

UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
FACULTAD MATEMÁTICA, FÍSICA Y COMPUTACIÓN



MAESTRÍA
"CIENCIAS DE LA COMPUTACIÓN"

Título: "Servidor de Objetos Geográficos para Internet, basados en las especificaciones del Consorcio Internacional OpenGIS".

Autor: **Lic. Osmani Herrera González.**

Tutor: **Dr. Carlos Pérez Risquet.**

Santa Clara, Julio del 2005
"Año de la Alternativa Bolivariana para las Américas"



El Servidor de Objetos Geográficos (*WFS*, por sus siglas en inglés) es la especificación de Interfase de OpenGIS, liberada en Septiembre del 2002 que permite al cliente obtener datos geoespaciales codificado en GML (Geography Markup Language) desde múltiples servicios de objetos en Web; similar al Servicio de Mapas en Web (WMS), que permite al cliente obtener imágenes de mapas para ser visualizados por múltiples servicios de mapas en Internet.

El WFS habilita tecnologías para el acceso directo a objetos geográficos en almacenes de datos remotos y distribuidos en un formato estándar.

Permite acceder de forma directa a objetos geográficos como Puntos, Líneas y Polígonos, bajo el uso de restricciones espaciales o temáticas.

Un WFS implementa las operaciones **GetCapabilities**, **DescribeFeatureType** y **GetFeature** y es considerado un WFS de solo lectura, el cual es el primer paso para llegar al WFS de Negociación o Transaccional que no es más que un WFS básico que implementa la operación **Transaction** (Insert, Delete, Update y LockFeature).

La operación **GetCapabilities** devuelve un documento XML que brinda información sobre el servicio, la lista de solicitudes que el WFS puede manipular, el listado de tipos de objetos que están disponibles en el WFS y las operaciones definidas para el filtro, en caso de que este exista.

La operación **DescribeFeatureType** devuelve el esquema en XML que describe la estructura de todos los tipos de objetos que el WFS puede manipular.

La operación **GetFeature** devuelve un documento GML que sirve las instancias de objetos solicitadas en la consulta al WFS.

La operación **Transaction** es usada para describir las operaciones de transformación de los datos que son aplicadas a las instancias de objetos accesibles por la Web. El WFS puede realizar la operación de **Transaction** directamente o traducirlo al lenguaje del gestor de datos con el cual se conecta y esta a su vez ejecutar la transacción. Al ser completada la operación, el WFS genera como



respuesta un documento XML, indicando el estado de completamiento de la transacción.

Esta implementación es un paso lógico para describir las operaciones de manipulación de datos en objetos geográficos, usando el protocolo http, la cual incluye la habilidad de obtener o consultar objetos basado en restricciones generadas por el cliente y pasadas al WFS usando este protocolo. Esto brinda la *factibilidad técnica* de acceder a la información ubicada en diferentes almacenes de datos y la *facilidad* de usar estas factibilidades técnicas para presentar los datos al cliente en una aplicación visual, útil y significativa como son los servidor de mapas basado en las especificaciones de OpenGis; como es el caso el servidor de mapas desarrollado por nuestro grupo.



A Web Feature Server is the Open GIS Interface specification, liberated in September of the 2002 that allows a client to retrieve geospatial data encoded in Geography Markup Language (GML) from multiple Web Feature Services, similar to the Service of Maps in Web (WMS) that allows the client to overlay map images for display served from multiple Web Map Services on the Internet.

The WFS enables technologies for the direct access to geographical objects in remote and distributed data base. It allows direct access to a geographical object as Points, Lines and Polygons, under the use of space or thematic restrictions.

A WFS implements the GetCapabilities, DescribeFeatureType and GetFeature operations and it is a readonly WFS. This is the first step to arrive to the Negotiation or Transaccional WFS, which is WFS that implements the operation Transaction (Insert, Delete, Update and LockFeature).

GetCapabilities

A web feature service must be able to describe its capabilities. Specifically, it must indicate which feature types it can service and what operations are supported on each feature type.

DescribeFeatureType

A web feature service must be able, upon request, to describe the structure of any feature type it can service.

GetFeature

A web feature service must be able to service a request to retrieve feature instances. In addition, the client should be able to specify which feature properties to fetch and should be able to constrain the query spatially and non-spatially.



Transaction

A web feature service may be able to service transaction requests. A transaction request is composed of operations that modify features; that is create, update, and delete operations on geographic features.

This implementation is a logical step to describe the operations of manipulation of data in geographical objects, using the protocol http, which includes the ability to obtain or to consult objects based on restrictions generated by the client and pass to the WFS using this protocol. This offers the technical feasibility to access to the information in different data bases and the easiness of using these technical feasibilities to present the data to the client in a visual, useful and significant application as they are the maps server, based on the OpenGis specifications; for example the maps server developed by our group.



El acceso a la información geográfica sobre Web a través de navegadores comunes (“browsers”) ha crecido tan vertiginosamente como la propia red de redes y los servicios WWW (*World Wide Web*). La presentación de mapas de propósito general como herramientas sofisticadas interactivas y personalizables y la recuperación de información espacial rápida y de forma fácil por muchos usuarios, requiere de herramientas de lectura de Mapas y de Bases de Datos, para poder realizar análisis visuales o la recuperación en directo desde la Base de los Objetos Geográficos y de sus rasgos; sin la alta complejidad y el alto costo que implican por lo general los software de sistemas profesionales que manipulan este tipo de datos.

En este primer capítulo realizaremos una descripción de la problemática antes planteada en nuestro país y las principales líneas de trabajo seguidas para sumarnos a este desarrollo mundial, el cual describiremos en función de las iniciativas del Consorcio Internacional OpenGIS (OGC). Seguidamente se expresan los objetivos que nos trazamos con esta investigación y los trabajos que desarrollaron con anterioridad y que sirven de base para la realización de la misma. Finalmente se brinda un pequeño resumen de cómo está organizado el presente trabajo de tesis.

1.1 Descripción del Problema.

Nuestro país no dispone en estos momentos de softwares y herramientas propias que permitan la inserción eficiente en el mundo de la georeferenciación de nuestras informaciones y su publicación en Internet; y la adquisición de las aplicaciones profesionales para ello, se hacen en extremo difícil por su costo monetario y en muchos casos político, ya que la mayoría de ellos son de manufactura norteamericana y está limitada la venta para nuestro país.

Además, se hace necesario introducir otro de los elementos que hoy se aplica universalmente, el cual consiste en la interoperabilidad de la información, los procesos y las aplicaciones, que rige el desarrollo de los nuevos productos de software, base sobre la que se han elaborado estándares y especificaciones públicas para desarrolladores.



Por tanto el problema que resuelve este trabajo es la implementación de un Servicio Web para la recuperación de Objetos y Rasgos Geográficos, lo cual se convierte en una nueva herramienta cubana basada en los estándares del Consorcio internacional OpenGIS para integrar datos geográficos desde múltiples y heterogéneos servidores de Internet y presentar estos como imágenes de mapas o documentos en formatos interoperables. Este resultado, desarrollado dentro del campo de prueba de OGC para los servicios Web, soporta la presentación de reportes de las diferentes bases de información geográfica disponibles en nuestro país, la implementación de métodos estándares de acceso y transferencia de información geográfica y el aseguramiento de los diferentes servicios que están surgiendo y se vienen implementando en el ambiente Web y sobre las herramientas de los Sistemas de Información Geográfica (SIG).

1.2 Entorno Mundial y cubano donde se desarrollan las iniciativas de OGC.

El Consorcio OpenGIS [1] emana de los grupos de trabajo que definieron y estandarizaron en sus orígenes (años 80) el mundo de los Sistemas de Información Geográfica. Se formalizó posteriormente en el año 1994 como consorcio. Actualmente pertenecen a este consorcio socios Industriales, Gubernamentales, Universitarios y los propios socios fundadores. La principal misión del consorcio ha sido intentar estandarizar los servicios relacionados con la geografía en entornos distribuidos. Para ello han elaborado una serie de especificaciones abstractas que satisfacen el primer nivel de las necesidades propias de los SIG: interpretación de los datos geográficos. Algunos servicios estandarizados son: los Servicios de Mapas (WMS, siglas en inglés de *Web Map Services*), los servicios de objetos (WFS, siglas en inglés de *Web Feature Service*) y los servicios de coberturas (WCS, siglas en inglés de *Web Coverage Service*). Los miembros del OGC también han estado trabajando en la especificación de servicios de mayor nivel como servicios de geocodificación (*Geocoders*, *Gazetteers*, etc.) y el lenguaje para la representación e intercambio de los datos GML (*Geographic Mark Language*). En el OGC existe un grupo dedicado a la especificación de la interfaz del catálogo de metadatos. La arquitectura que están utilizando en el OGC define un número elevado de servicios,



algunos de los cuales están sin especificar, todo ello con el objetivo final de permitir el ensamblado de los servicios en entornos distribuidos. Para poder satisfacer este requisito hace falta que los servicios sean interoperables. Las especificaciones del OGC más consolidadas y validadas son: WMS y WFS. Algunas de estas especificaciones datan de los años 90 y están definidas de forma que el servicio se ofrezca a través del protocolo HTTP con una notación de parámetros orientada a un script o cgi. Todos los servicios poseen un método por el que se describe las capacidades del mismo en forma de fichero marcado en XML. Recientemente ha surgido un grupo de trabajo en el OGC cuyo objetivo es analizar las tendencias del software en los entornos de los Servicios Web, para decidir las directrices futuras de las especificaciones de los servicios del OGC para proporcionar la interoperabilidad. Este grupo de trabajo ha recibido el nombre de OWS (*OpenGIS Web Services*) y se está apoyando en la norma ISO-19119 para los SIG. La misión de las comisiones técnicas del OGC consiste en el consenso de propuestas, asociadas a los servicios relacionados con el tratamiento de la información geográfica en entornos distribuidos. El principal problema con el que tropezaron los miembros de las comisiones, es la falta de consenso en un estándar para la representación de la información geográfica (principalmente cartografía), común para todos ellos. Existe y se usa como estándar de intercambio de información cartográfica en ficheros ASCII: el formato DXF de *Autodesk*, o los ficheros *Shape* de *ArcInfo* cuando se asocia información tabular a las entidades gráficas. Estos han sido utilizados como formatos de intercambio, pero tienen dueño. El OGC comenzó tratando de lograr un consenso y formalizando un formato para el almacenamiento de información vectorial independiente de cualquier firma comercial; así apareció la especificación del *Geographic Mark Language* (GML). Otra de las primeras iniciativas del OGC, si no la primera, fue la definición de una especificación para el servicio de interpretación de la información cartográfica, tanto vectorial como ráster y que soportara la gestión de la información por capas (solapes, transparencias, rótulos, ...): *Web Map Server* (WMS). Este servicio es el más estudiado, implementado y probado de todos los que han surgido dentro del OGC. Además adolece de limitaciones relacionadas con el acceso a información de coberturas geográficas, ya que al no ser consideradas entidades no se puede



solicitar información de las mismas; por esta razón en el grupo de desarrollo de software de GEOCUBA, está trabajando con la especificación del servidor de coberturas.

Otra limitación fuerte de este servicio es la extracción, actualización o modificación de la información geográfica vectorial. Par dar solución a estos entornos, en los que se necesite este tipo de servicio, se ha definido la especificación del WFS, cuyo formato nativo de datos es GML. Existen otros servicios en un estado avanzado de especificación como son: el servicio de descripción de los estilos de las capas SLD (*Style Layer Descriptor*) y el servicio de acceso al catálogo de los recursos geográficos, de modo que se pueda localizar y acceder a la información distribuida. Los estándares de metadatos que se manejan aún no son definitivos, si bien ISO tiene un estándar en estado de borrador cuya fecha de aprobación definitiva está próxima. Se han hecho grandes esfuerzos internacionales en el consenso de los estándares, partiendo de la iniciativa Americana del FGDC e iniciativas Europeas como el Dublin Core, para llegar a un estándar ISO.

En Cuba, el Grupo Empresarial GEOCUBA es la entidad líder en la producción y manipulación de Información Geográfica, posee una rica experiencia en el trabajo de implementación de los estándares del OGC y dispone en estos momentos de un conjunto de herramientas y aplicaciones que permiten el servicio de mapas en Web; ejemplo de ellas es el Servidor de Mapas para el servicio de imágenes de mapas en Web.

Estas herramientas están sustituyendo paulatinamente el uso de otros software profesionales que son necesario importar y dan solución a grandes perspectivas acometidas por la máxima dirección del país como son el caso de los SIG para las administraciones municipales y el portal cubano de la Infraestructura de Datos Espaciales de la República de Cuba (IDERC) [2], tarea de gran importancia asignada por el Consejo de Ministros a la Dirección Nacional del Grupo Empresarial GEOCUBA y la Oficina Nacional de Hidrografía y Geodesia (ONHG).



1.3 Objetivos del trabajo

Se quiere realizar un proyecto, que involucre el trabajo desarrollado en el grupo de software del Grupo Empresarial GEOCUBA. Así mismo, se pretende que este proyecto forme parte de los servicios que aquí se desarrolla y se utilice como una solución a los problemas que se presentan con la transportación y utilización de mapas en la Web.

1.3.1 Objetivo general

Diseñar, Estructurar e Implementar un Servicio Web de Objetos Geográficos para Internet, basado en las especificaciones del Consorcio Internacional OpenGIS, que amplíe las prestaciones que hoy se realizan con el servidor de mapas del Grupo Empresarial GEOCUBA en cuanto a la publicación de mapas en Internet mediante el Lenguaje de Marcas Geográficas (GML), con vistas a disponer de un nuevo servicio para publicar y consultar Información Geográfica en la Web y mejorar la transmisión de la información y el acceso a los datos primarios en el servicio de mapas actual.

1.3.2 Objetivos específicos

1. Desarrollar un trabajo de investigación teórica para la creación de un Servidor de Objetos Geográficos y su publicación en formato GML, de acuerdo a las tendencias actuales de desarrollo de los SIG en Web según OGC y las características específicas de Cuba.
2. Implementar un Servicio Web de Objetos Geográficos, basado en estándares internacionales.
3. Mejorar la transmisión de la Información Geográfica y la del acceso a los datos primarios, para los casos en que estas variables sean una limitante para el actual Servicio de Mapas.
4. Reducir las posibles desventajas en seguridad que presenta la implementación del nuevo servicio mediante el Lenguaje de Marcas Geográficas.



1.4 Trabajo Previo

La realización de este proyecto investigativo fue antecedido por otros, que sirvieron de base para poder lograr una arquitectura y funcionamiento que se rigiera por los estándares internacionales que crea el OGC para la creación, almacenamiento, manipulación y publicación de la cartografía.

1.4.1 Campo de prueba de la Cartografía en Web del OGC.

El Consorcio OpenGIS que trabaja estrechamente con ISO TC/211 (Geografía/Geomática) [3] ha desarrollado un grupo de especificaciones de interoperabilidad, logrando que las componentes de software operen recíprocamente para acceder a recursos distribuidos y eliminar las barreras impuestas por ambientes de procesamiento heterogéneos y datos de fuentes igualmente heterogéneas.

Durante la iniciativa de este consorcio, *Web Mapping Test* (WMT1), se desarrollaron dos especificaciones:

- *OpenGIS® Web Map Service Implementation Specification.*
- *OpenGIS® Geography Markup Language (GML) 2.0 Implementation Specification.*

El documento que define a los WFS, surge como un siguiente paso a esta iniciativa y propone interfases para describir operaciones de manipulación de datos sobre objetos geográficos usando HTTP como plataforma de cómputo distribuido.

En la actualidad existen aplicaciones comerciales y no-comerciales que implementan WFS de una forma robusta y escalable, en conformidad con el OGC, las que se consideran entidades que desarrollan tecnologías de punta. Ejemplo de ello son el WFS del “*Canada Centre for Remote Sensing*” desarrollado por CubeWerx [4] y el WFS de “*NSW Department of Land and Water Conservation*” de Australia [5].



1.4.2 Servidores de Mapas.

El trabajo más importante que se está sistematizando actualmente en el mundo respecto a los Visores de mapas online es el producto del Consorcio OpenGIS, específicamente *Web Mapping Testbed Specifications* (WMT). Ellos han liberado un conjunto de especificaciones de software que posibilitan que los usuarios puedan combinar capas temáticas de diferentes fuentes y formatos, con un enfoque basado en la interoperabilidad.

El WMT es la primera iniciativa de interoperabilidad planificada que incluye auspiciadores y participantes [6]. A medida que los productores de software geoespacial implementen estos estándares abiertos, los usuarios del Web fácilmente encontrarán visores, análisis y combinaciones de mapas temáticos para diferentes regiones.

David Schell, presidente de OGC, dijo “Así como crece el valor de los “*browsers*” del Web con el tamaño del Web, así mismo el valor de un SIG o colección “*on-line*” de datos geográficos se incrementa con el número de software basados en Web que pueden acceder a estos datos a través de interfases OpenGIS.

Pronto los usuarios de Web, podrán usar sus “*browsers*” para acceder, ver y explotar los amplios y diversos recursos distribuidos de geoprocesamiento y datos geospaciales en el Web. Esto tendrá un profundo efecto en el mercado para servicios y datos geoespaciales.

Las especificaciones comunes de los “Visores de mapas Web” están basadas en la combinación de protocolos existentes (WWW3, ISO) y otros que se encuentran en fase de desarrollo (que serán formalizados por grupos tales como ISO). Dichas especificaciones ofrecen la única forma de posibilitar la combinación visual de información geográfica distribuida (mapas) y compleja sobre Internet.

Las acciones mínimas que involucra un Visor de mapas Web [7] son las siguientes:



1. Un cliente hace una solicitud a uno o más Servidores de Catálogos (basados en las especificaciones de Servicios de Catálogos de OpenGIS) para obtener los Urls que contienen la información deseada.
2. Los Servidores de Catálogos retornan URLs y la información acerca de los métodos mediante los cuales puede ser accedida la información de cada URLs.
3. El cliente localiza uno o más servidores que contienen la información deseada (ej., usando la tecnología de servidores de catálogos de OpenGIS), y los invoca simultáneamente.
4. Cada Servidor de Mapa accede a la información solicitada desde éste y la ensambla (*render*) adecuadamente para mostrar una o más capas en un mapa compuesto de muchas capas.
5. Los Servidores de Mapas proveen la información lista-para-mostrar al cliente o clientes, los cuales entonces la muestran. Los clientes pueden mostrar información desde muchas fuentes en una sola ventana.

