZADAS A HIBERNATE PARA INCORPORAR LOS TIPOS DE DATOS ESPACIALES AL MOTOR DE PERSISTENCIA.

Como se explicó brevemente en el capítulo anterior, para incorporar los tipos de datos espaciales al gestor de base de datos PostgreSQL, es necesario auxiliarse del POSTGIS; sin embargo una vez incorporadas estas modificaciones al gestor de base de datos, el motor de persistencia Hibernate adolece de la información suficiente para poder persistir estos nuevos tipos de datos de la misma manera que lo hace con los tipos de datos primitivos que el mismo brinda.

Ante este problema se realizaron las modificaciones necesarias a partir de la extensión de la interfaz `UserType`, perteneciente a `Hibernate`. En la figura se muestra las funcionalidades que provee dicha interfaz:

```

public abstract interface org.hibernate.usertype.UserType extends java.lang.Object {
    public abstract int[] sqlTypes();
    public abstract Class returnedClass();
    public abstract boolean equals(Object arg, Object arg) throws org.hibernate.HibernateException;
    public abstract int hashCode(Object arg) throws org.hibernate.HibernateException;
    public abstract Object nullSafeGet(ResultSet arg, String[] arg, Object arg) throws org.hibernate.HibernateException, java.sql.SQLException;
    public abstract void nullSafeSet(PreparedStatement arg, Object arg, int arg) throws org.hibernate.HibernateException, java.sql.SQLException;
    public abstract Object deepCopy(Object arg) throws org.hibernate.HibernateException;
    public abstract boolean isMutable();
    public abstract Serializable disassemble(Object arg) throws org.hibernate.HibernateException;
    public abstract Object assemble(Serializable arg, Object arg) throws org.hibernate.HibernateException;
    public abstract Object replace(Object arg, Object arg, Object arg) throws org.hibernate.HibernateException;
}

```

Figura 23 – Interfaz UserType

A partir de la utilización de esta interfaz se realizó la implementación de la clase **GeometryUserType** la cual se muestra a continuación:

- assemble(Serializable, Object)
- deepCopy(Object)
- disassemble(Object)
- equals(Object, Object)
- hashCode(Object)
- isMutable()
- nullSafeGet(ResultSet, String[], Object)
- nullSafeSet(PreparedStatement, Object, int)
- replace(Object, Object, Object)
- returnedClass()
- sqlTypes()

Figura 24 – Interfaz GeometryUserType

Para esto se redefinieron las funcionalidades siguientes:

```
public Object nullSafeGet(ResultSet result, String[] names, Object arg0)
```

```
public void nullSafeSet(PreparedStatement st, Object arg0, int index)
```

En este caso las funcionalidades anteriores son las que van a permitir a **Hibernate** reconocer los tipos de datos geométricos y poder utilizar los mismos de la misma manera que el motor de persistencia utiliza los datos primitivos.

En el caso del método **nullSafeGet** el mismo permitirá leer un objeto **POSTGIS** en la base de datos mientras que en el caso de **nullSafeSet** permitirá insertar un objeto **POSTGIS** de la misma.

3.7 CREACIÓN DE LA INTERFAZ DE SERVICIOS

Para la definición de la interfaz de servicios como se abordó en la capítulo anterior se utilizó SOAP, específicamente el protocolo RPC, utilizado para la ejecución de código de forma remota. Para esto se redefinió el fichero de configuración **webservice.xml**, el cual provee dicho estándar:

Tabla 5 – Esquema de configuración basado en XML para la definición de los Web Services

```
<service name="GIS-WSUCI">
<description/>
<parameter name="ServiceClass" locked="false">component.GISComponent
</parameter>
<operation name="Draw">
<messageReceiver
class="org.apache.axis2.rpc.receivers.RPCMessageReceiver"/>
</operation>
<operation name="getExtend">
<messageReceiver
class="org.apache.axis2.rpc.receivers.RPCMessageReceiver"/>
</operation>
<operation name="getLayers">
<messageReceiver
class="org.apache.axis2.rpc.receivers.RPCMessageReceiver"/>
</operation>
</service>
```

La sencillez de este servicio radica en permitir la descripción de manera muy breve de que clase dentro del componente previamente elaborado es la que contendrá los servicios que se expondrán en la interfaz de servicios cartográficos y cuales son las operaciones que se publicarán en el mismo.

El esquema de configuración que se muestra en la tabla 4 expone los principales parámetros necesarios para una adecuada publicación de los Web Services.

En este caso **component.service** es la clase donde estarán las funcionalidades que se publicarán como servicio utilizando para esto la tecnología que brinda Apache Axis2.

Cada uno de los servicios que se publicarán se describe a través de la marca **<operación>** que se encuentra en el XML. En este caso los métodos **Draw**, **getExtend**, **getLayers** son métodos que se exponen como Web Service, estos son gestionados por Apache Axis2 y resueltos por la clase GISComponent perteneciente al paquete **component**

La marca `<messageReceiver>` es la clase de Apache Axis2 que se encargara de manipular el servicio y de gestionar cualquier error que se produzca durante la ejecución del mismo.

3.8 ANÁLISIS DE APLICABILIDAD.

Desde el punto de vista de la explotación de los servicios es destacable la reducción de los parámetros en comparación con el CGI de de MapServer. A continuación se muestran la utilización de la solicitud de un mapa partir de las especificaciones de WMS a través de MapServer y a partir de la utilización de la interfaz.

```
http://a-map-  
co.com/mapserver.cgi?VERSION=1.3.0&REQUEST=GetMap&CRS=CRS:84&  
BBOX=-97.105.24.913,-  
78.794.36.358&WIDTH=560&HEIGHT=350&LAYERS=AVHRR-09-  
27&STYLES=&FORMAT=image/png&EXCEPTIONS=INIMAGE
```

Tabla 6 – Ejemplo de la utilización del servicio GetMap a través de MapServer

```
http://localhost:8080/axis2/services/MapService?wsdl
```

Tabla 7 – Ejemplo de la utilización del servicio GetMap a través de la interfaz

La diferencia de la ejecución del servicio radica en que el mismo le permite encapsular dentro de un objeto las diferentes funcionalidades del servicio, de manera tal que el programador a partir de una solicitud inicial puede utilizar todas las funcionalidades del mismo.

Es importante destacar que desde el punto de vista del desarrollo del proceso de software, es posible la incorporación de nuevas funcionalidades en futuras iteraciones, las cuales se incorporarán en función de las necesidades que se vayan generando en cada unos de los diferentes proyectos en desarrollo de la comunidad de desarrolladores o entorno universitario por citar un ejemplo.

A continuación se exponen algunos ejemplos de cómo es posible utilizar la interfaz de servicios para diferentes lenguajes de programación.

Tabla 8 – Ejemplo de utilización de la interfaz de manipulación de mapas utilizando PHP

```

<?
require_once('nusoap.php');
$part1 = "<DrawWS xmlns='\"http://awt.java/xsd\"'/>";
$wsdl = "http://10.33.12.37:8080/axis2/services/GisServicesUCI?wsdl";
$soap = new soapclient($wsdl,"wsdl");
$proxy = $soap->getProxy();
$result = $proxy->DrawWS($part1);
?>

```

Tabla 9 – Ejemplo de utilización de la interfaz de manipulación de mapas utilizando JAVA

```

public class MapServiceClient {
    public DibujarMapa() {
        try {
            String nameSpaceUri = "urn:Foo";
            String serviceName = "Draw";
            String portName = "DrawFacePort";

            // Specify the location of the WSDL file
            URL url = new
            URL("http://localhost:8080/MapServices/MapService?WSDL");

            // Create an instance of service factory
            ServiceFactory serviceFactory = ServiceFactory.newInstance();

            // Create a service object to act as a factory for proxies.
            Service mathService = serviceFactory.createService(url, new
            QName(nameSpaceUri, serviceName));

            // Create a proxy
            dynamicproxy.MathFace myProxy =
            (dynamicproxy.MathFace)mathService.getPort
            (new QName(nameSpaceUri, portName), dynamicproxy.MathFace.class);
            // Invoke the add method
            System.out.println(myProxy.Draw());
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}

```

A partir de los ejemplos de uso ilustrados anteriormente se observa la facilidad para consumir un Web Services de la interfaz, sin la necesidad de tener ningún conocimiento de la Geomática. En este caso los tiempos de asimilación sobre la utilización de estos servicios, radica precisamente en el estudio de cómo consumir un Web Services, de cualquier tipo.

En el entorno universitario de la Universidad de las Ciencias Informáticas, la política definida es el uso de los Web Services, por lo que en este caso sólo es necesario descubrir el servicio para su utilización inmediata por parte de las comunidades de desarrollo.