Las primeras especificaciones de OpenGIS para Visores de Mapas Web especifican los protocolos de solicitud y respuesta para interacciones entre Servidores de Mapas y Clientes basados en Web, y están soportadas sobre las especificaciones ya anteriormente liberadas por OGC, tales como *Simple Feature* [8] y *Catalog Services* [9]; así como, los estándares de metadatos de ISO que brindan las bases para construir ambientes abiertos y robustos para visores de mapas Web.

La segunda fase de estas especificaciones (*Web Mapping Testbed Phase 2*) centrarán su atención fundamentalmente en información codificada en XML, trabajo adicional sobre catálogos y servicios de búsqueda y para transportar información codificada en XML a través de Internet. Adicionalmente, las fases futuras de *Web Mapping Testbed* serán dirigidas a cuestiones relativas a la seguridad y el comercio electrónico entre otros importantes tópicos.



Servidores de Mapas.

Como se ha visto, para que un visor de mapas Web funcione, tienen que estar interconectados una serie de servidores de mapas a través del uso de protocolos comunes si estos están en un escenario Intranet, Extranet o Internet.

Por lo general, un servidor de mapas puede hacer lo siguiente [7]:

- Producir un mapa (como una imagen, una serie de elementos gráficos, o como un conjunto empaquetado de datos)
- Responder consultas básicas acerca del contenido de un mapa
- Decirle a otros programas qué mapas puede producir y cuáles de ellos pueden después ser consultados.

Un navegador estándar puede ordenar y recuperar información espacial solicitándola a los Servidores de Mapas en la forma de URLs.

Clasificación de los Servidores de Mapas.

Existen dos formas de clasificación de los servidores de mapas, la primera es teniendo en cuenta el grado de procesamiento que realiza el cliente y la segunda es de acuerdo de la información que se traslada entre el cliente y el servidor. Por otra parte existen 4 etapas de procesamiento que ocurren cuando los datos se mueven desde la fuente de datos hacia display que es donde se hacen visibles para usuario:

1. Filtrado de los elementos geográficos
2. Generación de los elementos a visualizar
3. Conformación de la imagen
4. Visualización

Debido al tipo de información que se mueve entre el cliente y el servidor, la forma en que esta es empaquetada y buscando la forma de utilizar otros términos que dieran



una mejor idea del problema las especificaciones OpenGIS definieron 3 casos: el caso Imagen, el caso Elementos Gráficos y el caso Datos.

Caso Imagen: Lo que viaja a través de Internet en respuesta a una solicitud del cliente no es más que una imagen de un mapa construida por el servidor y que puede ser una imagen a GIF, JPEG o PNG.

Caso Elemento Gráfico: Lo que viaja entre el servidor Web y el cliente es un conjunto empaquetado de elementos individuales, en un sistema de referencia proyectado y con sus atributos de representación. Los formatos de elementos gráficos más usados son SVG y WebCGM.

Caso Dato: Da la posibilidad de enviar datos sobre el objeto geográfico del servidor al cliente. La codificación de estos datos mediante XML ha dado como resultado la especificación de implementación de OGC para GML”.

Interfases Básicas de un Servidor de Mapas.

Un Servidor de Mapas debe exponer tres interfases, ellas son: *Capabilities*, *Map* y *FeatureInfo*, Al menos las dos primeras son obligatorias en el proceso de obtención de un Mapa y la tercera aunque es opcional es de gran importancia cuando se habla de desarrollar Sistemas de Información Geográficos basados en Web.

En sentido General las tres interfases anteriores serán expuestas a través de una mas general que se nombra "*Request*" mediante ella se le dirá al servidor que interfaz se está encuestando.

ej: ...Request=capabilities...

Además ésta interfaz general tiene que llevar como parámetro la versión de servidor de mapas que se solicita y puede o no tener alguna información adicional del usuario.



La interfaz Capabilities es requerida en un servidor de mapas. La misma está diseñada para proveer una lista de que interfases soporta el servidor de Mapas, que capas de mapas puede servir, que formatos y otros detalles.

Un servidor de Mapas debe siempre implementar esta interfaz o asegurarse que existe otro servidor que puede proporcionar la respuesta a una solicitud de esta interfaz. Si no hay medio de que un cliente acceda a la interfaz capabilities entonces el Servidor de Mapas no será un servidor con un buen formato. Internamente un Servidor de Mapas puede escoger entre generar dinámicamente una respuesta a Capabilities o simplemente retornar un fichero XML con la respuesta.

La interfaz Map es de obligatoria implementación y mediante ella se puede obtener el mapa deseado. Esta interfaz esta diseñada para proporcionar a los clientes del Servidor de Mapas con imágenes de mapas, posiblemente de múltiples servidores. Una vez recibida una solicitud de mapa el Servidor debe satisfacerla o generar una excepción acorde a las especificaciones de OpenGis para cada caso de Servidor de Mapas.

El resultado a una solicitud de la interfaz Map de un Servidor de Mapas debe ofrecerse en el formato descrito por el campo FORMAT de la interfaz, de lo contrario debe generarse la excepción correspondiente.

La interfaz FeatureInfo está diseñada para proporcionar más información a los clientes de un Servidor del Mapa sobre los elementos en los mapas que fueron resultado de solicitudes de Mapa anteriores. Básicamente la interfaz le ofrece a un cliente la posibilidad de especificar sobre qué píxel está preguntando, sobre qué capa(s) debe preguntarse, y en qué formato debe devolverse la información. Para proporcionar un protocolo sin estado, la demanda del Mapa es una de las partes de una solicitud de FeatureInfo. El uso principal de FeatureInfo que es que un usuario ve el resultado de una solicitud de Mapa y sobre ese mapa escoge un punto para conseguir más información. La semántica real de cómo un Servidor del Mapa decide sobre qué información devolver, queda a al diseñador de Servidor de Mapa.



Cliente del Servidor de Mapas.

En la cara cliente el visor depende del caso de Servidor de Mapas que se implemente. En este caso en que los mapas del servidor son de tipo imagen cualquier navegador que permita visualizar el formato de imagen que devuelve el servidor puede servir como cliente. Solamente es necesario preparar las consultas adecuadas para que cumplan con las interfases del servidor antes expuestas.

El caso se complica cuando se sirven elementos tipo dato. En este caso es necesario poseer un visor para el formato que se escogió para el transporte. En nuestro caso que utilizamos SVG para las primeras pruebas utilizamos un visor producido por Adobe que se descarga gratis desde el sitio <http://www.adobe.com/svg/>. El trabajo con este componente nos es de gran ayuda ya que sirve como prueba adicional al formato del contenido SVG que genera dinámicamente nuestro servidor Web.

1.5 Organización de la tesis.

El resto de este documento está organizado en tres capítulos de la siguiente forma:

Capítulo II: Marco Teórico. En este capítulo se explican las principales concepciones teóricas que fundamentan y soportan la creación de de las herramientas de publicación de cartografía en Web. Se tratan conceptos como Especificaciones de Servicio según OGC, Sistemas de Información Geográfica, Estándares de manejo de la Información Geográfica y los elementos que demuestran la componente científica de la visualización de este tipo de información.

Capítulo III: Servidor de Objetos Geográficos. En este capítulo se describe como se realiza el acceso a los datos almacenados y la implementación de las diferentes especificaciones de OGC, GML y FILTER, que son necesarias para conformar el servicio de Objetos geográficos y la implementación de la especificación del servicio en si, WFS.

Capítulo IV: Resultados y Conclusiones. Este capítulo recoge, en breve síntesis, los diferentes escenarios donde se introducido el servicio en forma de prueba, así como



las principales prestaciones que se han realizado con el mismo como herramienta en funcionamiento. Se relacionan además los principales resultados alcanzados con este proyecto investigativo, las perspectivas a seguir y las principales conclusiones a que hemos llegado con el mismo.



Para poder llegar a comprender las bases teóricas que fundamentan y soportan la creación de las herramientas de publicación de cartografía en Web, es necesario adentrarse en ese mundo, el cual está regido hoy por los lineamientos que traza el Consorcio Internacional OpenGIS. Estos a su vez se expresan en especificaciones de implementación para desarrolladores de aplicaciones, con lo cual se lograría que las producidas sean interoperables.

También es importante conocer elementos básicos relacionados con los Sistemas de Información Geográfica, pues el fin que se persigue es fortalecer en el funcionamiento de los mismos en ambientes Web.

Finalmente es necesario exponer aquellos elementos que muestran las componentes científicas de la visualización de este tipo de información.

2.1 Especificaciones de Implementación de Servicios de OpenGIS.

El Proceso de Desarrollo de Tecnologías de OGC [10] crea dos tipos de especificaciones: las abstractas y las de implementación. El propósito de las especificaciones abstractas es crear y documentar un modelo conceptual suficiente para permitir la creación de las especificaciones de implementación. Las mismas son plataformas de especificaciones de tecnologías precisas para la implementación de estándares industriales y para la programación de interfaces de aplicación de software.

El OGC usa un proceso de consenso general, que logra entre sus miembros, con el objetivo de lograr las especificaciones para componentes del software relacionados a los Sistemas de Información Geográficas y Geoprocesamiento.

Las Especificaciones Abstractas se desarrollan a través de las actividades de Grupos de Intereses Especiales y los Grupos de Trabajo miembros del consorcio. Éstas se evalúan en el Comité Técnico donde finalmente se logra el acuerdo. Cuando un tema de una especificación abstracta está suficientemente logrado, los miembros de OGC pueden emitir una solicitud de confección de la especificación de implementación. En



respuesta se obtiene una primera aproximación de las especificaciones en una plataforma neutral. Si la especificación de implementación es aceptada por los miembros de OGC, la misma pasa a formar parte del conjunto de especificaciones del consorcio y se prueba su conformidad con software seleccionados por miembros del mismo. Y si se logra la misma, se distingue la especificación con la marca del OGC.

Al disponer de las especificaciones de implementación para desarrolladores, se resuelve el problema de la estandarización de los servicios relacionados con la geografía en entornos distribuidos y la representación de la información geográfica (principalmente cartografía), en una forma común para todos. En la actualidad existe y se usa como estándar de intercambio de información cartográfica en ficheros ASCII: el formato DXF de Autodesk, o los ficheros Shape de ArcInfo cuando se asocia información tabular a las entidades gráficas. Estos han sido utilizados como formatos de intercambio, pero tienen dueño, por lo que para un grupo considerable de desarrolladores de estas tecnologías en el mundo, es muy importante la implementación de sus aplicaciones sobre estas bases de fundamentos tecnológico probados y libres para el uso de todos.

El trabajo desarrollado se basó en el estudio de las especificaciones abstractas y de implementaciones relacionadas a un Servidor de Objetos Geográficos y libres en el sitio de este consorcio, para lograr la implementación de aquellas que dieron lugar al primer servidor cubano de objetos geográficos interoperable con las demás aplicaciones estándares con OpenGIS en el mundo.

2.2 Sistemas de Información Geográfica en Web.

La tecnología actual, en particular la informática, permite desarrollar sistemas automatizados de información de gran capacidad; en este ámbito, se han generado herramientas para la manipulación computarizada interactiva de mapas u objetos que tienen atributos espaciales, estos se denominan Sistemas de Información Geográfica (SIG), cuya característica primordial es que permiten conservar la referencia territorial de la información.



Ligados en sus orígenes al manejo de grandes bases de datos y a la cartografía automatizada, los SIG han desbordado esos campos y situado su función en el manejo y análisis de la información que los define. Al respecto, Burrough [11] ha señalado que los SIG son resultado de la amalgama de desarrollos de cómputo llevados a cabo por diversas disciplinas y técnicas relacionadas con el procesamiento de datos espaciales (cartografía, fotogrametría, tecnología de sensores remotos, geometría computacional, representación gráfica, entre otros).

Como herramienta técnica desarrollada en el campo del quehacer geográfico, los SIG se han destacado por el hecho de facilitar las tareas básicas características de la Geografía: análisis, integración y síntesis de los procesos espaciales.

Los SIG son instrumentos tecnológicos de capacidades múltiples, diseñados y habilitados para registrar y almacenar información geográfica, a partir de la cual desarrollan y ejecutan la serie de funciones de análisis espacial que los distinguen. Son, en consecuencia, herramientas útiles a todas aquellas labores relacionadas con la planeación, el ordenamiento y la administración de procesos y actividades con clara expresión territorial. Con el empleo de un SIG es posible observar gráficamente la localización de objetos, hechos o fenómenos que tengan una expresión espacio – temporal; por ejemplo, las áreas de influencia, las relaciones geográficas y las tendencias regionales. Con ello se pueden realizar correlaciones de variables sociales y ambientales, calcular distancias y áreas; diseñar estrategias, construir modelos matemáticos, identificar rutas de acceso o evacuación y estimar las necesidades de equipamiento urbano o de infraestructura en general para una localidad o región.

Los SIG cuentan con facilidades para manipular distintas clases de objetos en forma diferenciada, así como para relacionar las clases entre sí y formar jerarquías de objetos entrelazados. Además de la información geográfica, que incluye localización, morfología y estructura, los objetos geográficos tienen información asociada de carácter no geográfico, organizada en atributos. Para que las bases de datos estadísticas tengan sentido de análisis espacial, deben estar ligadas con las bases



de datos geográficas, ya que de no ser así se estaría desperdiciando la potencialidad del SIG y no se aprovecharían las virtudes de la representación territorial.

2.2.1 Sistema de Información Geográfica (SIG).

Un Sistema de Información Geográfica (SIG) permite ver el mundo y lo que hay en él con una perspectiva distinta.

En materia computacional, cuando hablamos de consultas, pensamos en la clásica consulta a la base de datos, haciendo uso de “*select*” en SQL. Por ejemplo, selecciona todas las ciudades que fueron fundadas entre 1959 y 2005. En una base de datos con las ciudades y los años en las que fueron fundadas, esta selección nos reportaría las ciudades que fueron fundadas durante el proceso revolucionario.

Pero, ¿qué sucede cuando la consulta que esperamos realizar es sobre la relación geográfica que existe entre los datos? Por ejemplo, queremos seleccionar los ríos que pasan por las ciudades antes seleccionadas, o las que tienen un área superior a 3 kilómetros cuadrados. En este caso, ¿cómo se realiza la consulta?, ¿Cuál sería una forma sencilla para capturar los requerimientos del usuario? Además, si consideramos común los tipo de preguntas anteriores, cabría preguntarse: ¿es necesario contar con un sistema especializado en este tipo de datos?, o ¿Podemos tener acceso remoto a ellos?

Los Sistemas de Información Geográfica pueden incluir información física, biológica, cultural, demográfica, o información económica. Son una herramienta valiosa en las ciencias naturales, sociales, medicina y la ingeniería, así como en los negocios y en la planificación. [12].

Históricamente el concepto de Sistema de Información Geográfica ha recibido varias definiciones por especialistas dedicados al tema. Entre las más sobresalientes encontramos:

- *“Un Sistema de Información Geográfica es un tipo especializado de base de datos, que se caracteriza por su capacidad de manejar datos gráficos, es*



decir, espacialmente referenciados, los cuales se pueden representar gráficamente como imágenes” [13].

- *“Un sistema de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión” [14].*

Estos sistemas nos permiten administrar diferentes recursos (energía, tierra, flora, fauna, por ejemplo), facilitando la planeación de estos recursos y la toma de decisiones. El centro de atención de estos sistemas son los usuarios y sus capacidades analíticas, ya que la efectividad depende de cómo los usuarios puedan requerir la información espacial y en que forma se presente a ellos [15].

En un sentido estricto un SIG es un sistema por computadora capaz de congrega, almacenar, manipular y desplegar información geográficamente referenciada (por ejemplo, datos que son identificados según sus localidades). Un SIG incluye los datos y el personal que lo opera [16].

Se dice que estos sistemas forman una clase distinta de sistemas de información por sus requisitos únicos, ya que deben permitir manipular cualquier dato que pueda ser referenciado por su localidad [17].

2.2.2 Características de un Sistema de Información Geográfica.

Pissinou realiza un análisis sobre las características que debe tener un SIG [18] y afirma que para la funcionalidad de estos sistemas son necesarias diferentes características, independientemente del ambiente al que de soporte. Entre estas características destaca el soporte a la toma de decisiones, por medio de módulos especializados.

Por otro lado, Ramesh Subramanian y Nabil R. Adam [19] concluyen que dentro del conjunto de funcionalidades de un modelo de datos espaciales se deben soportar diferentes características. Entre ellas está la facilidad para representar las relaciones



entre los objetos espaciales y la implementación de operadores espaciales tales como "cerca-de," "entre" o "adyacente-a".

Otros autores como Max Egenhofer [20], Bernard Jansen [21] y Gísli Hjaltanson [22], se han dedicado al desarrollo de nuevos modelos que permiten el análisis de datos espaciales, por la importancia que tiene el apoyo a los usuarios durante el proceso de toma de decisiones.

Modelo de datos

Muchos SIG son organizados como una colección de temas o capas, como se muestra en la figura 1. Cada capa representa los valores de un atributo único del espacio geográfico y puede fragmentarse por espacios, llamados objetos o entidades, los cuales son almacenados en la base de datos.

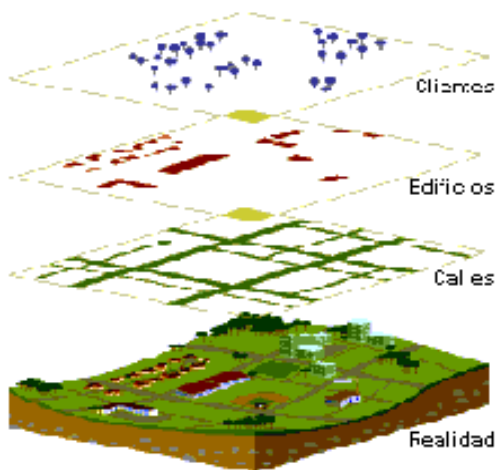


Figura 1. Colección de Temas o Capas

Para cada entidad geográfica se identifica la geometría y la topología. La geometría se refiere al conjunto de propiedades geométricas como son medidas: relación entre puntos, líneas, ángulos y superficies. La geometría puede ser vectorial o ráster.

La geometría vectorial esta basada en puntos, líneas y polígonos, los cuales representan el espacio ocupado por entidades del mundo real. La ráster maneja un

La plataforma creada en este trabajo está limitada a los datos de tipo vectorial y en la actualidad se desarrolla un proyecto para incluirle el manejo de datos en formato ráster.

Un SIG vectorial está basado en la representación geométrica explícita de la componente espacial de los datos geográficos y, con la descripción digital de sus características espaciales, llevan asociados un conjunto de aspectos temáticos, véase la figura 2. Algunos almacenan los datos espaciales y tabulares en bases de datos separadas, otra forma es mediante una organización híbrida, tratando ambas partes como una única base de datos.

[illegible]



Figura 2. Componente espacial y temática de la base de datos de un SIG vectorial.

Componente espacial de un SIG Vectorial.

La componente espacial de los datos geográficos tiene como base la figura, siendo una figura la representación abstracta de un fenómeno del mundo real. Las figuras geográficas usualmente almacenan información que determina su posición relativa a las coordenadas terrestres, o relativas a otras. La técnica más común para dar a conocer la forma y posición de una figura es a través de una geometría.

Una geometría es la combinación de una geometría de coordenadas y un sistema de referencia. La geometría está compuesta por algunos de los siguientes elementos:

1. Una secuencia de puntos coordinados del mismo sistema de referencia.
2. Una colección de otras geometrías pertenecientes al mismo sistema de referencia.
3. Un algoritmo de interpretación (proyección) que usa estas geometrías y puntos coordinados para "construir" una entidad geométrica de coordenadas que define indirectamente la extensión de la geometría en tiempo y espacio. Nótese que las entidades geométricas pueden estar compuestas por otras, y muchas entidades pueden compartir otras como componentes.
4. Un sistema de referencia espacial, que establece una relación entre la geometría de coordenadas y la ubicación, dando a la geometría una interpretación correcta del mundo real.

Las geometrías son figuras representables en un plano como: puntos, segmentos, áreas, o colecciones de ellas. Entre ellas se establece una estrecha interrelación llamada topología. Por ejemplo, un segmento se determina por dos puntos, una superficie se define como un conjunto de líneas que la limitan. Otras nociones de relación entre geometrías que también pertenecen al concepto de topología son, por ejemplo, adyacencia entre áreas, conectividad entre elementos, interior a un área. Estas relaciones se optimizan si la topología forma un grafo plano, o sea, ningún arco



se corta con otro excepto que se encuentren en un vértice o nodo, y ninguna área se corta con otra excepto las que tengan un arco en común.

Sistema de referencia espacial.

Las expresiones de relaciones espaciales entre las figuras, y las mediciones de distancia, dependen del conocimiento de una caracterización de las ubicaciones o posiciones. Los sistemas de coordenadas están diseñados para proporcionar una representación sistemática de referencias a posiciones.

Para los anteriores procesamiento de la información geográfica se requiere de una descripción de la ubicación o posición. La ubicación puede ser especificada a través de una descripción espacial o coordenadas (X: 711124, Y: 740705). Las coordenadas son partes de un sistema de coordenadas: éste puede ser un sistema local con un origen arbitrario, o uno relacionado a un modelo matemático de la Tierra.

Para representar la Tierra en un plano, primeramente hay que proyectar la superficie terrestre sobre la superficie del elipsoide terrestre o esfera terrestre, lo cual se realiza en los procesos geodésicos y levantamientos topográficos; y posteriormente la representación de la superficie del elipsoide sobre un plano, a través de una proyección cartográfica.

Una proyección cartográfica es un método de representación, determinado matemáticamente, de la superficie del elipsoide o esfera terrestre sobre el plano, estableciendo una dependencia analítica entre las coordenadas geográficas (Latitud, Longitud) de los puntos del elipsoide terrestre y las coordenadas rectangulares (X, Y) de los mismos puntos en el plano como se muestra en la figura 3.

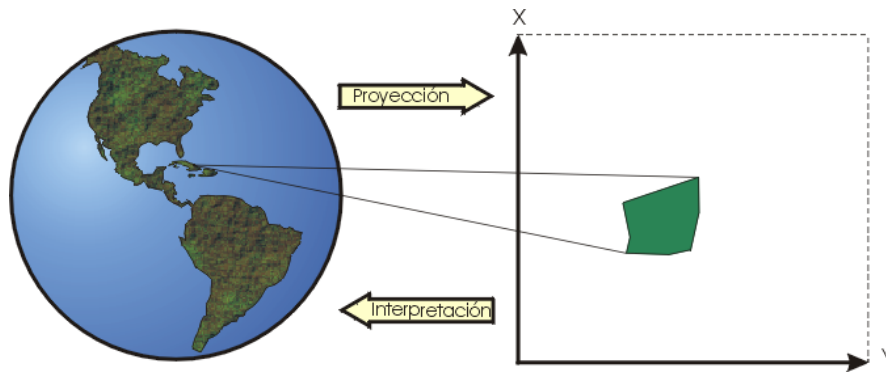


Figura 3. Sistema de Referencia Espacial.

Componente temática de un SIG Vectorial.

Los Sistemas de Información Geográfica vectoriales tienen un sistema de gestión de bases de datos de los atributos temáticos. Una cuestión importante para diseñar la base de datos es llevar a cabo un análisis previo de la información que se va a incluir, especialmente las relaciones entre los diversos elementos que la integran. Uno de los métodos más aceptados para el análisis de datos ha sido el llamado entidad-relación.

En un SIG el conjunto de entidades lo integran cualquier hecho que puede ser localizado espacialmente: carreteras, escuelas, hospitales, monumentos. Los atributos son las características asociadas a la entidad, por ejemplo, longitud de la carretera, nombre de la escuela, altura del monumento. Las relaciones son cualquier tipo de interacción que existe entre las entidades.

Operaciones de Extracción de Información.

Las operaciones de extracción de información de un SIG vectorial se utilizan para seleccionar objetos geográficos de la base de datos, tomando aquellos que cumplen con una condición establecida por el usuario. Los resultados se muestran en mapas y/o tablas de valores, que contienen todos los objetos geográficos con un conjunto particular de atributos, ya sean espaciales (localizaciones) o temáticos (valores de algunas variables).



Existen dos formas diferentes de operaciones de selección de la información: recuperación de tipo espacial: ¿qué existe, qué valor adopta un atributo temático en esta localización concreta?, y la búsqueda temática: ¿en qué lugar existe, qué localización tiene este atributo temático concreto?

La recuperación de información temática se puede realizar mediante una especificación nominal, o también utilizando una condición aritmética o lógica, referida a los atributos temáticos. La búsqueda espacial se puede realizar especificando un dominio espacial, o mediante una condición geométrica/topológica.

Representación y Entrada de la Información.

Anteriormente explicábamos que en un SIG los fenómenos del mundo real se modelan a través de las figuras, y éstas, a la vez, se representan en forma de geometrías. Existen muchos tipos de geometrías, como muestra la Figura 4, y cada desarrollador toma el conjunto que considera más representativo para modelar los fenómenos reales con la mayor exactitud posible. El conjunto total se clasifica en cuatro grupos: geometrías puntuales, lineales, areales y combinaciones de estos tres.

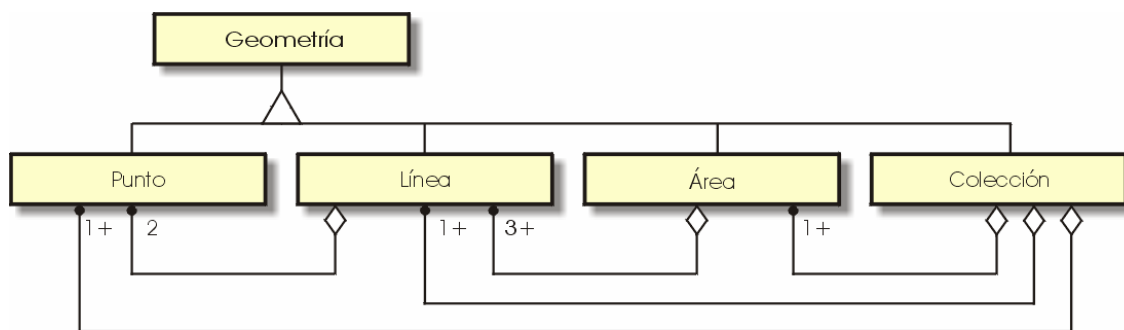


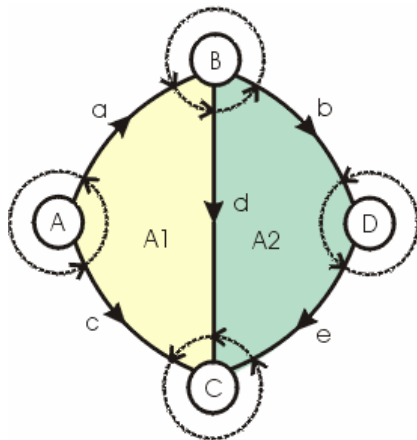
Figura 4. Geometrías.

Considerando la relación topológica entre las geometrías al definir las, un punto es un par coordinado, un segmento se forma entre dos puntos y un área está limitada por segmentos. La estructura de datos más conocida para representar topológicamente las geometrías es ARCO/NODO, para el cual las geometrías básicas reciben los



nombres de nodo, arista y área. Esta estructura de datos tiene la ventaja que puede ser almacenada fácilmente en una base de datos relacional.

Tabla 1. Estructura de datos ARCO/NODO.



ID	A	B	C	D
Coordenada	X1,Y1	X2,Y2	X3,Y3	X4,Y4
Incide	a	b	d	e

Nodo

ID	a	b	c	d	e
Nodo inicial	A	B	A	B	D
Próxima	c	a	a	b	b
Nodo final	B	D	C	C	C
Próxima	d	e	e	c	d
Área derecha	A1	A2	Null	A1	A2
Área izquierda	Null	Null	A1	A2	Null

Arco

La tabla 1 muestra cómo se representa la estructura de datos ARCO/NODO en dos tablas. Nodo tiene como atributos un identificador, el par coordenado de su ubicación y una arista cualquiera que parte o llega a él. La tabla arco tiene como atributos un identificador, los nodos extremos: inicial y final entre los que está comprendido el arco, la próxima arista que incide en cada nodo extremo, tomándola en sentido inverso a las manecillas del reloj y las áreas que se encuentran a ambos lados del arco, tomando como referencia de la posición la dirección del arco.

La introducción de nuevas geometrías al sistema puede hacerse de dos maneras: por la participación directa del usuario al entrar las coordenadas manualmente, o importando un archivo con información espacial. Esta operación puede resultar inexacta y traer redundancias e inconsistencias en la información topológica. La plataforma debe tener entonces la capacidad de depurar la información de forma interactiva y proporcionar herramientas de validación para identificar y arreglar problemas en la topología rápidamente, y antes de cualquier operación de análisis. Esta capacidad se conoce como limpieza de la topología y tiene los siguientes pasos, los cuales se representan en la Figura 5:

1. Segmentar las aristas que se cortan.

2. Eliminar las aristas duplicadas o semejantes a otras.
3. Eliminar los fragmentos lineales.
4. Eliminar aristas o segmentos de aristas que no aportan una información significativa.
5. Cerrar aberturas: dos aristas cuyos extremos deben ser el mismo nodo, o una arista cuyo extremo está casi sobre otra arista.

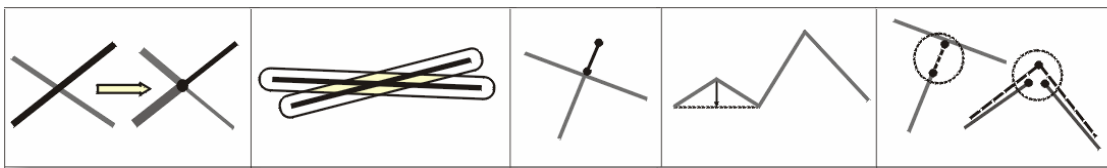


Figura 5. Los cinco pasos de limpieza de la topología.

Terminada la limpieza debe completarse la topología creando las áreas que se forman entre las aristas, para que exista un objeto geográfico que represente algún fenómeno real, y que sea útil en las operaciones de análisis. Por ejemplo, una manzana es, en realidad, el área interior a las aristas que la forman, y no las propias aristas. Esto es necesario por un problema práctico a la hora de enlazar los elementos de la base de datos temática y espacial.

El centroide del área constituye la forma más común de enlazar las áreas con la base de datos temática, aunque no constituye estrictamente una necesidad luego de crear el objeto areal. Sólo resta enlazar los centroides creados con el área correspondiente.

Creación y Clasificación de Grupos Informativos.

Un SIG que represente un área geográfica no muy extensa posee como promedio miles de elementos gráficos que deben ser clasificados y distribuidos de alguna forma para mayor eficiencia y comodidad del usuario. Tradicionalmente se ha usado



el concepto de nivel o capa para agrupar objetos geográficos de la misma naturaleza. Por ejemplo, nivel para redes hidrográficas, o para redes viales, nivel para manzanas, parcelas, etc.

Los niveles se disponen en orden, afectando las partes visibles de cada uno: sólo se ve completo el nivel por encima de todos, y parcialmente los inferiores en aquellos lugares que no han sido solapados por objetos de niveles superiores. A voluntad del usuario se puede ocultar (apagar) un nivel cuando no le interese los detalles que contiene, o desee ver completamente una capa solapada por otra.

Se puede representar una región geográfica en varias escalas donde los elementos gráficos pueden tener una gran variedad de tamaños. Consecuentemente, a escalas muy grandes desaparecen los elementos pequeños, o cambian su aspecto. Por ejemplo, en la escala 1:250 000 los edificios de importancia se representan como puntos en el mapa, sin embargo en la escala 1:10 000, se representan las parcelas que forman. Esta dificultad se soluciona asignando al nivel un rango de visibilidad según la escala, y representando los objetos geográficos en niveles diferentes, de rangos complementarios, con figuras correspondientes a la escala. En la Figura 6 se representa un edificio en tres escalas distintas: para 1: 250 000 es visible el nivel A donde el edificio se representa como un punto; en la escala 1: 100 000 continúa visible la misma capa pero con un dibujado diferente. Ya en la escala 1: 10 000 el nivel A no es visible, y aparece el nivel B donde se representa el mismo edificio, pero como una parcela. Note que a pesar de cambiar los niveles al representar el mismo objeto, las tablas con los atributos temáticos del objeto no cambian.

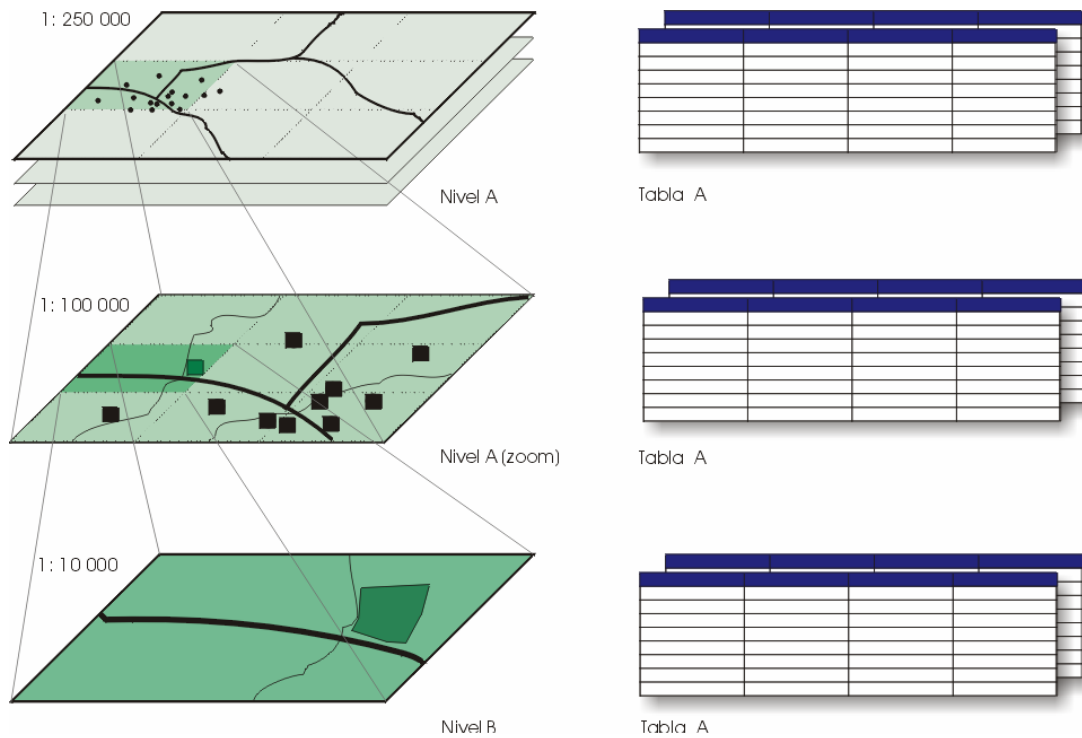


Figura 6. Efecto del cambio de escala en la representación de las geometrías.

La categoría es una concepción diferente de agrupamiento y superior a la noción de nivel. Consiste en reunir en una capa figuras tomadas de niveles u otras categorías determinadas según un conjunto de atributos espaciales y temáticos especificado, y que reúne objetos geográficos pertenecientes a un mismo concepto o clase. Por ejemplo, tomando los objetos del tipo hospital, policlínico, hogar materno de la capa edificaciones, podemos hacer una generalización creando la categoría centros de salud. Tomando de los objetos del tipo edificios aquellos que están en construcción, podemos hacer una particularización creando la categoría edificios incompletos.

Gestión de la Información Espacial.