Dicha plataforma permite la adecuación de la cartografía de cualquier región, siempre que utilice los parámetros de diseño adecuados, en función de ganar en eficiencia desde el punto de vista organizativo.

Además de esto se define la cartografía a nivel de clases y de esquemas de base de datos, como se explica en el capítulo 2.

En estos momentos dicha interfaz se está utilizando en el proyecto "Plataforma Digital de Servicios Postales". En este sentido se está desarrollando la cartografía de cada uno de los estados de Venezuela para la utilización de esta interfaz en el proyecto antes mencionado, con vistas a integrar toda la información catastral con la información postal que se genera en las instituciones de correo postal de este país.

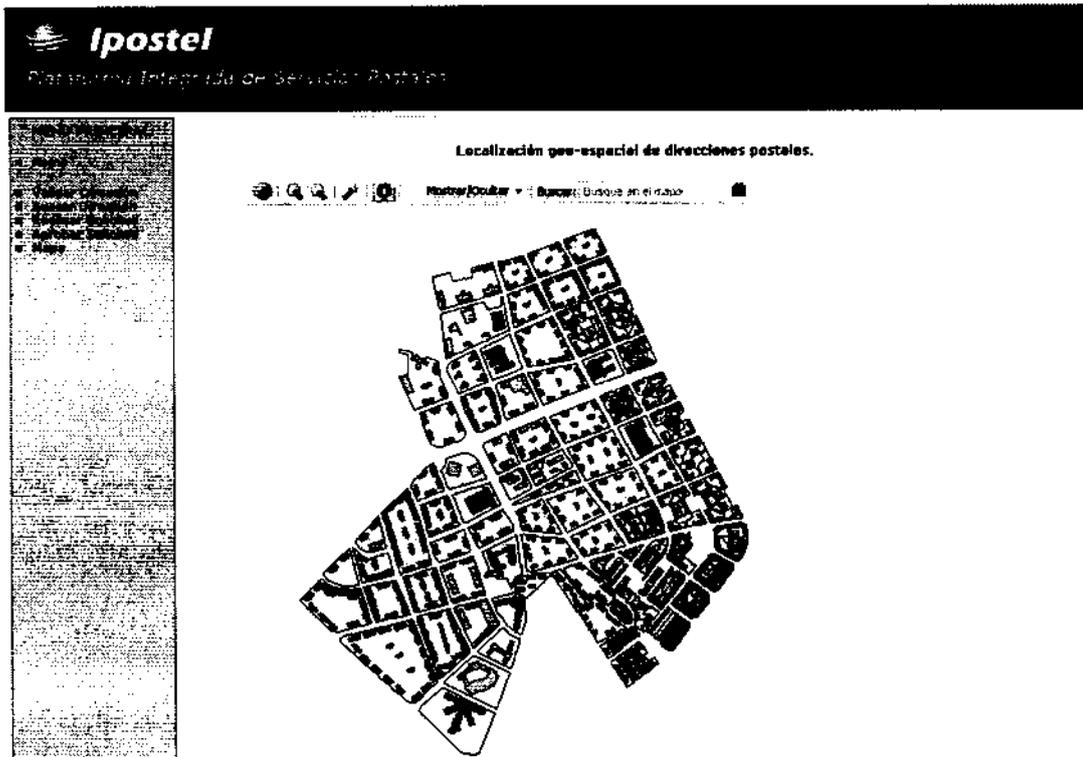


Figura 25 – Resultado de una aplicación que utiliza la interfaz de manipulación de mapas. Además de esto existe un proyecto con el Ministerio de la Industria Básica el cual tiene altas perspectivas para el país en el cual se pretende introducir la experiencia y los resultados adquiridos. Dicho proyecto se encuentra en fase organizativa.

CONCLUSIONES

- 1 Se demostró que utilizando tecnología ISAPI y otras, basadas en SOA se puede aumentar el rendimiento de los Servidores de Mapas.
- 2 La posibilidad de las comunidades de desarrolladores de utilizar la interfaz de servicios desarrollada para la gestión de la cartografía en WEB, sin necesidad de instalación y configuración de herramientas auxiliares permitirá la integración de los datos de las mismas de manera georeferenciada, incorporándole funcionalidades adicionales a los sistemas.
- 3 Los sistemas que integren sus datos de manera georeferenciada a partir de la utilización de la interfaz de servicios, no requieren desviar recursos humanos y materiales para incorporar estas funcionalidades a sus aplicaciones, sino solamente utilizar los servicios que se exponen de manera adecuada.
- 4 La definición de las capas del mapa a partir de clases y a partir de un esquema de bases de datos permitirá a Hibernate la integración de estos dos mecanismos para la manipulación de los diferentes elementos geospaciales de cada una de las capas de la cartografía, por lo que se obtienen mejores resultados desde el punto de vista de mantenimiento de la interfaz y disminuyendo los tiempos de mantenimiento de la aplicación por conceptos de incorporación de nuevas funcionalidades.
- 5 La incorporación de la capa de datos permitirá efectuar la migración de la base de datos geográfica donde este almacenada la cartografía de la región sin que esto conlleve a la modificación de la implementación de la interfaz. En este caso las modificaciones a realizar estará en función de la configuración para el Sistema de Gestión de Base de Datos Geográfica seleccionado.
- 6 La incorporación de una capa de datos permitirá eliminar inyecciones de SQL de manera desordenada, definiendo para esta tarea a la capa de persistencia, la cual tendrá como responsabilidad principal la gestión de toda la información relacionada con la base de datos geográfica.

- 7 La incorporación de la capa de servicios permitirá hacer uso de la interfaz a las diferentes comunidades de desarrollo, sin tener en cuenta el lenguaje y entorno de desarrollo seleccionado.

RECOMENDACIONES

- 1 Implantación de la interfaz de servicios para ser utilizada por la comunidad de desarrolladores de la Universidad de las Ciencias Informáticas para comprobar los rendimientos en comparación con las alternativas anteriores de MapServer.
- 2 Incorporar servicios de mapas en función de las necesidades que se vayan generando en las comunidades de desarrollo en aras de que la misma sea de utilidad para todos los desarrolladores de la misma en tareas relacionadas con la integración de datos de manera georeferenciada.
- 3 Aplicar las experiencias de los resultados que se obtengan en comunidades de desarrollo que no pertenezcan a la Universidad de las Ciencias Informáticas.
- 4 Creación de un grupo de trabajo para el mantenimiento de la interfaz de servicios cartográficos.
- 5 Utilización de la interfaz de servicios postales en el proyecto del MINBAS, sobre la gestión de recursos minerales, el cual se encuentra en fase organizativa.
- 6 Analizar su posible incorporación en la Infraestructura de Datos Espaciales de la República de Cuba (IDERC)

GLOSARIO DE TÉRMINOS

- **SIG:** Sistema de Información Geográfica.
- **Framework:** Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Son diseñados para facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software.
- **IDE:** Infraestructura de datos espaciales. Se encarga de proveer información con vistas a su utilización. El objetivo principal radica en utilizar la interoperabilidad para gestionar la información.
- **WMS:** Este estándar internacional define un mapa como una representación de la información geográfica en forma de un archivo de imagen digital conveniente para la exhibición en una pantalla de ordenador.
- **Interoperabilidad:** Condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos.
- **Pool de conexiones:** Conjunto limitado de conexiones que se reutilizan constantemente para dar servicio a los diferentes clientes, aumentando la escalabilidad y el rendimiento debido a que no es necesario abrir las conexiones constantemente.
- **Multiplexado de conexiones:** Combinación de dos o más recursos para ser compartidos eficientemente.
- **Balanceo de carga dinámico:** Técnica usada para compartir el trabajo a realizar entre varios procesos, ordenadores, discos u otros recursos. El balance de carga utiliza un algoritmo que divide de manera equitativa el trabajo, para evitar los cuellos de botella.
- **Servicios Web:** Colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones sin importar el lenguaje de programación que utilicen y la plataforma sobre la que se ejecuten.

- **ISAPI:** (Internet Server API), consiste en un conjunto de funciones que facilitan el trabajo en aplicaciones basadas en WEB.
- **OpenSource:** Cualidad de algunos softwares de incluir el código fuente en la distribución del programa. En general se usa para referirse al software libre.
- **MapServer:** 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).
- **e-business:** Consiste principalmente en la distribución, compra, venta, mercadotecnia y suministro de información complementaria para productos o servicios a través de redes informáticas como Internet u otras.
- **Metadatos:** Datos estructurados y codificadas que describen características de instancias conteniendo informaciones para ayudar a identificar, descubrir, valorar y administrar las instancias descritas.
- **Geoprocesamiento:** Es un proceso en el que se aplica el análisis geográfico y se modelan los datos para producir nueva información.
- **Georeferenciación:** Proceso mediante el cual se logra una definición geográfica precisa de la ubicación de puntos, líneas y polígonos presentes en un mapa o foto, gracias a la correlación de estos y sus respectivos representados en un sistema de coordenadas reales.