La gran cantidad de información que maneja un SIG conduce a la creación de una voluminosa base de datos, inclusive si se almacenara solamente información espacial. Un manejo deficiente de ella podría traer ineficiencias en el tiempo general de ejecución, siendo muy tedioso para el usuario; sin contar que en general la



cantidad de memoria operativa de una computadora está muy limitada. El objetivo de un buen diseño es ofrecer un acceso rápido y eficiente independientemente del volumen de la información que exista.

Dividir en partes equivalentes el volumen resulta un método sencillo, de poco costo y a la vez se obtienen muy buenos resultados. La principal incógnita para llevarlo a cabo consiste en cuál criterio escoger para la fragmentación, que cumpla con nuestro propósito y sin que pierda integridad la base de datos.

Algunas plataformas de SIG dividen el mapa en rectángulos iguales, semejante a un tablero de ajedrez. Cada rectángulo, llamado hoja, contiene toda la información de las geometrías en sus respectivos niveles, como si constituyera por sí mismo la base de datos espacial. Para localizar en cuál hoja se almacena un conjunto de elementos gráficos específicos, se crea una tabla de índices con la descripción geométrica de cada rectángulo y la localización y nombre del fragmento de la base de datos correspondiente.

Pero como las geometrías no se distribuyen regularmente, en el plano existirán hojas muy cargadas, o un conjunto de hojas casi vacías y por tanto innecesarias. Pero lo más desventajoso es que los límites de las hojas segmentan los objetos gráficos, alterando la topología y fragmentando en partes un objeto, distribuyéndolas en bases de datos diferentes, lo que provoca mayores dificultades en el mantenimiento y búsqueda de la información.

Aprovechando los elementos lineales que ya existen, se puede dividir el mapa en hojas, Figura 7, con cualquier forma y tamaño, controlar la carga de cada una, y sin segmentar las geometrías. Por ejemplo, para formar los límites de las hojas se pueden aprovechar los límites intermunicipales. Este método exige que la topología sea válida, por tanto, para que la división sea correcta, antes se debe hacer la limpieza de la topología.

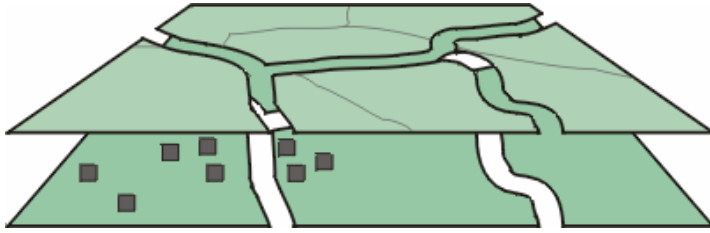


Figura 7. División del mapa en hojas

Este método repite elementos (los límites) entre las hojas, pero la cantidad de información redundante es despreciable. También es algo más complejo a la hora de determinar dónde se encuentra una geometría, pero es determinable a través de los operadores espaciales usando el mismo método de la tabla de índices.

Búsqueda espacial rápida.

A pesar de dividir el volumen total de la base de datos, es conveniente implementar algún mecanismo de búsqueda rápida sobre cada hoja para mayor eficiencia con los operadores espaciales. La búsqueda espacial rápida consiste en un algoritmo apoyado en una estructura de datos específica que acelera el proceso de localización de un objeto espacial dentro de una colección grande de ellos.

Este método, también llamado búsqueda geométrica, debe evaluarse siguiendo los siguientes criterios: cuánto tiempo se requiere, el caso promedio como el peor de los casos para responder a una búsqueda simple; la cantidad de memoria para almacenar la estructura de datos; el tiempo para preparar y llenar y dejar lista la estructura de datos, y el tiempo para añadir o quitar un elemento en la estructura de datos.

Lograr el éxito en la búsqueda, depende de la habilidad de reducir rápidamente el conjunto de elementos. Los mejores métodos que se conocen se basan en la división del plano (semejante a la división que se hace en hojas para minimizar el conjunto de búsqueda), y la búsqueda binaria. Los métodos hash y quadtree constituyen un ejemplo de la primera clase, y los métodos del polígono y árboles binarios multidimensionales del segundo.



Consultas Espaciales y Temáticas.

El análisis espacial agrupa un conjunto de operaciones orientadas al estudio de las características espaciales de los datos geográficos. La mayor parte de los análisis espaciales, se desarrollan teniendo en cuenta sólo las características geométricas del fenómeno geográfico, sin considerar los atributos temáticos asociados a ellos. Entre las operaciones más usadas están:

- Mediciones: distancia entre dos geometrías, longitud de una secuencia de líneas, calcular el área y perímetro de un polígono.
- Operaciones de cercado: dada una geometría de selección, obtener las que se encuentran totalmente dentro, las que están dentro y la solapan, las que están dentro y cortar las que la solapan, las que están fuera.
- Operadores de redes: determinar entre dos puntos: si existe un camino, el camino óptimo, los posibles caminos.
- Operadores generales: dada una geometría de selección, obtener las que la tocan, cruzan, la contienen, la zona de buffer, la envoltura convexa. También determinar dadas dos geometrías la unión, intersección, si se cortan o cruzan.

La mayoría de los SIG almacenan los datos temáticos en bases de datos relacionales, y adoptan como lenguaje de consulta el SQL. A través de éste se formulan preguntas (condición aritmética o lógica) referidas a los atributos temáticos para obtener en forma de tablas la información de los objetos geográficos que cumplan con una condición dada.

Una herramienta de análisis muy eficaz se logra mediante la combinación del análisis espacial y el temático. A los operadores brindados por el SQL se le suman los operadores espaciales, y de esta forma se pueden formular preguntas tales como:



- Desde una estación de bomberos determinar la vía más rápida de llegar al incendio teniendo en cuenta las pautas de circulación, el tráfico, los posibles obstáculos.
- Determinar en cuáles municipios del país la línea férrea central se cruza con la carretera central o la autopista nacional.

2.2.3 Plataformas de Desarrollo de SIG.

Las plataformas para el desarrollo de un SIG son aquellas aplicaciones con las siguientes capacidades:

- Entrada de información: son los procedimientos que permiten convertir la información geográfica del formato analógico, el habitual en el mundo real, al formato digital que puede manejar el ordenador. Aquí se incluyen los procedimientos que permiten eliminar errores o redundancias en la información incorporada a la base de datos espacial.
- Representación gráfica y cartográfica de la información: se refiere a las actividades que sirven para mostrar al usuario los datos incorporados en la base de datos del SIG, y los resultados de las operaciones analíticas realizados sobre ellos. Permiten obtener mapas, gráficos, tablas numéricas y otro tipo de resultados en diferentes soportes: pantallas gráficas, papel u otros.
- Gestión de la información espacial: se extraen de la base de datos las porciones que interesan en cada momento, y es posible reorganizar todos los elementos integrados en ella de diversas maneras.
- Las funciones analíticas: o como las llamamos anteriormente, las operaciones de extracción de la información. Son los elementos más característicos de un Sistema de Información Geográfica. Facilitan el procesamiento de los datos integrados en él de modo que sea posible obtener mayor información, y con ella mayor conocimiento del que inicialmente se disponía.



Un error común consiste en confundir el concepto de SIG con las plataformas de montaje de SIG. Sobre una plataforma se desarrolla un SIG, y brinda el conjunto de operaciones necesarias para hacerlo funcionar; pero conceptualmente no lo es porque no posee una base de datos espacial y temática, que es una parte esencial del mismo. Aunque sea común que la plataforma se convierta en un SIG al incluirle dicho componente con las herramientas específicas para el tipo y estructura de la información que maneje.

2.2.4 Información Geográfica Distribuida.

Se denomina Información Geográfica Distribuida (IGD) a los productos o servicios que se relacionan con el empleo de las tecnologías de Internet, las cuales permiten a los usuarios tener acceso a la información geográfica distribuida en una gran variedad de formas como lo pueden ser mapas, imágenes, conjuntos de datos, análisis de información y reportes [23].

Las aplicaciones para la IGD van desde una simple imagen de un mapa en una página de HTML hasta el empleo de un SIG en el cual los usuarios trabajan de manera remota con archivos comunes.

Actualmente la mayoría de las aplicaciones que existen para el manejo de IGD se especializan solamente en un formato y un tipo específico de datos, lo cual las hace aplicaciones para datos específicos y nos limita para consultar datos de interés común que también estén disponibles.

Los servicios de información geográfica con que cuenta el Grupo Empresarial GEOCUBA, es la base para un nuevo modelo de empleo de la Información Geográfica Distribuida. La obtención de información a partir de una consulta realizada sobre un mapa en un entorno Internet y la utilización de formatos estándares basados en XML para el intercambio de esta información, tanto temática como espacial, hacen de estos servicio los idóneos para soportar sistemas informativos distribuidos. En estos nuevos sistemas informativos la información



geográfica será una mas a consultar por medio de las interfaces que expone el servidor de mapas.

Para la completa interoperabilidad de los datos se obtuvo una adecuada infraestructura de datos espaciales y de metadatos donde mediante normas y especificaciones se establezcan los procedimientos para permitir la coexistencia varios proveedores de metadatos para una misma entidad. Por ejemplo: un consultorio médico de la familia tiene un conjunto de información geográfica que la definen como un objeto de ese tipo y permiten ubicarlo espacialmente. Además posee un conjunto de datos pertenecientes al Ministerio de Salud Pública que lo definen como una instalación de Salud con un médico, una enfermera, 2 camas y otras informaciones. Por otra parte otras instancias, como el organismo nacional que controla las instalaciones del gobierno, también poseen datos referentes a ese consultorio. Para la correcta utilización de estos servicios en un entorno de sistema informativo distribuido, se estandarizaron de los identificadores a utilizar en cada capa informativa. De esta forma se crearon una gran cantidad de mapas informativos diferentes, a partir de clientes desarrollados por los proveedores de metadatos para una entidad determinada.

2.3 Visualización Científica y Cartografía en la Web.

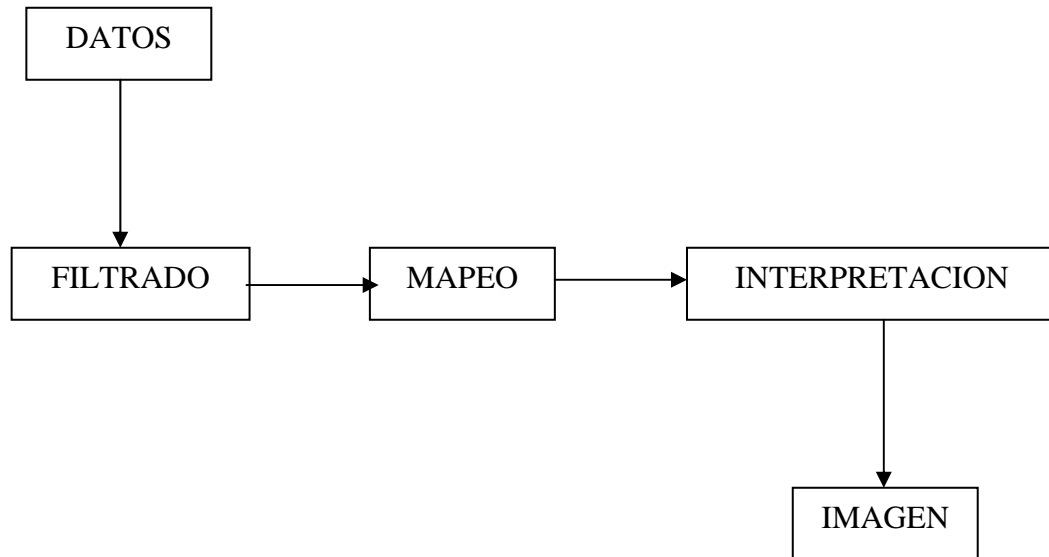
La Visualización Científica [24] significa encontrar una apropiada representación visual para un conjunto de datos dado, con el objetivo de permitir un análisis y evaluación más eficaz de los datos; lo cual simplifica el análisis, el entendimiento y la comunicación de los modelos, conceptos y los datos; con la perspectiva de facilitar a los usuarios no solo el ver sino también reconocer, entender y evaluar los datos.

Esto significa que la Visualización Científica provee una representación visual apropiada para los usuarios para mostrar las relaciones internas entre los datos, la cual se pierde de otra manera y soporta el intercambio de los resultados.



La visualización de un conjunto de datos debe ser expresiva, lo cual significa que la representación refleja los datos; eficaz, si la representación visual permite una percepción espontánea; y apropiada, si tiene un adecuado balance costo beneficio.

El proceso de convertir datos en imágenes consta de tres pasos, como se observa en la gráfica siguiente:



1. Filtrado: Extracción de datos o de propiedades de los datos y reducción o completamiento de los datos.
2. Maqueo: Conversión de los datos en gráficos primitivos 2D o 3D. Ejemplo, creando una lista de polígonos.
3. Interpretación: Generación de imágenes o secuencias de imágenes.

La aplicación obtenida producto de esta investigación, además de estar soportada en los estándares internacionales que existen para ello, implementa los tres pasos anteriormente explicados, lo que garantiza un enfoque científico adecuado al proceso de visualización de la información geográfica que soporta la misma.

La información geográfica devuelta como mapas con significación visual es lo que hace "vivos" a los datos de cara a los usuarios en la Web. La Tabla 2 siguiente lista



los estándares que se aplican para la interoperabilidad de la cartografía en Web ("maps") y la visualización.

Tabla 2. Estándares para la interoperabilidad de la cartografía en Web y la visualización.

	Petición de Servicios	Transferencia de información
Especificaciones de Implementación	Interfaz OGC Web Map Service I.S. ISO 19128 (Web Map Service)	Codificación GeoTIFF, SVG, PNG, JPEG OGC Styled Layer Descriptor
Especificaciones de Implementación	Interfaz OGC Web Map Service I.S. ISO 19128 (Web Map Service)	Codificación GeoTIFF, SVG, PNG, JPEG OGC Styled Layer Descriptor
Modelos Abstractos	Comportamiento OGC Doc. 98-060 (User interaction with geospatial data) OGC Doc. 98-061 (Essential model of interactive portrayal)	Contenido ISO CD 19117, Portrayal

La especificación de implementación de Servicio de Mapas en Web OGC es la principal especificación para la petición de mapas y su visualización; detalla un interfaz simple para la petición de mapas mediante la World Wide Web. Sus peticiones "GetMap" son precedidas por la petición "GetCapabilities" para verificar la disponibilidad de capas ("layers") de información cartografiada en el servidor, y sus habilidades de proceso y traducción.



Aunque los mapas son imágenes mucho más complejas que los datos, no requieren esquemas especiales de codificación; en su lugar se emplean formatos "ráster" convencionales como (PNG) Formato Gráfico Portable, (JPEG/JFIF) Formato del Conjunto de Expertos Fotográficos, (TIFF) Formato de Archivo de Imagen Etiquetado, o GeoTIFF, o formatos de vector como (SVG) Formato Gráfico Escalable Vectorial.

El papel de discusión informal de OGC Descriptor de Estilo de Capa diseña reglas de representación de expresiones en XML que dicen a un Servidor de Mapas OGC cómo devolver sus capas o las de un Servidor Web de Características OGC.

Los modelos abstractos, a pesar del amplio uso de la cartografía en Web y la visualización, ha sido solamente limitado a una definición formal:

- El borrador estándar de ISO Representación ("portrayal") (ISO 19117 / DIS: 2001-06; FDIS: 2002-01; IS: 2002-03) define reglas para la representación de características ("features") geoespaciales.
- Dos documentos internos de trabajo de OGC sobre Interacción del usuario con los datos geoespaciales (Doc. 98-060) y representación interactiva (Doc. 98-061) diseñan los servicios genéricos requeridos. La especificación de implementación Servicio de Mapas en Web de OGC, señalado arriba, resume los principales conceptos en estos documentos.



Un Servidor de Objetos Geográficos puede realizar consultas a una base de datos OpenGIS que contiene información geográfica. Las consultas son de carácter descriptivo y espacial en GML, las cuales pueden construidas utilizando filtros que son estructura construida para restringir las propiedades de un tipo de objeto, con el propósito de identificar un subconjunto de instancias de objeto para ser manejados de alguna la manera; finalmente el WFS da la solución a estas consultas también en GML, que es un documento estándar de almacenamiento y transporte de información geográfica.

3.1 Accesos a las Bases de Datos.

GEOCUBA ha venido desarrollado varios proyectos con el fin de implementar las Interfaces de OpenGIS que definen el tratamiento de la geometría, los sistemas de referencia espacial, la manipulación de objetos geográficos y el servicio de mapas en Web.

Se implementó un Modelo de Base de Información Geográfica (BIG) que presenta una estructura jerárquica basada en catálogos donde se encuentran agrupados los diferentes elementos que conforman la misma. Las premisas del modelo implementado se fundamentan en la disposición de un conjunto predefinido por el sistema de clases de objetos y los métodos para crear dinámicamente otras nuevas por parte del usuario, con las cuales se puede conformar una Base de Información Geográfica personalizada. La manipulación y el almacenamiento de los objetos geográficos cumple con las especificaciones: *Feature Geometry*, *Simple Feature* y *Spatial Reference System*, publicadas por OGC para este fin.

La Base de Información Geográfica creada utiliza el Modelo de Objetos Componentes y será el punto de convergencia de futuras tecnologías que utilicen la información geográfica.

A partir del estudio de estas especificaciones de OGC para el servicio de mapas en Internet se implementó un Servidor de Mapas y varias aplicaciones clientes que permiten visualizar y consultar la información geográfica de este servidor.



La ejecución de estos proyectos, teniendo en cuenta los estándares internacionales proporcionados por OpenGIS, nos ha permitido lograr estrategias acorde al desarrollo global de las tecnologías de la Información. Se han decantado un conjunto de metodologías y tecnologías de desarrollo de software que giran todas ellas alrededor de un viejo concepto, el componente. De forma general todas ellas se nutren del modelo orientado a objetos, agregándoles una capa adicional de aislamiento, llamada interfaz, con el cliente. Esta concepción permite centrar el problema del desarrollo de una solución en identificar los componentes que actuarán en la misma y sus relaciones contractuales.