BIBLIOGRAFÍA

1. Alcubilla B, Julio César. "ABC...SOA...Primera Parte". <http://solucionesytecnologia.net>, Mayo, 2007.
2. Aronoff, S. (1987). "Geographical Information Systems: A management perspective, Ottawa, WDL Pub., 294 págs".
3. Benedict, Karl, K. Hudspeth, William "Technology Products of the PHAiRS REASoN Project – Year 2 Web Services and Demonstration Interfaces Development", Earth Data Analysis Center, University of New Mexico, ESTC 2006
4. Boulos, M. N. K. and K. Honda (2006). "Web GIS in practice IV: publishing your health maps and connecting to remote WMS sources using the Open Source UMN MapServer and DM Solutions MapLab." International Journal of Health Geographics.
5. Cadavid, J. Valbuena, R, Amaya, W. "Adaptación de los estándares globales para las naciones en desarrollo". GSDI-5 Conference Proceedings Mayo, 2001, Colombia.
6. Dabrowski, C. "The Reference Model for ISO/TC 211 Geographic Information and Geomatics", National Institute of Standards and Technology, September 1996
7. Delgado Fernández, T (2003). "Área de Servicios de Información Geográfica Proyecto UCI-Ciudad Digital.", Ponencia de jornada de Arranque de Informática en la UCI. GeoCuba, Cuba.
8. Delgado Fernández, T. (2005). "Infraestructuras de datos espaciales para países de bajo desarrollo tecnológico. Implementación en Cuba.", Tesis de Doctorado. Instituto Técnico Militar "José Martí".
9. Díaz, A. A. P. "Graphical User Interface (GUI) para el programa servidor de mapas MapServer 4.6.1., Chile", 2005.
10. DiBiase, David, and others "Body of Knowledge", UCGIS, 2005
11. Donato, Fabio Di. "Web Services: la nueva tecnología para integrar organizaciones", Grupo ASSA, Mayo 2004. <http://www.grupoassa.com/>

12. Echeverría Martínez, Manuel. "Las infraestructuras de datos espaciales: Experiencias de implantación", Artículo publicado en la revista Boletín (septiembre-octubre 2001), pp. 38-50. España, 2001
13. Esbrí, M. Á. and J. V. Higón (2005). "Pruebas benchmark de soluciones cliente/servidor en software libre". Jornadas Técnicas de la IDE de España, Madrid, España, 2005
14. Escartín Sauleda, E. R. "La Tecnología de los Sistemas de Información Geográfica en el Contexto de la Exploración Petrolera", Memorias de GEOMATICA 2002, Febrero 2002.
15. Sigifredo E. Badani H. "Herramientas de desarrollo", Ingeniería de Ejecución en Computación, Universidad Central de Chile, Fuente electrónica:
<http://eiec.ucentral.cl/ftp/material/apuntes/iec61/Arquitectura/3%20Arquitectura.PDF>
16. González Jaramillo, V.H. "MapServer y su aplicación SIG", Universidad Técnica Particular de Loja, Ecuador, 2005
17. Guinea de Salas, Alejandro. Jorin Abellan, Sergio. "Arquitectura SOA para la integración entre software libre y software propietario en entornos mixtos". I Jornada de SIG Libre, España, Marzo, 2007
18. Gutiérrez Valenzuela, Mariella "El Rol de las Bases de Datos Espaciales en una Infraestructura de Datos Espaciales", GSDI-9 Conference Proceedings, 6-10 November 2006, Santiago, Chile
19. <http://es.wikipedia.org>
20. <http://plentz.org/2007/03/28/hibernate-com-postgis/>
21. Jacobson, I. Booch, G Rumbaugh, J. "El proceso unificado de desarrollo de software", PEARSON EDUCACIÓN, S. A, Madrid, 2000.
22. Kolodziej, Kris (2004). "OpenGIS® Web Map Server Cookbook", OGC Document Number: 03-050r1