Base de Información Geográfica (BIG).

Como resultado de los trabajos realizados se obtuvo una plataforma que permite implementar Sistemas de Información Geográfica de una forma novedosa y que cumple con las normalizaciones internacionales dictadas por el consorcio OpenGIS, las bases de dicha plataforma descansan en la Base de Información Geográfica.

La BIG es la implementación en forma de tablas relacionadas de una base de datos orientada a objetos. Por su diseño jerárquico en forma de árbol, permite tanto la modelación de estructuras sencillas de organización de la información, como un diseño de una jerarquía corporativa altamente compleja. Dentro de una BIG se almacenan, por niveles jerárquicos, los distintos tipos de catálogos, carpetas, colecciones de objetos y otras entidades que la componen.

A través de una interfaz adecuada que brinda una aplicación que se implementó al efecto, el usuario puede visualizar la jerarquía de las entidades de la BIG en forma de árbol, además de disponer de las herramientas necesarias para navegar dentro de este, es decir manipular los objetos geográficos, visualizarlos, realizar consultas, tanto espaciales como temáticas, sobre ellos y las funciones necesarias para lograr un fin informativo determinado.

En el diseño de esta BIG se tienen en cuenta algunas de las especificaciones del consorcio OpenGIS. La geometría se almacena en un campo binario de las tablas junto a las demás propiedades de los elementos geográficos como lo indica OpenGIS. Esta geometría se almacena en el formato especificado por dicho



consorcio y para su manejo se implementó la especificación “*Features Geometries*” que define las clases de geometrías necesarias para la implementación de un Sistema de Información Geográfica, así como las operaciones espaciales y de relaciones que se establecen entre ellas.

Además, fue necesario implementar las clases y funciones que permiten el manejo de diferentes Sistemas de Referencia Espacial, lo que le proporcionará al sistema interoperar con otros sistemas en la manipulación de los datos espaciales permitiendo realizar operaciones con datos que procedan de diferentes sistemas de referencia.

Trabajo con las Geometrías.

Geometría de Objetos Geográficos constituye una implementación del modelo conceptual definido en las especificaciones 98-101 “*Feature Geometry*” y “*OpenGIS Simple Feature Specification por OLE/COM*” del Consorcio OpenGIS, en una biblioteca de enlace dinámico denominada *FeatureGeometries.dll* mediante la cual se pueden instanciar los objetos COM relacionados con la representación gráfica y el procesamiento de la Información Geoespacial.

El Modelo de Objetos Geométricos utilizado es propuesto por el Consorcio OpenGIS. En este modelo la clase **Geometría** tiene como subclases a las clases **Punto**, **Curva**, **Superficie** y **Colección Geométrica**. Asocia a cada objeto geométrico un Sistema de Referencia Espacial, que describe la coordenada espacial donde se define el objeto geométrico. Existen otras Clases nombradas **MultiPunto**, **MultiPolilínea** y **MultiPolígono** que corresponden a colecciones de Puntos, Polilíneas y Polígonos respectivamente, la **MultiCurva** y la **MultiSuperficie** son introducidas como superclases abstractas, las cuales generalizan la colección que contienen **Curvas** y **Superficies**.

Además se incluyeron en la implementación de estas clases dos interfaces, una de ellas define las relaciones espaciales entre estas geometrías y la otra las operaciones espaciales que admitirán las mismas.

Las funciones de Relación Espacial implementadas son:



- *Equals*: Determina si dos geometrías son iguales.
- *Touches*: Determina si dos geometrías se tocan.
- *Contains*: Determina si la geometría en análisis contiene a otra.
- *Within*: Determina si la geometría en análisis está dentro de otra.
- *Disjoint*: Determina si las geometrías son disjuntas.
- *Crosses*: Determina si dos geometrías se cruzan.
- *Overlaps*: Determina si dos geometrías se solapan.
- *Intersects*: Determina si dos geometrías se interceptan.

Las funciones de Operaciones Espaciales implementadas son:

- *Distance*: Retorna la distancia entre las geometrías.
- *Boundary*: Retorna la frontera de la geometría.
- *Intersection*: Calcula la intersección de dos geometrías.
- *Union*: Calcula la unión de dos geometrías.
- *Difference*: Calcula la diferencia de dos geometrías.
- *SymmetricDifference*: Calcula la diferencia simétrica de dos geometrías.
- *Buffer*: Retorna un polígono que incluye todos los puntos a una distancia de la frontera de la geometría fuente
- *ConvexHull*: Retorna la envoltura convexa de la geometría fuente

Sistemas de Referencia Espacial.

Sistemas de Referencia Espacial constituye una implementación del modelo conceptual definido en las especificaciones 98-102 "Spatial Reference Systems" y "OpenGIS Simple Feature Specification por OLE/COM" del Consorcio OpenGIS, en



una biblioteca de enlace dinámico denominada **VAGOsrs.dll** mediante la cual se pueden instanciar los objetos COM para el manejo de los Sistemas de Referencia Espacial.

Las clases implementadas son:

AngularUnit	Clase para representar una unidad angular en radianes.
LinearUnit	Clase para representar una unidad lineal en metros.
Ellipsoid	Clase para representar el elipsoide de revolución que se utiliza como una abstracción de la superficie de la tierra.
HorizontalDatum	Clase para representar un Horizontal Datum que representa la abstracción de la superficie terrestre para una determinada región.
PrimeMeridian	Clase para representar un meridiano primo en un Sistema de Referencia Espacial (SRE). Un meridiano primo es el meridiano a partir del cual se miden los valores de las latitudes en el SRE y este es relativo al Meridiano de Greenwich.
GeographicCS	<p>Clase que representa un Sistema de Coordenadas Geográfico. Un Sistema de Coordenadas Geográfico (SCG) es un sistema donde las coordenadas se miden en unidades angulares, longitud y latitud.</p> <p>Un SCG esta formado por un datum que representa la abstracción de la superficie terrestre y un meridiano primo que se utiliza para medir las latitudes relativas al meridiano de Greenwich.</p>
<u>Parameter</u>	<p>Clase que representa uno de los parámetros utilizados en los Sistemas de Referencia Espacial.</p> <p>Los parámetros son utilizados en la Proyecciones y los</p>



	Sistemas de Coordenadas Proyectados.
<u>ProjectedCS</u>	<p>Clase que representa un Sistema de Coordenadas Proyectado (SCP). Un SCP es un sistema de coordenadas plano, generalmente rectangular, que es la representación en un plano (x,y) de toda o una porción de la superficie de la tierra.</p> <p>Un SCP se basa en un Sistema de Coordenadas Geográfico y una Proyección que transforma la superficie de la tierra a un plano.</p>
<u>SRFactory</u>	Clase para crear las distintas entidades utilizadas en los Sistemas de Referencia Espacial.

Geometría

La geometría es la combinación de una geometría de coordenadas y un sistema de referencia espacial. La geometría de coordenadas puede ser de una de estas 4 formas:

1. Una sucesión de puntos coordinados, todos del mismo sistema de la referencia espacial.
2. Una colección de otras geometrías, todas del mismo sistema de la referencia espacial.
3. Un algoritmo de interpretación que utiliza estas geometrías y puntos coordinados para construir otra geometría.
4. Un Sistema de la Referencia Espacial temporal que interpola entre la geometría de las coordenadas y su ubicación, dando una interpretación del mundo real.

Las instancias de una geometría simple están asociadas con varios puntos coordinados que proyectan “el mundo real” a través del uso de un sistema de referencia espacial temporal, constante para esa geometría particular.



Pueden construirse geometrías más complejas uniendo algunas geometrías simples. Esta clase particular de geometría tendrá un esquema de interpretación asociado que le permitirá definir el límite espacial de la geometría para su sistema de referencia espacial. Debe señalarse que la profundidad de la definición recursiva de una geometría puede ser tan grande como sea necesario, pero el sistema de referencia espacial temporal es constante a lo largo de la construcción. Esto se indica en el diagrama por la relación del Sistema de Referencia Espacial temporal con la geometría tipo punto, en este caso la relación con el punto es un subconjunto de la relación con la geometría. Por ejemplo, cada anillo de un polígono podría representarse como una línea simple, cerrada (una sucesión de puntos con interpolación lineal). El polígono puede representarse entonces como la colección de anillos con la interpretación usual.

Los objetos del tipo geometría deben poder descargar su definición en formatos Well Known Struct para lograr la comunicación entre objetos para manipularlos en mensajes necesarios en la implementación del modelo.

Breve Descripción de las clases.

Geometría.

La Figura 8 presenta el Diagrama de Clases del paquete Geometría definido por OGC y donde la clase geometría es la clase de mayor jerarquía, y no se pueden crear instancias de la misma, ya que es una clase abstracta.

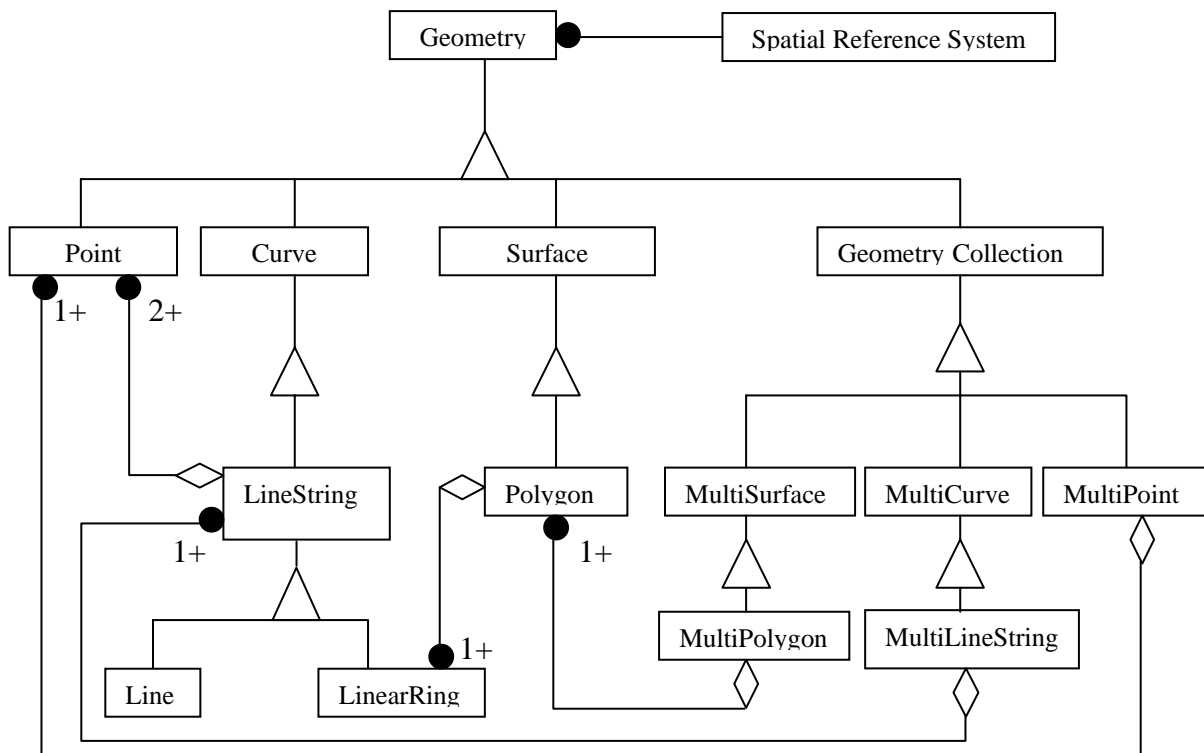


Figura 8: Diagrama de Clases del paquete Geometría

Las subclases instanciables de Geometría se restringen a objetos geométricos de 0, 1 y 2 dimensiones, los cuales son representados en coordenadas espaciales de dos dimensiones.

Métodos de la clase Geometría.

Todos los objetos geométricos implementan 4 interfaces. La interfaz principal es *IGeometry*, esta identifica el comportamiento lógico que deben tener todos los objetos geométricos. Mediante una segunda interfaz llamada *ISpatialOperator* se realizan las operaciones espaciales con otras geometrías, se verifican relaciones espaciales entre las mismas utilizando la interfase *ISpatialRelation* y la interoperabilidad se logra mediante la interfase *IWkt* que tiene como nombre las siglas de *Well Known Text*.

Métodos de *IGeometry*:

Clone Crea y retorna una nueva geometría idéntica a la geometría



fuente.

Dimension	Retorna la dimensión inherente de la geometría.
Envelope	Retorna el rectángulo que delimita el límite geográfico (<i>Envelope</i>) de la geometría.
IsA	Retorna el tipo de objeto de la geometría.
IsEmpty	Determina si la geometría es vacía.
IsSimple	Determina si la geometría es simple.
Project	Retorna una nueva geometría como resultado de la proyección aplicada.
SpatialReference	Retorna el Sistema de Referencia Espacial.

Los métodos de las interfases *ISpatialRelation* e *ISpatialOperator* dan respuesta a todas las funciones descritas en el epígrafe de trabajo con geometrías para Relaciones Espaciales y Operaciones Espaciales.

Métodos de IWkt:

ImportFromWKT Crea una geometría a partir de su representación en el formato *Well-known Text*

ExportToWKT Retorna la representación en formato *Well-know Text* del objeto.

Punto

Un punto es una geometría de dimensión 0 y es representada por una localización simple en coordenadas de espacio. Un Punto tiene un valor para la coordenada "x" y un valor para la coordenada "y".



MultiPunto

Un multipunto es una colección geométrica. Los elementos de un multipunto son restringidos a Puntos, los cuales no están conectados ni ordenados. Un multipunto es simple si no existen Puntos con valores idénticos, es decir no existen Puntos iguales. Un ejemplo de MultiPunto se muestra en la Figura 9. El límite (boundary) de un MultiPunto es el conjunto vacío.

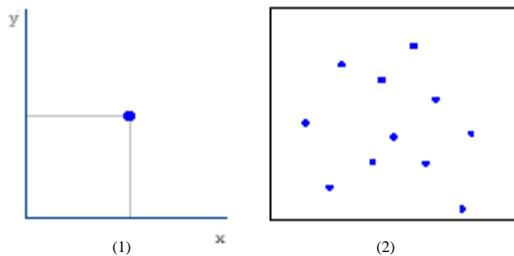


Figura 9. Ejemplo de Puntos (1) y Multipuntos (2)

Curva

Una Curva es un objeto geométrico de una dimensión, usualmente es una sucesión de puntos, con el subtipo curva se especifica la interpolación entre puntos.

Una Curva es simple si no pasa por el mismo punto dos veces. El ejemplo 1 de la Figura 10 muestra una curva simple. El ejemplo 2 representa una curva no simple. Se dice que es una curva cerrada, si el punto inicial es igual al punto final, ver ejemplo 3 de la misma figura. El ejemplo 4 muestra una curva cerrada no simple.

El límite de una curva cerrada es el conjunto vacío. Una Curva que es simple y cerrada es un Anillo (*Ring*). El límite de una Curva abierta consiste en dos puntos, el inicial y el final.

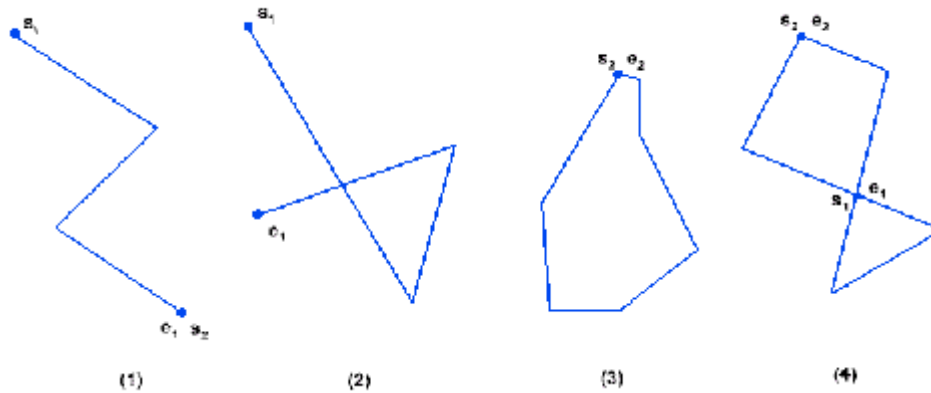


Figura 10. Ejemplo de Curvas. Curva simple (1), curva no simple (2), curva cerrada simple (3) y curva cerrada no simple (4).

Polilínea

Una Polilínea es una subclase de Curva, la cual es una interpolación lineal entre puntos. Cada par consecutivo de puntos define un segmento de Línea. Un Anillo es una Polilínea cerrada y simple. Una línea es una Polilínea de 2 puntos exactamente.

MultiCurva

MultiCurva, una clase no instanciable, es una Colección Geométrica de dimensión uno y sus elementos son Curvas. Es simple si todos sus elementos son Simples y solo puede haber intersección entre los límites de sus elementos.

La MultiCurva es cerrada si todos sus elementos son cerrados. El límite de una MultiCurva Cerrada es siempre el conjunto vacío.

MultiPolilínea

Una MultiPolilínea es una MultiCurva y sus elementos son Polilíneas. La Figura 11 muestra 3 ejemplos de MultiPolilínea. La primera esta formada por dos Polilíneas que se tocan en uno de sus puntos iniciales. La segunda esta formada por una Polilínea no simple y una cerrada. La tercera MultiPolilínea, está formada por dos Polilíneas cerradas.

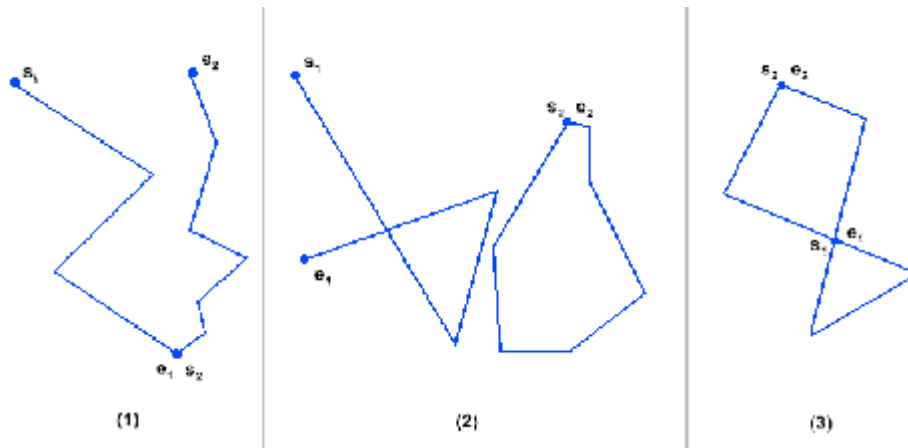


Figura 11: Ejemplo de MultiPolilínea. Dos Polilíneas que se tocan en uno de sus puntos iniciales (1), una Polilínea no simple y una cerrada (2) y dos Polilíneas cerradas (3).

Superficie

OpenGIS define a una Superficie Simple como un límite exterior con 0 o más límites interiores. El límite de una Superficie simple es el conjunto de curvas cerradas correspondientes a los límites exterior e interior.

Polígono

Un Polígono es una Superficie Simple Plana. Una Superficie es un objeto geométrico de dos dimensiones.

Un Polígono es topológicamente cerrado. El límite de un Polígono consiste de un conjunto de Líneas que forman el interior y el exterior. Un Polígono no puede tener líneas que cortan, púas o perforaciones.

El interior de un Polígono está formado por un conjunto de puntos conectados. El exterior de un Polígono es el conjunto de puntos no conectados.

Un Polígono es definido por un límite exterior y cero o más límites interiores. La Figura 12 muestra tres ejemplos de Polígonos, formados por un límite exterior.



Figura 12. Ejemplo de Polígonos formados por un límite exterior y ningún límite interior.

MultiSuperficie

Una MultiSuperficie es una Colección Geométrica de dimensión dos cuyos elementos son Superficies.

MultiSuperficie no es una clase instanciable. La subclase instanciable es MultiPolígono la cual corresponde a una Colección de Polígonos.

MultiPolígono

MultiPolígono es una MultiSuperficie cuyos elementos son Polígonos; en la Figura 13 se incluyen algunos ejemplos. En un MultiPolígono los interiores de dos polígonos no pueden intersectarse. Los límites de dos Polígonos no pueden cruzarse, sin embargo si pueden tocarse por un número finito de puntos. Un MultiPolígono es topológicamente cerrado y no puede tener líneas que lo corten, púas ni perforaciones.

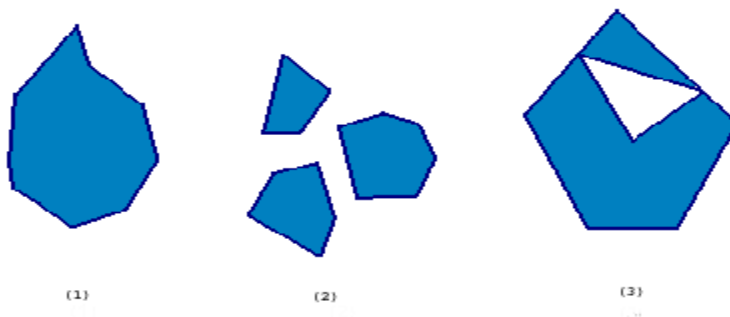


Figura 13. Ejemplos de MultiPolígonos. Un único polígono (1), conjunto de polígonos que no se intersectan (2) y polígonos que se tocan (3).



El límite de un MultiPolígono es el conjunto de curvas cerradas (Polilíneas) correspondientes a los límites de los Polígonos. Cada una de estas Curvas corresponde al límite de un Polígono exactamente.

3.2 Implementación de la especificaciones de implementación GML versión 2.

GML es la especificación de interfase de OpenGIS liberada en enero 2003, que describe la realidad en términos de objetos geográficos llamados *features*.

El "*Geographic Markup Language*" (GML) es una codificación de XML para el transporte y almacenamiento de la información geográfica. Incluyendo las propiedades espaciales y no espaciales de un objeto (*feature*) geográfico.

GML esta diseñado para soportar interoperabilidad y lo hace a través de proveer: etiquetas (*tags*) geométricas básicas (todos los sistemas que soportan GML usan los mismos "*tags*" geométricos), un modelo de datos común (*features/properties*), y un mecanismo para crear y compartir la aplicación de esquemas (*schemas*).

3.2.1 Esquemas para datos Geoespaciales.

En términos generales un esquema define las características de una clase de objetos, en XML un esquema también describe como son marcados los datos.

GML v.2 especifica tres esquemas base para codificar la información espacial, vease el Anexo C. El esquema Geometría (*geometry.xsd*) que reemplaza el que apareció en GML v.1. Proporciona un esquema Objetos (*feature.xsd*) mejorado que soporta colecciones de objetos (*feature collections*), tipos de objetos (*features types*) e incluye propiedades comunes para los objetos tal como *fid*, *name* y *description* (un identificador de objetos, nombre y descripción). El esquema XLink (*xlincs.xsd*) que proporciona atributos para soportar las funcionalidades de enlace.

El esquema XML proporciona un conjunto de tipos de datos primitivos (cadena, booleanos, flotante,etc), y permite la creación de tipos de datos incorporados (*built-in datatype*) y definidos por el usuario (*user-defined datatype*).



3.2.2 Interpretación de los datos Geoespaciales.

GML se ha diseñado para mantener el principio de separar el contenido de la presentación. GML proporciona los mecanismos para la codificación de los datos geográficos de un objeto sin consideración alguna hacia cómo los datos se pueden presentar a un lector humano. Puesto que GML es una extensión de XML, puede ser estilizado fácilmente en una variedad de formatos para su presentación. Dentro de estos formatos están: gráficos vectoriales (vector graphics), gráficos de trama (raster graphics), texto, sonido y voz.

La generación de la salida gráfica en forma de mapas es una de las presentaciones más comunes de GML y se puede lograr en una variedad de maneras, incluyendo la representación directa por los applet gráficos, o estilizar los gráficos en una tecnología XML (SVG o X3D). Es oportuno mencionar que GML no es dependiente de ninguna especificación gráfica particular de XML.

Es importante establecer algunas distinciones entre datos geográficos (que es lo que se codifica utilizando GML) e interpretaciones gráficas de esos datos (que es lo que aparece en un mapa o otra forma de visualización). Los datos geográficos son una representación del mundo en términos espaciales de modo que sea independiente de cualquier modo de visualización de datos. Cuando hablamos de datos geográficos tratamos de capturar información sobre las propiedades y geometrías de los objetos que nos circundan. Cómo simbolizamos eso en un mapa, qué colores o anchos de líneas usamos es algo diferente. Así como XML está siendo utilizado en Internet para separar el contenido de la presentación, GML tendrá la misma aplicación en el mundo de la geografía.

GML está relacionado con la representación del contenido de los datos geográficos, pero también se puede utilizar para hacer mapas. Esto se logra con herramientas de dibujo que interpretan los datos GML. De cualquier forma, esto debe venir después que GML sea un estándar. Para hacer un mapa desde GML solo necesitamos diseñar los elementos en GML de forma que puedan ser interpretados para mostrarlos en forma de mapa. En la actualidad existen varios formatos potenciales



de gráficos como SVG (Scalable Vector Graphics), VML (Microsoft Vector Markup Language).

Un diseñador de mapas es utilizado de esa manera para ubicar elementos GML e interpretarlos utilizando estilos gráficos propios.

3.2.3 GML es texto

Como cualquier codificación XML, GML representa información geográfica en forma de texto. Lo que en otros tiempos fue considerado no adecuado, en la actualidad ha ganado muchos adeptos. Los textos tienen cierta simplicidad y visibilidad, además son fáciles de revisar y modificar.

Los formatos textos para codificar las geometrías y para ser utilizados en la geografía han sido empleados con anterioridad. El formato SAIF de la provincia de Columbia Británica es uno de los pioneros en este tema. Mas de 7000 ficheros de datos en escala 1:20 000, incluyendo topografía, planimetría (hidrografía, construcciones, caminos etc.) y toponimia están disponibles en este formato. Esta provincia ha mostrado que los formatos textos son prácticos y fáciles de utilizar. Otro ejemplo del uso del formato texto en la representación de complejos conjuntos de datos geométricos es VRML (Vector Markup Language). Modelos VRML grandes y complejos han sido desarrollados y navegados en Internet, todos utilizando codificaciones basadas en textos.

GML codifica con certeza objetos complejos, los cuales pueden estar compuestos por varios objetos. Por ejemplo, un objeto puede ser un aeropuerto, el cual a su vez está compuesto por otros objetos como las terminales, parqueos, vías de comunicación, pistas de aterrizaje, etc. La geometría de un objeto geográfico puede estar compuesta por varios elementos geográficos que pueden ser de tipo mixto.

Para codificar la geometría de un objeto simplemente se escribe:

```
<Feature fid="142" featureType="school" Description="A middle school">
  <Polygon GeometryID="1" SRS_Name="epsg.27354" NumRings="1">
    <LRing>
      <CDATA dimension="2">491888.999999459,5458045.9996
                          91904.999999458,5458044.9996
                          491908.999999462,5458064.9996
                          491924.999999461,5458064.999
```



491925.999999462,5458079.9996
 491977.999999466,5458120.999
 491953.999999466,5458017.9996

</CData>
 </LRing>
 </Polygon>
 </Feature>

Nótese que aquí no reflejamos otras propiedades, solo la geometría. Las otras propiedades se pueden adicionar a la codificación como sigue:

<Feature fid="142" featureType="school" Description="Balmoral Middle School">
 <Property Name="NumFloors" type="Integer" value="3" />
 <Property Name="NumStudents" type="Integer" value="987" />
 <Polygon GeometryID="1" SRS_Name="epsg.27354" NumRings="1">
 <LRing>
 <CData dimension="2">491888.999999459,5458045.9996
 91904.999999458,5458044.9996
 491908.999999462,5458064.9996
 491924.999999461,5458064.999
 491925.999999462,5458079.9996
 491977.999999466,5458120.999
 491953.999999466,5458017.9996

</CData>
 </LRing>
 </Polygon>
 </Feature>

3.2.4 GML Codifica los sistemas de Referencia Espacial.

Una componente esencial de un sistema geográfico es la georeferenciación de los objetos con respecto a la superficie de la tierra o a alguna estructura relacionada con ella. Esta versión de GML incorpora los Sistemas de Referencia Espacial básicos de la superficie de la tierra, los cuales son extensibles e incorporan las principales proyecciones de marcos de referencia geocéntricos usadas en la actualidad.

Esta versión de GML es capaz de codificar todos los sistemas de referencia que se pueden encontrar en el sitio Web de European Petroleum Standards Group (EPSG). Adicionalmente codifica esquemas de sistemas de referencia y unidades definidas por usuarios y en futuras versiones se prevé la codificación de sistemas de coordenadas locales.



3.2.5 Verificación automática de integridad de los Datos.

Uno de los aspectos mas importantes de XML es la habilidad para verificar la integridad de los datos. En la recomendación de XML 1.0 esto se cumple a través del DTD (Document Type Definition). El DTD especifica la estructura del documento XML y es la vía que utiliza el parser para la validación de un documento. GML es especificado también por un DTD y se prevé que las futuras versiones sean soportadas sobre esquemas de XML o mecanismos más flexibles que este, que ya data de las recomendaciones de W3C en el 2000.

Usando el DTD de GML, los servidores y los clientes pueden verificar fácilmente que los datos que se envían o se reciben cumplen con la especificación. Además, se logra que la verificación sea independiente de la gran variedad de los diferentes suministradores de parser que existen, así como independiente de los sistemas operativos, gestores de bases de datos y aplicaciones servidoras y visualizadoras.

3.2.6 GML es Transformable.

El aspecto más importante de XML es su transformabilidad. Es relativamente muy fácil escribir una transformación que lleve un XML relativo a un DTD hacia otro XML relativo a otro DTD. Esto no es más que la transformación que realizamos cuando generamos un flujo de elementos geográficos SVG a partir de un fichero de datos GML y estas transformaciones pueden ser realizadas usando una gran variedad de mecanismos dentro de los cuales se incluyen XSLT, Java, Javascript and C++ por solo mencionar algunos.

3.2.7 Otras Facilidades de GML

Con el surgimiento de otras especificaciones como Xlink y Xpoint, GML se proyecta no solo como una buena opción para el transporte de información geográfica sino también como una buena opción para el almacenamiento de este tipo de información. En la mayoría de las administraciones a nivel mundial los datos geográficos son recopilados por determinadas agencias estatales o particulares. El surgimiento de estas especificaciones puede llevar a una adecuada distribución y disponibilidad de la información.



3.3 Implementación de la especificaciones de FILTER versión 1.

Esta especificación define una codificación XML para expresiones de filtro, basada en la es la especificación de Interfase de OpenGIS, liberada en Septiembre del 2001.

Una expresión del filtro es una estructura construida para restringir las propiedades de un tipo de objeto, con el propósito de identificar un subconjunto de instancias de objeto para ser manejados de alguna la manera.

El propósito del documento es describir una codificación XML sobre el OGC Common Catalog Query Language (CQL) [2] como un sistema para la representación neutra de una consulta.

Usando el numeroso conjunto de herramientas XML disponible en la actualidad, esta representación de XML puede validarse y analizarse fácilmente, y entonces puede transformarse a otros lenguajes si se requiere modificar o recuperar instancias almacenadas en objetos persistentes.

Las normativas de esta especificación pueden ser chequeadas en el anexo B. El marco de trabajo, conceptos y metodología por probar, y el criterio para lograr la conformidad se especifica en ISO 19105: Información Geográfica—la Conformidad y Probado.

3.3.1 Introducción

Un filtro es cualquier expresión válida que puede formarse usando los elementos definidos en esta especificación. El elemento de la raíz <Filter> contiene la expresión por la que se crea, combinando los elementos definidos en esta especificación.

3.3.2 Codificación

El elemento de la raíz de una expresión de filtro, <el Filtro>, se define por el siguiente fragmento de esquema XML:

```
<xsd:complexType name="FilterType">  
  <xsd:choice>
```



```

<xsd:element ref="ogc:spatialOps" />
<xsd:element ref="ogc:comparisonOps" />
<xsd:element ref="ogc:logicOps" />
<xsd:element ref="ogc:FeatureId" maxOccurs="unbounded" />
</xsd:choice>
</xsd:complexType>

```

Los elementos <logicalOps>, <comparisonOps> y <spatialOps> son sustituidos por grupos de operadores lógicos, espacial y de comparación. Además, usando el elemento, <FeatureId> un filtro puede codificar referencias a uno o varias instancias de objetos.

3.3.3 Operadores Espaciales

Un operador espacial determina si sus argumentos geométricos satisfacen el declarado en la relación espacial. El operador devuelve Verdadero si la relación espacial se satisface y Falso en otro caso

3.3.3.1 Codificación

La codificación XML para los operadores espaciales se define por el siguiente fragmento de esquema:

```

<xsd:element name="spatialOps" type="ogc:SpatialOpsType" abstract="true" />
  <xsd:element name="Equals" type="ogc:BinarySpatialOpType"
    substitutionGroup="ogc:spatialOps" />
  <xsd:element name="Disjoint" type="ogc:BinarySpatialOpType"
    substitutionGroup="ogc:spatialOps" />
  <xsd:element name="Touches" type="ogc:BinarySpatialOpType"
    substitutionGroup="ogc:spatialOps" />
  <xsd:element name="Within" type="ogc:BinarySpatialOpType"
    substitutionGroup="ogc:spatialOps" />
  <xsd:element name="Overlaps" type="ogc:BinarySpatialOpType"
    substitutionGroup="ogc:spatialOps" />
  <xsd:element name="Crosses" type="ogc:BinarySpatialOpType"
    substitutionGroup="ogc:spatialOps" />
  <xsd:element name="Intersects" type="ogc:BinarySpatialOpType"
    substitutionGroup="ogc:spatialOps" />
  <xsd:element name="Contains" type="ogc:BinarySpatialOpType"
    substitutionGroup="ogc:spatialOps" />
  <xsd:element name="DWithin" type="ogc:DistanceBufferType"
    substitutionGroup="ogc:spatialOps" />

```



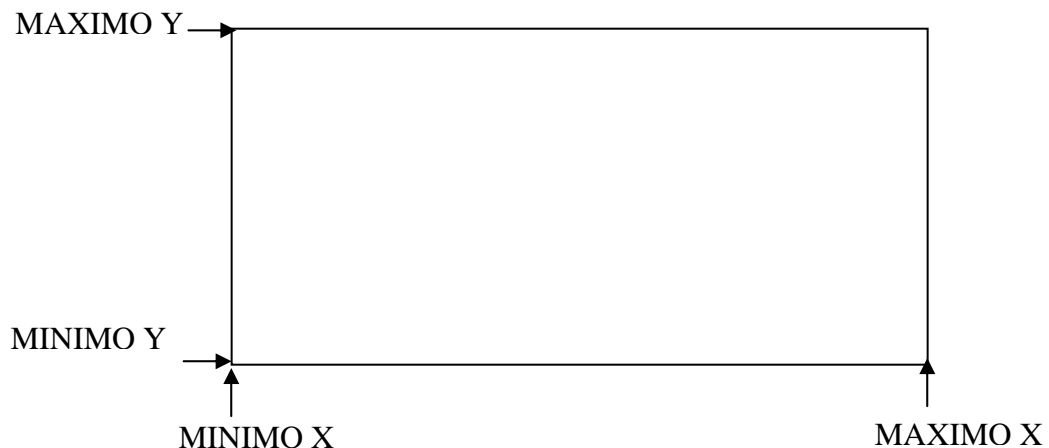
```

<xsd:element          name="Beyond"          type="ogc:DistanceBufferType"
substitutionGroup="ogc:spatialOps" />
  <xsd:element          name="BBOX"          type="ogc:BBOXType"
substitutionGroup="ogc:spatialOps" />
<xsd:complexType name="SpatialOpsType" abstract="true" />

```

Como está definido en esta especificación, los operadores espaciales son usados para probar si el valor de una propiedad geométrica, referenciada usando el nombre de la propiedad, y un valor geométrico como el literal satisface la relación espacial implicada por el operador. Por ejemplo, el operador <Overlap> evalúa si el valor de la propiedad geométrica especificada y el valor geométrico del literal especificado solapan espacialmente. Los valores geométricos literales son expresados usando GML [12].