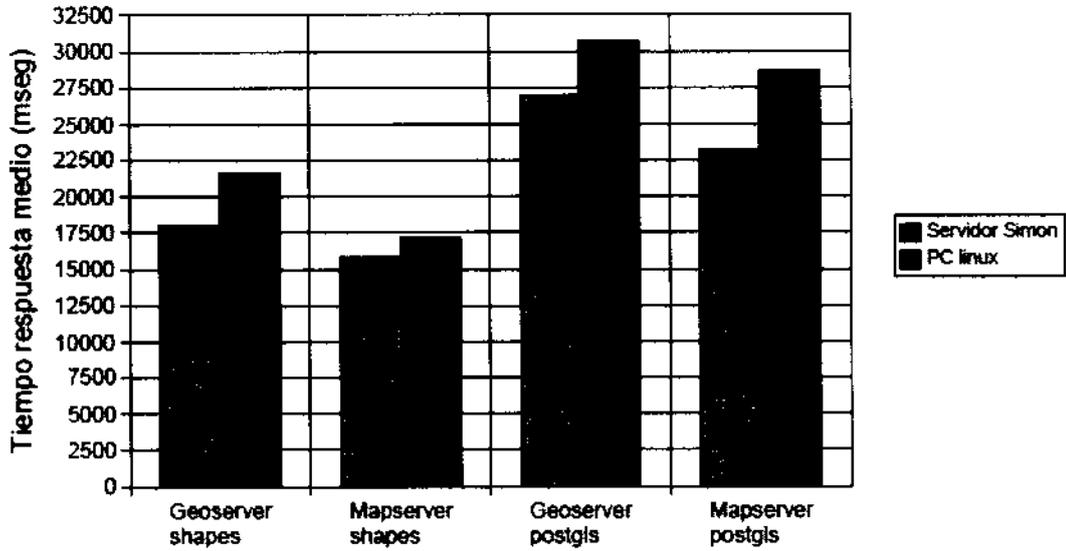
23. Montesinos Lajara, M. Snaz Salinas, Jorge G. "Panorama actual del ecosistema de software libre para SIG", I JORNADAS DE SIG LIBRE, Valencia, 2007
24. Nebert, Douglas D., "The IDE Cookbook". Version 2.0, GSDI, January 2004. <http://www.gsdi.org/>
25. OGC, 2000, "OpenGIS® Web Map Server Interface Implementation Specification", OGC 00-028, Revision 1.0.0, Abril, 2000. <http://www.opengis.org/>
26. OGC, 2002, "OpenGIS® Catalog Services Specification", OGC 02-087r3, Version 1.1.1, December 2002. <http://www.opengis.org/>
27. OGC, 2003 c, "OpenGIS® Reference Model", OGC 03-040, Version 0.1.2, March 2003. <http://www.opengis.org/>.
28. OGC, 2003 e. "OpenGIS® Web Coverage Service". Project Document 02-024. 2003. <http://www.opengis.org/>
29. OGC, 2003 f. "OpenGIS® Web Feature Service". Project Document 01-065. 2003. <http://www.opengis.org/>
30. Ostensen, O. "Global Geomatics Standards supporting Sustainable Geospatial Data Infrastructure". 5ta Conferencia de GSDI, Cartagena de Indias, Colombia, Mayo, 2001
31. Pebesma, Edzer J. "SOA in GIS", <http://www.geog.uu.nl/~pebesma/rags1/soa.pdf>, March 31, 2006
32. Rossiter, D. G. "Notas de conferencia: Bases De Datos Geográficos de Suelos y el Uso de Programas para su Construcción". International Institute for Aerospace Survey & Earth Sciences (ITC) Enschede-The Netherlands, Version Traducida, Junio 2006.
33. Sitio Oficial de deegree <http://deegree.sourceforge.net/>
34. Sitio Oficial de GeoServer <http://geoserver.org/>
35. Sitio Oficial de gvSig <http://www.gvsig.gva.es/>
36. Sitio Oficial de MapGuide Open Source <http://mapguide.osgeo.org/>
37. Sitio Oficial de PostGIS <http://www.refractive.org/>