El elemento <BBOX> se define como la forma convenientemente y más compacta de hacer la codificación para la consulta sobre un rectángulo, la cual se basa en la geometría de gml:Box. Es equivalente a la operación espacial <Not><Disjoint>...</Disjoint></Not>, lo que significa que el operador <BBOX> debe identificar todas las geometrías en que espacialmente actúan con el rectángulo de alguna manera.



La semántica de los otros operadores Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects, and Contains se define en sección 3.2.19.2 del *OpenGIS Simple Feature Specification for SQL* [5].

3.3.4 Operadores de Comparación.



Un operador de la comparación es usado para formar expresiones que evalúan la comparación matemática entre dos argumentos. Si los argumentos satisfacen la comparación entonces la expresión evaluada es verdadera, en otro caso es falsa.

Codificación.

La codificación XML para los operadores de comparación se define por el siguiente fragmento de esquema:

```
<xsd:element name="comparisonOps" type="ogc:ComparisonOpsType"
  abstract="true" />
  <xsd:element name="PropertyIsEqualTo" type="ogc:BinaryComparisonOpType"
    substitutionGroup="ogc:comparisonOps" />
  <xsd:element name="PropertyIsNotEqualTo" type="ogc:BinaryComparisonOpType"
    substitutionGroup="ogc:comparisonOps" />
  <xsd:element name="PropertyIsLessThan" type="ogc:BinaryComparisonOpType"
    substitutionGroup="ogc:comparisonOps" />
    <xsd:element name="PropertyIsGreaterThan"
      type="ogc:BinaryComparisonOpType" substitutionGroup="ogc:comparisonOps"
      />
      <xsd:element name="PropertyIsLessThanOrEqualTo"
        type="ogc:BinaryComparisonOpType" substitutionGroup="ogc:comparisonOps"
        />
        <xsd:element name="PropertyIsGreaterThanOrEqualTo"
          type="ogc:BinaryComparisonOpType" substitutionGroup="ogc:comparisonOps"
          />
          <xsd:element name="PropertyIsLike" type="ogc:PropertyIsLikeType"
            substitutionGroup="ogc:comparisonOps" />
            <xsd:element name="PropertyIsNull" type="ogc:PropertyIsNullType"
              substitutionGroup="ogc:comparisonOps" />
            <xsd:element name="PropertyIsBetween" type="ogc:PropertyIsBetweenType"
              substitutionGroup="ogc:comparisonOps" />
  <xsd:complexType name="ComparisonOpsType" abstract="true" />
```

El *Common Catalog Query Language* [4] define un juego estándar de operadores comparación. Además del juego estándar (=, <, >, >=, <=, <>) de operadores de la comparación, esta especificación define los elementos *<PropertyIsLike>*, *<PropertyIsBetween>* y *<PropertyIsNull>*.

El elemento *<PropertyIsLike>* tiene la intención de codificar la comparación de una cadena de caracteres con un determinado patrón con el cual haga juego. El patrón



se define por una combinación de regular los caracteres, caracteres comodines y otros caracteres especiales.

El elemento *<PropertyIsNull>* codifica a un operador que verifica para ver si el valor de la propiedad es NULO. Un NULO es equivalente a ningún valor presente. El valor 0 es un valor válido

y no es considerado NULO.

El elemento *<PropertyIsBetween>* se define como una manera compacta de codificar el chequeo de un rango, donde los límites inferiores y superiores son incluidos dentro del rango.

3.3.5 Operadores Lógicos.

Un operador lógico es usado para combinar una o más expresiones condicionales. El operador lógico AND da como respuesta un valor Verdadero, si todas las expresiones combinadas son verdaderas. El operador OR da como respuesta un valor Verdadero, si cualquiera de las expresiones combinadas es Verdadera. El operador NOT invierte el valor lógico de una expresión.

Codificación.

La codificación XML para los operadores lógicos se define por el siguiente fragmento de esquema:

```
<xsd:element name="logicOps" type="ogc:LogicOpsType" abstract="true" />
  <xsd:element name="And" type="ogc:BinaryLogicOpType"
    substitutionGroup="ogc:logicOps" />
  <xsd:element name="Or" type="ogc:BinaryLogicOpType"
    substitutionGroup="ogc:logicOps" />
  <xsd:element name="Not" type="ogc:UnaryLogicOpType"
    substitutionGroup="ogc:logicOps" />
<xsd:complexType name="LogicOpsType" abstract="true" />
```

Los elementos *<And>*, *<Or>* and *<Not>* pueden usarse para combinar el valores escalares, espacial y otras expresiones lógicas para formar expresiones compuestas más complejas.



Los elementos `comparisonOps` y `spatialOps` son elementos abstractos que pueden sustituirse por los operadores de la comparación (<, >, = ,etc...) y los operadores espaciales.

3.3.6 Identificador de Objetos.

Un identificador de objetos significa representar un único identificador para una instancia de un objeto geográfico, dentro del contexto del servicio Web que está sirviendo el objeto.

Codificación.

El elemento `<FeatureId>` es definido por el siguiente fragmento de esquema XML:

```
<xsd:element name="FeatureId" type="ogc:FeatureIdType" />
<xsd:complexType name="FeatureIdType">
  <xsd:attribute name="fid" type="xsd:anyURI" use="required" />
</xsd:complexType>
```

El elemento `<FeatureId>` puede ser usado para referenciar instancias de objetos geográficos en expresiones de filtro o en otras documentaciones XML.

3.3.7 Literales.

Esta sección define cómo la codificación del filtro definió en esta especificación el código de los valores literales. Un valor literal es cualquier parte de una declaración o expresión que serán usadas exactamente como se especifica, no como una variable u otro elemento.

Codificación.

El siguiente fragmento de esquema define el elemento `<Literal>` :

```
<xsd:element name="Literal" type="ogc:LiteralType"
  substitutionGroup="ogc:expression" />
<xsd:complexType name="LiteralType">
  <xsd:complexContent mixed="true">
    <xsd:extension base="ogc:ExpressionType">
      <xsd:sequence>
        <xsd:any minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```



El elemento **<Literal>** es usado para codificar valores escalares y geométricos. En el caso de los valores geométricos deben validado contra el esquema de XML *geometry.xsd*.

3.3.8 Capacidades.

Esta sección del documento define un esquema de capacidades, el cual puede ser incluido el documento de las capacidades del servicio que usa la codificación del filtro. Los documentos describen cuales capacidades del filtro son soportadas por el servicio.

Las capacidades del filtro son divididas en dos categorías: las capacidades espaciales y las capacidades escalares. El siguiente fragmento de XML define el elemento raíz de las capacidades de filtrado:

```
<xsd:element name="Filter_Capabilities">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element
        type="ogc:Spatial_CapabilitiesType" />
        <xsd:element
        type="ogc:Scalar_CapabilitiesType" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

name="Spatial_Capabilities"

name="Scalar_Capabilities"

El ejemplo siguiente muestra un fragmento de capacidades para un servicio que soporta todas las capacidades de filtrado definidas en este documento e incluyen una lista de funciones:

```
<Filter_Capabilities>
  <Spatial_Capabilities>
    <Spatial_Operators>
      <BBOX />
      <Equals />
      <Disjoint />
      <Intersect />
      <Touches />
      <Crosses />
      <Within />
      <Contains />
```



```

    <Overlaps />
    <Beyond />
  </Spatial_Operators>
</Spatial_Capabilities>
<Scalar_Capabilities>
  <Logical_Operators />
  <Comparison_Operators>
    <Simple_Comparisons />
    <Like />
    <Between />
    <NullCheck />
  </Comparison_Operators>
  <Arithmetic_Operators>
    <Simple_Arithmetic />
  </Arithmetic_Operators>
  <Functions>
    <Function_Names>
      <FunctionName nArgs="1">MIN</FunctionName>
      <FunctionName nArgs="1">MAX</FunctionName>
    </Function_Names>
  </Functions>
</Scalar_Capabilities>
</Filter_Capabilities>

```

Otras funciones

```

    <FunctionName nArgs="1">SIN</FunctionName>
    <FunctionName nArgs="1">COS</FunctionName>
  </Function_Names>
</Functions>
</Arithmetic_Operators>
</Scalar_Capabilities>
</Filter_Capabilities>

```

3.4 Implementación de las especificaciones de Servicio de Objetos Geográficos.

El Servidor de Rasgos Geográficos (WFS por sus siglas en inglés) es la especificación de interfase de OpenGIS, liberada en Septiembre del 2002 que permite al cliente obtener datos geoespaciales codificado en GML (*Geography Markup Language*) desde múltiples servicios de objetos en Web; similar al Servicio de Mapas en Web (WMS), que permite al cliente obtener imágenes de mapas para ser visualizados por múltiples servicios de mapas en Internet.

WFS especifica tres esquemas base para codificar sus operaciones: WFS-basic.xsd, WFS-capabilities.xsd y WFS-transaction.xsd, ver Anexo D.



Los requisitos para un servidor de objetos geográficos son:

- Las interfaces deben ser definidas en XML.
- GML debe ser usado para expresar objetos dentro de la interfase.
- Como mínimo un WFS debe poder presentar objetos usando GML.
- El predicado o filtro de lenguaje será definido en XML y ser derivado de CQL como esta definido en el "OpenGIS Catalogue Interface Implementation Specification".
- El almacenes de datos usado para almacenar objetos geográficos puede ser transparentes a las aplicaciones del cliente y solo verán los datos a través de la interfase del WFS.
- El uso de un subconjunto de expresiones XPath para referirse a las propiedades.

El propósito será describir una interfaz servidor de objetos geográficos, como la de la Figura 14:

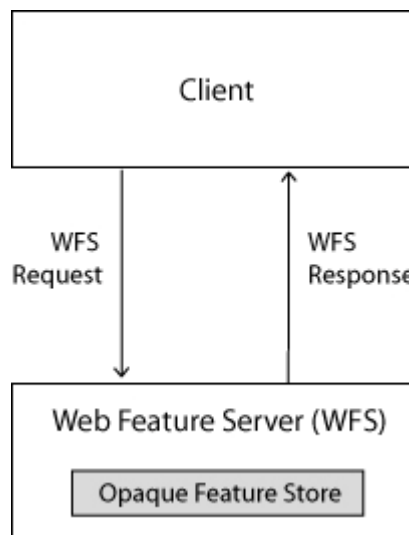


Figura 14: Interfaz de un WFS.

El WFS habilita tecnologías para el acceso directo a objetos geográficos en almacenes de datos remotos y distribuidos en un formato estándar, como muestra la Figura 15:

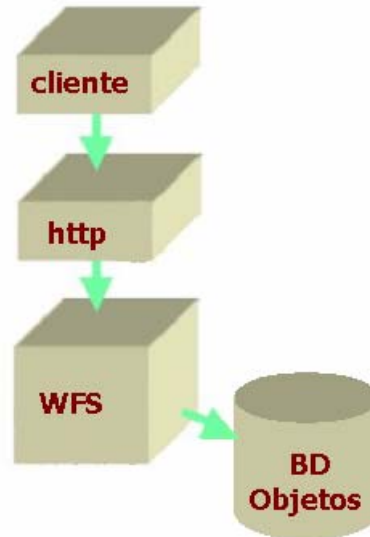


Figura 15. Acceso a los datos usando el WFS.

Permite acceder de forma directa a objetos geográficos como Puntos, Líneas y Polígonos, bajo el uso de restricciones espaciales o temáticas.

Un WFS implementa las operaciones GetCapabilities, DescribeFeatureType y GetFeature y es considerado un WFS de solo lectura, el cual es el primer paso para llegar al WFS de Negociación o Transaccional que no es más que un WFS básico que implementa la operación **Transaction** (Insert, Delete, Update y LockFeature).

En sentido general las tres operaciones anteriores serán expuestas a través de una interfase que se nombra "Request", mediante ella se le dirá al servidor que operación se está encuestando y el procesamiento de peticiones (Processing request) a seguir sería de la siguiente forma:

- Una aplicación cliente solicitaría el documento de capacidades del WFS. Tal documento contiene una descripción de todas las operaciones que el WFS soporta y una lista de todos los objetos que puede servir.
- Una aplicación cliente (opcional) hace una petición a un WFS de la definición de uno o más tipos de objetos que un WFS puede servir.
- Basado en la definición de tipos de objetos, la aplicación cliente genera una petición.



- La petición es enviada a un Servidor Web.
- El WFS es invocado para leer y servir dicha petición.
- Cuando el WFS ha completado el procesamiento de la petición, generará un reporte y lo dará de nuevo al cliente. En el caso de que un error ocurriera, el reporte indicará ese hecho.

La operación **GetCapabilities** devuelve un documento XML que brinda información sobre el servicio, la lista de solicitudes que el WFS puede manipular, el listado de tipos de objetos que están disponibles en el WFS y las operaciones definidas para el filtro, en caso de que este exista.

La operación **DescribeFeatureType** devuelve el esquema en XML que describe la estructura de todos los tipos de objetos que el WFS puede manipular.

La operación **GetFeature** devuelve un documento GML que sirve las instancias de objetos solicitadas en la consulta al WFS.

La operación **Transaction** es usada para describir las operaciones de transformación de los datos que son aplicadas a las instancias de objetos accesibles por la Web. El WFS puede realizar la operación de transacción directamente o traducirlo al lenguaje del gestor de datos con el cual se conecta y este a su vez ejecutar la transacción. Al ser completada la operación, el WFS genera como respuesta un documento XML, indicando el estado de completamiento de la transacción.

Esta implementación es un paso lógico para describir las operaciones de manipulación de datos en objetos geográficos, usando el protocolo http, el cual incluye la habilidad de obtener o consultar objetos basado en restricciones generadas por el cliente y pasadas al WFS usando este protocolo. Esto brinda la factibilidad técnica de acceder a la información ubicada en diferentes almacenes de datos y la facilidad de usar estas factibilidades técnicas para presentar los datos al cliente en una aplicación visual, útil y significativa como son los servidores de mapas basado en las especificaciones de OpenGIS, como es el caso el servidor de mapas desarrollado por nuestro grupo



3.4.1 Requerimientos, análisis y arquitectura.

El Servidor de Objetos geográficos es una aplicación CGI que interactúa desde Internet Information Server con el servidor de Bases de Información Geográfica. Implementa las interfaces definidas por OpenGIS y constituye un intermediario entre el cliente y el servicio NT. El servidor recibe una solicitud en forma de URL, la procesa y la pasa al servicio geomixms.exe, el resultado se devuelve al browser del cliente.

Este servicio se desarrollo de forma que pudiera operar en sistemas operativos servidores como Windows NT, 2000, XP. El Servidor de Objetos Geográficos fue desarrollado utilizando como herramienta de programación Borland Delphi7 por sus facilidades en la implementación de servidores Web, tanto CGI como ISAPI.

Los componentes que conforman el Servidor de Objetos para Internet implementado son:

- 3 Bibliotecas COM de vínculo Dinámico.
 - LocalBIGManager.dll (Manejo de una BIG)
 - FeatureGeometries.dll (Manejo de las objetos geográficos)
 - VagoSRS.dll (Manejo de los Sistemas de Referencia Espacial)
- 1 Aplicación Web (CGI) (WFSserver.exe)
- 1 Servicio Windows NT (geomixms.exe)
- 1 Base de Información Geográfica.

GeomixMS.exe es el servidor de Bases de Información Geográfica implementado como un servicio Windows NT que a solicitud del servidor de mapas (mapserver.exe) recupera los datos de la BIG y los devuelve en forma de imagen o GML.

3.4.2 Operación GetCapabilities.

La operación **GetCapabilities** de un WFS debe ser capaz de describir sus capacidades y las devuelve un documento XML conformado por 4 secciones principales:



1. **Servicio:** Esta sección provee información sobre el servicio. La tabla siguiente muestra algunos de los elementos que componen esta sección.

NOMBRE DEL ELEMENTO	DESCRIPCIÓN
Name	El nombre que el proveedor de servicios asigna al WFS.
Title	El título es el nombre que identifica de forma breve el servidor en menús.
Abstract	Descripción narrativa para mayor información sobre el servidor.
AccessConstraints	Contiene la descripción de todas las restricciones de acceso impuestas por el proveedor de servicios en el WFS o los datos recuperados desde el servicio.

2. **Capacidades:** Especifica la lista de consultas que el WFS puede manipular. El WFS implementado publica:

- GetCapabilities.
- DescribeFeatureType.
- GetFeature.

3. **Listado de tipos de objetos:** Especifica el listado de tipos de objetos que están disponibles en el WFS e información adicional como por ejemplo el Sistema de Referencia Espacial. La tabla siguiente muestra algunos de los elementos que componen esta sección.

ELEMENTO	DESCRIPCIÓN
Name	El nombre del tipo de objeto. Es obligatorio.



Title	El título es el nombre que identifica de forma breve el tipo de objeto en menús.
Abstract	Descripción narrativa para mayor información sobre el tipo de objeto.
SRS	Es usado para indicar cual Sistema de Referencia Espacial debe ser usado para expresar el estado de los objetos.
LatLongBoundingBox	Indica el rectángulo que encierra el objeto en el Sistema de Referencia Espacial asociado.

4. Capacidades de Filtrado: Especifica las operaciones de filtrado soportadas por el WFS e implementadas según el esquema de la Especificación de Codificación de Filtros.