38. Sitio Oficial de UMN MapServer. <http://mapserver.gis.umn.edu/>
39. Sitio Web <http://www.cartografia.cl>
40. Song, Xianfeng, Kono, Yasuyuki, Shibayana, Mamoru, "An Open Source GIS Solution for Discovering Cambodia through Maps and Facts", FOSS/GRASS Users Conference, Thailand, September, 2004.
41. Vaccari, L. Ivanyuckovich, A. Marchese, M. "A Web Service approach to geographical data distribution among public administration". 5th IFIP Conference on e-Commerce, e-Business, and e-Government I3E'2005, Poznan, Poland, October 26-28, 2005

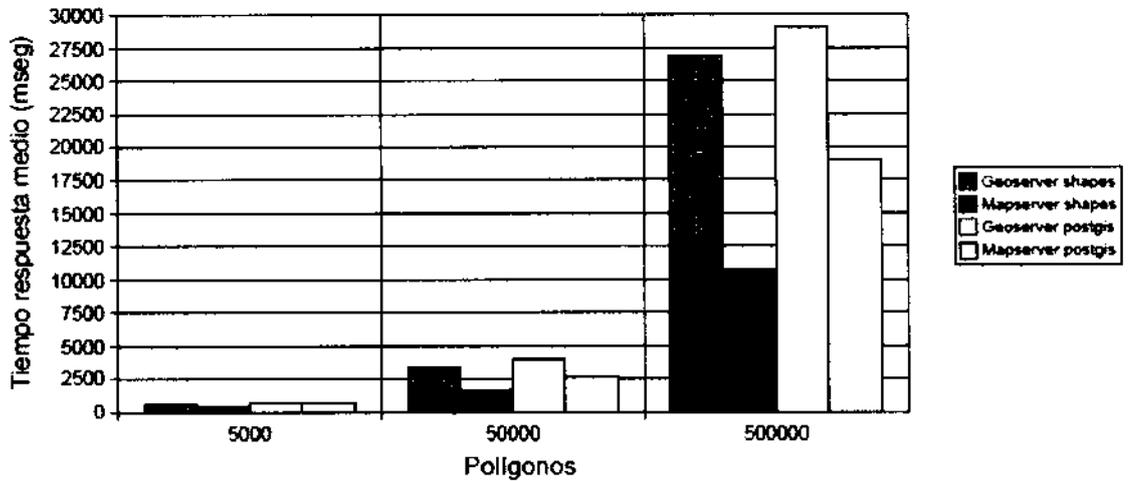
ANEXOS

Anexo 1 - Comparación de Mapserver y Geoserver en acceso secuencial de ficheros.

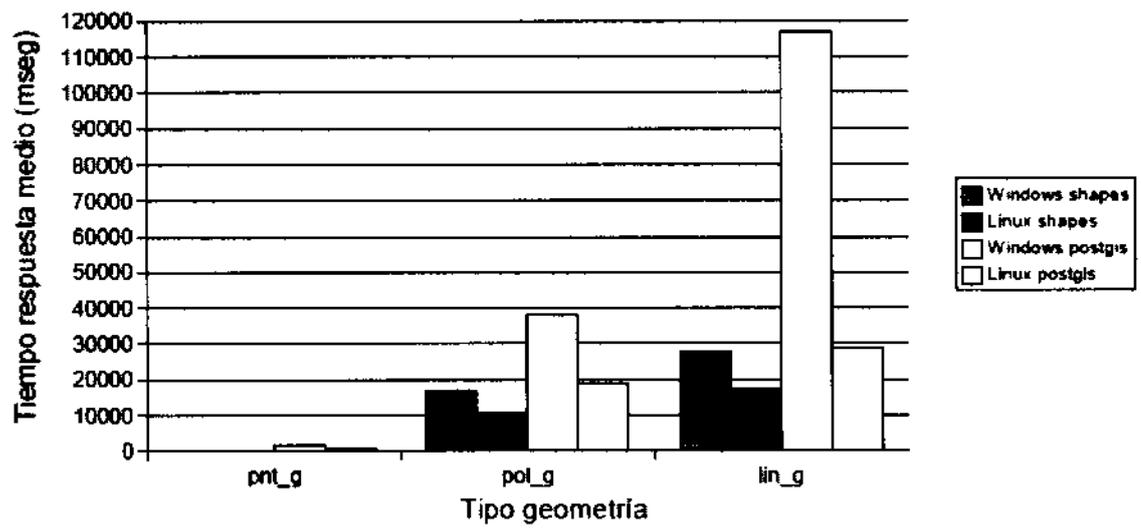
PC vs Servidor. Acceso secuencial (líneas)



Acceso secuencial (polígonos). PC linux

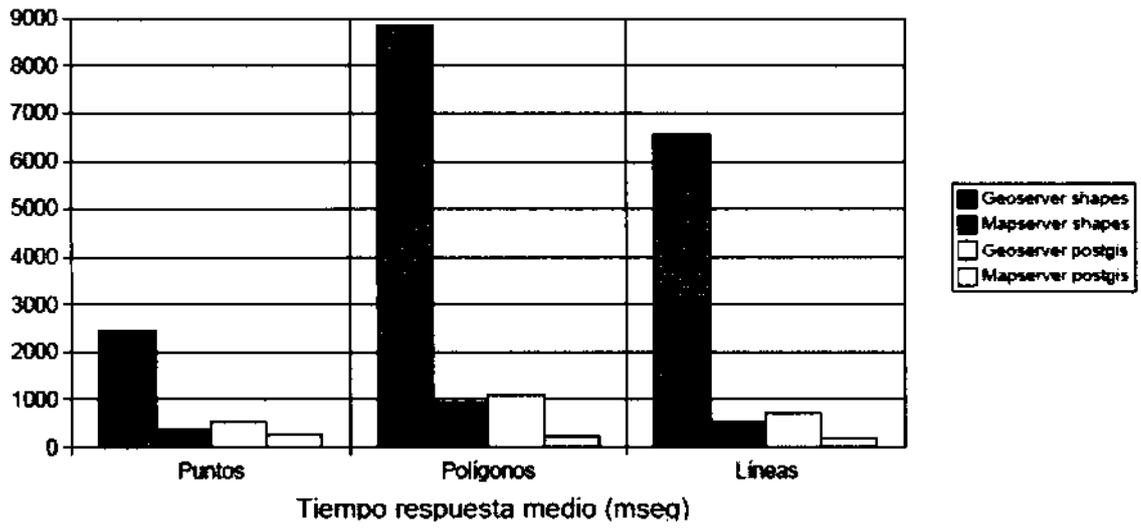


Mapserver Linux vs Windows. Acceso secuencial.

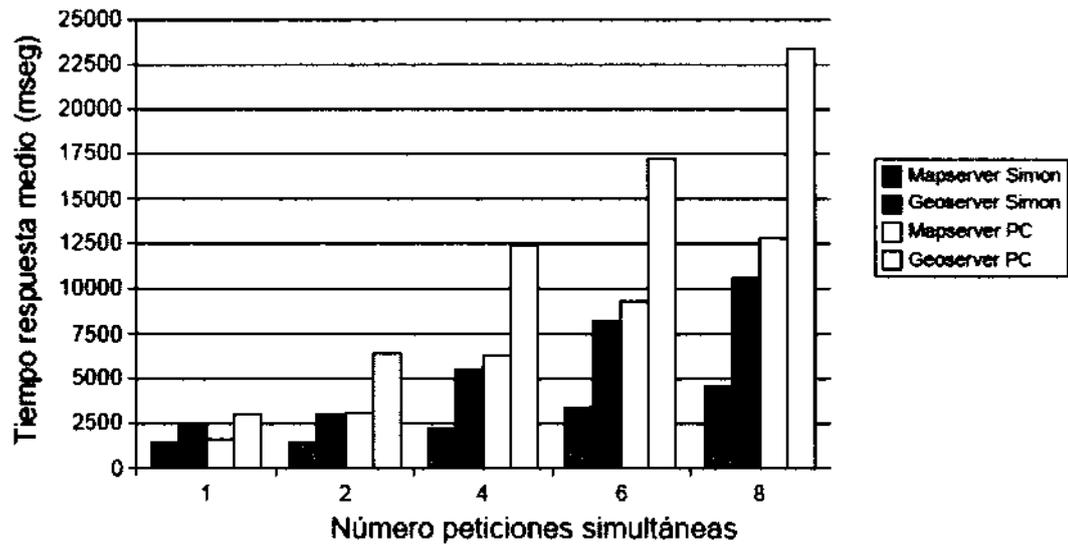


Anexo 2 – Comparación de Mapserver y Geoserver en acceso aleatorio

Acceso aleatorio. PC linux



Anexo 3 – Comparación de Mapserver y Geoserver ante peticiones simultáneas



Anexo 4 – Arquitectura de UMN Mapserver