El siguiente ejemplo muestra las capacidades del WFS Básico implementado como respuesta a la consulta realizada al servidor:

Request= GetCapabilities

En respuesta a esta consulta, el WFS generó un documento como el que se muestra a continuación:

```
<WFS_Capabilities xmlns="http://www.opengis.net/wfs" version="1.0.0">
  <Service>
    <Name>WFS GEOMIX </Name>
    <Title>GEOMIX Web Feature Service</Title>
    <Abstract>WFS Desarrollado y mantenido por GEOMIX
      GEOCUBA Manicaragua</Abstract>
  </Service>
  <Capability>
    <Request>
      <GetCapabilities />
      <DescribeFeatureType />
      <GetFeature />
    </Request>
  </Capability>
  <FeatureTypeList>
    <FeatureType>
```



```

<Name>Municipios</Name>
<SRS>GCS_Cuba</SRS>
  <LatLongBoundingBox minx="22.5601977734903" miny="-
    82.9814322601738" maxx="23.1880387650422" maxy="-
    81.5842493152693" />
</FeatureType>
<FeatureType>
  <Name>Cayos</Name>
  <SRS>GCS_Cuba</SRS>
    <LatLongBoundingBox minx="22.3445090099167" miny="-
      82.6930461149863" maxx="22.6674004557013" maxy="-
      82.2409008580577" />
    </FeatureType>
  <FeatureType>
    <Name>Pueblos</Name>
    <SRS>GCS_Cuba</SRS>
      <LatLongBoundingBox minx="22.6216693323846" miny="-
        82.9476418688805" maxx="23.1802598614333" maxy="-
        81.6059572674499" />
      </FeatureType>
    <FeatureType>
      <Name>Manzanas</Name>
      <SRS>GCS_Cuba</SRS>
        <LatLongBoundingBox minx="22.7747086168612" miny="-
          82.7822530026526" maxx="22.9952305487886" maxy="-
          82.0171571700793" />
        </FeatureType>
    <FeatureType>
      <Name>Rotulos</Name>
      <SRS>GCS_Cuba</SRS>
        <LatLongBoundingBox minx="22.3436910709984" miny="-
          82.9493543421145" maxx="23.186360841849" maxy="-
          81.6475448569627" />
        </FeatureType>
    <FeatureType>
      <Name>TPI</Name>
      <SRS>GCS_Cuba</SRS>
        <LatLongBoundingBox minx="19.7477089671463" miny="-
          85.7673666169381" maxx="19.7477123126382" maxy="-
          85.7673596010889" />
        </FeatureType>
    <FeatureType>
      <Name>Calles</Name>
      <SRS>GCS_Cuba</SRS>
        <LatLongBoundingBox minx="22.7082778734469" miny="-
          82.7804436922547" maxx="22.9947415574967" maxy="-
          82.0156953151841" />
    </FeatureType>
  </FeatureType>
</FeatureType>

```



```

</FeatureType>
<FeatureType>
  <Name>Rot_Mun</Name>
  <SRS>GCS_Cuba</SRS>
  <LatLongBoundingBox minx="19.7477092340121" miny="-
    85.7673671139951" maxx="19.7477130376043" maxy="-
    85.7673568134757" />
</FeatureType>
</FeatureTypeList>
<Filter_Capabilities>
  <Spatial_Capabilities>
    <Spatial_Operators>
      <BBOX />
      <Equals />
      <Disjoint />
      <Intersect />
      <Touches />
      <Crosses />
      <Contains />
      <Overlaps />
    </Spatial_Operators>
  </Spatial_Capabilities>
  <Scalar_Capabilities>
    <Comparison_Operators>
      <Simple_Comparisons />
      <Like />
      <Between />
      <NullCheck />
    </Comparison_Operators>
  </Scalar_Capabilities>
</Filter_Capabilities>
</WFS_Capabilities>

```

Este documento XML expresa en su sección **Service** algunos elementos sobre el suministrador del servicio, que es el Grupo de Desarrollo de Software GEOMIX de la Oficina de GEOCUBA en Manicaragua; en la sección **Capabilities** cuáles son las operaciones que el WFS está sirviendo, que en nuestro caso son las ya explicadas **GetCapabilities**, **DescribeFeatureType**, **GetFeature** y **Transaction**, en la sección **FeatureTypeList (Listado de tipos de objetos)** brinda el listado de todos los tipos de objetos que estaba sirviendo el Servidor de Mapas en ese momento, así como el Sistema de Referencia Espacial (SRE) que debería ser usado para expresar el estado de los objetos y rectángulo que encierra el objeto en ese SER, y finalmente la



sección **Filter_Capabilities (Capacidades de Filtrado)** donde se expone los operadores espaciales y los de comparación que se pueden usar en el filtrado de los objetos.

3.4.3 Operación DescribeFeatureType.

La función de la operación **DescribeFeatureType** es generar la descripción de un esquema de los tipos de objetos servidos por la implementación del WFS. La descripción del esquema define como la implementación del WFS espera codificar las instancias de objetos para las entradas y como las instancias de objetos serán generados para las salidas.

A continuación describiremos como formular una consulta usando el estilo CGI con la codificación par llave-valor. La codificación es como muestra la tabla siguiente según la especificación de OGC:

Componente URL	Opcional/ Obligatorio	Por Defecto	Descripción
REQUEST=DescribeFeatureType	Obligatorio		Nombre de la consulta.
TYPENAME	Opcional		Lista de los tipos de objetos a describir separados por coma. Si no se especifica valor se consideran todos.
OUTPUTFORMAT	Opcional	XML	Formato de salida que se usará para describir los objetos.

A continuación aparece un ejemplo de consulta realizada al servidor sobre algunos de los tipos de objetos soportados:

Request= DescribeFeatureType&typename=Municipios,Cayos,Pueblos



El XML que se muestra a continuación es la respuesta brindada a esta consulta:

```
<?xml version="1.0" ?>
<schema>
  <!-- definición de los elementos globales -->
  <element name="Municipios" type="myns:Municipios"
    substitutionGroup="gml:_Feature" />
  <element name="Cayos" type="myns:Cayos"
    substitutionGroup="gml:_Feature" />
  <element name="Pueblos" type="myns:Pueblos"
    substitutionGroup="gml:_Feature" />
  <!-- definición de los tipos complejos -->
  <complexType name="Municipios">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="Geometria"
            type="gml:PolygonPropertyType" nillable="false" />
          <element name="Municipio" nillable="true">
            <simpleType>
              <restriction name="string">
                <maxLength value="100" />
              </restriction>
            </simpleType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="Cayos">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="Geometria"
            type="gml:PolygonPropertyType" nillable="false" />
          <element name="Tipo" nillable="true">
            <simpleType>
              <restriction name="string">
                <maxLength value="100" />
              </restriction>
            </simpleType>
          </element>
          <element name="nombre" nillable="true">
            <simpleType>
              <restriction name="string">
```



```

        <maxLength value="100" />
      </restriction>
    </simpleType>
  </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="Pueblos">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element
          name="Geometria"
          type="gml:PointPropertyType" nillable="false" />
        <element name="Text" nillable="true">
          <simpleType>
            <restriction name="string">
              <maxLength value="50" />
            </restriction>
          </simpleType>
        </element>
        <element name="Angle" type="double" nillable="true" />
        <element name="Size" type="double" nillable="true" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
</schema>

```

Como se puede observar este esquema se divide en dos partes fundamentales; la primera donde se definen los objetos de forma global y la segunda donde se describe cada tipo de objeto de forma específica. En el ejemplo se brinda un esquema que solo relaciona los tipos de objetos vinculados con la consulta: Municipios, Cayos, Pueblos y Manzanas. De ellos se brinda su descripción según la implementación del WFS. Para el último objeto relacionado (Manzanas), por ejemplo, se describe que en su estructura tiene un elemento que es la **geometría**, y que la misma es de tipo punto, que tiene otro elemento que es un texto con tres características principales: cadena de longitud máxima 50 caracteres, un ángulo de tipo real y un tamaño de tipo real.



3.4.4 Operación GetFeature.

La operación **GetFeature** devuelve un documento GML que sirve las instancias de objetos solicitadas en la consulta al WFS. Además, el cliente debe poder especificar que propiedad del objeto deberá traer y también deberá poder construir el filtro espacial y no espacial para especificar de qué objetos deberá brindar la información.

A continuación describiremos como formular una consulta usando el estilo CGI con la codificación par llave-valor. La codificación es como muestra la tabla siguiente según la especificación de OGC:

Componente URL	Obligatorio /Opcional	Descripción
REQUEST=GetFeature	Obligatorio	El nombre de la consulta al WFS.
PROPERTYNAME	Opcional	Listas de las propiedades, encerradas entre paréntesis “()” y separadas por coma “,”, que debe ser especificada para cada tipo de objeto que va a ser consultado. El carácter “*” indica que se deben devolver todas las propiedades. La ausencia de este componente indica que todas las propiedades deben ser servidas. Debe existir una relación 1:1 entre cada elemento de TYPENAME y las listas de PROPERTYNAME.
MAXFEATURES=N	Opcional	Un número entero positivo que indica la cantidad máxima de objetos que el WFS debe retornar en respuesta a la consulta. Si no se especifica este valor entonces se devuelven todas



		las instancias que existan.
TYPENAME (Opcional si se especifica el FEATUREID.)	Obligatorio	La lista de tipos de objetos a consultar.
FEATUREID Mutuamente exclusivo con FILTER y BBOX)	Opcional	Una lista enumerada de instancias de objetos a servir, seleccionados por el identificador del objeto.
FILTER (Requisito previo: TYPENAME) (Mutuamente exclusivo con FEATUREID y BBOX)	Opcional	La especificación de filtro describe el conjunto de objetos sobre los cuales se debe operar. El filtro es definido por una especificación OG (Filter EncodingSpecification). Si se usa el parámetro FILTER, entonces debe ser especificada para cada uno de los tipos de objetos definidos en el parámetro TYPENAME y deben ser encerrados entre paréntesis "(y)".
BBOX (Requisito previo: TYPENAME) (Mutuamente exclusivo con FEATUREID y FILTER.)	Opcional	En línea con FEATUREID o FILTER el cliente puede especificar el rectángulo que define el área de donde desea le sean servidos los objetos.

Seguidamente mostraremos algunos ejemplos de consultas realizada al servidor sobre los objetos soportados:

- Consulta todas las propiedades de todas las instancias del objeto **Manzanas**.

REQUEST=GetFeature&TYPENAME= Manzanas



- Consulta la propiedad **Geometría** de todas las instancias del objeto **Manzanas**.

REQUEST=GetFeature&PROPERTYNAME= Geometric &TYPENAME= Manzanas

- Consulta todas las propiedades de las instancias de los objetos con identificador 3 y 8.

REQUEST=GetFeature&FEATUREID= Manzanas.3, Manzanas.8

- Consulta todas las propiedades de las instancias de objetos **Manzanas** que están dentro del rectángulo definido por el parámetro **Box**:

REQUEST=GetFeature&TYPENAME= Manzanas &

FILTER=<Filter><Within><PropertyName> Geometria

PropertyName><gml:Box><gml><coordinates>10,10

20,20</gml:coordinates></gml:Box></Within></Filter>

El GML que se muestra a continuación es la respuesta brindada a la consulta del último ejemplo:

```
<FeatureCollection>
  <boundedBy>
    <Box>
      <coordinates>440451.247,356140.247,441450.753,357139.753
    </coordinates>
    </Box>
  </boundedBy>
  <featureMember>
    < Manzanas fid="56587BE809184555B609418C0613CD0F">
      < Geometria >
        <polygon SRS="Sistema RS">
          <outerBoundaryIs>
            <LinearRing>
              <coordinates>
                439909.721,357698.803,439897.262,35769
                7.229,439885.098,357694.106,439873.422,
                357689.483,439862.417,357683.433,43985
                2.258,357676.051,439843.103,357667.455,
                439835.098,357657.779,439828.369,35764
                7.175,439823.022,357635.812,439819.142,
                357623.869,439816.789,357611.533,43981
                6,357599,439816.789,357586.467,439819.
```



```

142,357574.131,439823.022,357562.188,4
39828.369,357550.825,439835.098,357540
.221,439843.103,357530.545,439852.258,3
57521.949,439862.417,357514.567,439873
.422,357508.517,439885.098,357503.894,4
39897.262,357500.771,439909.721,357499
.197,439922.279,357499.197,439934.738,3
57500.771,439946.902,357503.894,439958
.578,357508.517,439969.583,357514.567,4
39979.742,357521.949,439988.897,357530
.545,439996.902,357540.221,440003.631,3
57550.825,440008.978,357562.188,440012
.858,357574.131,440015.211,357586.467,4
40016,357599,440015.211,357611.533,440
012.858,357623.869,440008.978,357635.8
12,440003.631,357647.175,439996.902,35
7657.779,439988.897,357667.455,439979.
742,357676.051,439969.583,357683.433,4
39958.578,357689.483,439946.902,357694
.106,439934.738,357697.229,439922.279,3
57698.803,439909.721,357698.803</coordi
nates>
</LinearRing>
</outerBoundaryIs>
</polygon>
</ Geometria >
<Text>
  <Texto>Manzana_315</Texto>
  <Angle>35</Angle>
  <Size>24</Size>
</Text>
</ Manzanas >
</featureMember>
<featureMember>
<featureMember>
</FeatureCollection>

```

Este documento expresa en dos módulos fundamentales, las propiedades de los objetos manzanas que cumplen con la condición de que su geometría está dentro del recuadro especificado por el filtro. En el primer módulo, **boundedBy**, encontramos las coordenadas del recuadro que contiene a todos los objetos (3) que cumplen la condición especificada y en el segundo tenemos una lista de estos objetos con la



especificación sus propiedades y donde se expresa de forma trascendental todos las coordenadas, **coordinates**, que componen la geometría de dichos objetos.

3.4.5 Operación Transaction.

La operación **Transaction** devuelve un documento XML, indicando el estado de completamiento de la transacción, la misma puede contener varios, o ningún, elementos **INSERT, UPDATE o DELETE**, los cuales describen las operaciones de crear, modificar o eliminar instancias de objetos accesibles por Web.

El elemento **<Insert>** es usado para crear nuevas instancias de Objetos. El estado inicial del objeto que será creado es expresado en formato GML y debe ser validado con el esquema generado por la operación DescribeFeatureType. Pueden existir múltiples elementos **<Insert>** dentro de una misma transacción y con un solo elemento **<Insert>** se pueden crear varias instancias de objetos.

El elemento **<Update>** describe una operación de modificación que será aplicada a un objeto o conjunto de objetos del mismo tipo. Pueden existir múltiples elementos **<Update>** dentro de una misma transacción. Cada elemento **<Update>** contiene uno o más elementos **<Property>** que especifican el nombre y valor a reemplazar para la propiedad del objeto especificada en el atributo obligatorio **typeName**. El alcance del elemento **<Update>** está restringido por el uso del elemento **<Filter>** (La especificación de filtro describe el conjunto de objetos sobre los cuales se debe operar. El filtro es definido por una especificación OG (Filter EncodingSpecification)).

El elemento **<Delete>** es usado para indicar que uno o más instancias de objetos serán borrados. El alcance de esta operación también es restringida por el uso del elemento **<Filter>** descrito en la especificación de implementación de Filtro.

En respuesta a una consulta de transacciones el WFS debe generar un documento XML indicando el estatus de terminación de la transacción y adicionalmente incluye, para la consultas que incluye operaciones de creación de objetos, los identificadores de los nuevos objetos creados. En el caso que la transacción falle al ejecutarse, el WFS debe ser capaz de indicar esto en su respuesta.



El elemento **<Status>** es usado para indicar el estado de completamiento de la Transacción y los posibles estados son:

SUCCESS	La transacción fue completada con éxito.
PARTIAL	La transacción fue completada parcialmente y los datos pueden estar en un estado inconsistente.
FAILED	Fue encontrada una excepción procesando uno o varios elementos contenidos en la Transacción.

El siguiente es un ejemplo de XML generado por un WFS:

```
<?xml version="1.0" ?>
<wfs:TransactionResponse                                version="1.0.0"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs../wfs/1.0.0/WFS-
    transaction.xsd">
  <wfs:InsertResult handle="TPI">
    <ogc:FeatureId fid=" TPI.4567" />
    <ogc:FeatureId fid=" TPI.4568" />
    <ogc:FeatureId fid=" TPI.4569" />
  </wfs:InsertResult>
  <wfs:InsertResult handle=" TPI ">
    <ogc:FeatureId fid=" TPI.4569" />
  </wfs:InsertResult>
  <wfs:InsertResult handle=" TPI ">
    <ogc:FeatureId fid=" TPI.389345" />
  </wfs:InsertResult>
  <wfs:TransactionResult handle="TX01">
    <wfs:Status>
      <wfs:SUCCESS />
    </wfs:Status>
  </wfs:TransactionResult>
</wfs:TransactionResponse>
```



En este capítulo detallaremos en la dinámica de las principales pruebas realizadas al servidor, los principales resultados obtenidos en el tiempo de creación, prueba y explotación, así como el alcance real y limitaciones que hoy posee el producto que se obtuvo y las conclusiones a que hemos arribado con este esfuerzo..

De vital importancia es exponer las principales líneas de trabajo futuro, pues es el camino a seguir para mantener vivo los resultados con que hoy contamos

4.1 Pruebas.

Durante el desarrollo del proyecto se presentaron varios escenarios que facilitaron la implementación, prueba y consolidación del mismo. Estos se caracterizaron por su heterogeneidad de arquitecturas, sistemas operativos, gestores de bases de datos, browser y tecnologías en forma general. Los servicios obtenidos, donde consideramos el de Objetos Geográficos, han sido puestos en explotación en una red local (Nuestra Oficina y Gobierno Municipal), en una red corporativa (ETECSA) y en Internet (Infraestructura de Datos Espaciales de la República de Cuba (IDERC)).

Nuestra red local es donde nacen y dan sus primeros pasos nuestros servicios y herramientas. En ella se ponen a punto los resultados antes de ser llevados a otros escenarios.

El gobierno municipal ha sido uno de nuestros principales clientes de aplicaciones de sistemas de información geográfica y espacio propicio de pruebas con elementos reales del entorno.

Ya validado el servicio en estos entornos, se pudo entonces probar el mismo en las aplicaciones que con anterioridad se le habían comercializado a la Empresa Telefónica de Cuba, específicamente a la Gerencia de Telefonía Pública, la cual contaba ya con las prestaciones del servicio de mapas y de un cliente que interactuaba con el mismo en todas las provincias del país; y en Internet, en los servicios que presta el Geoportal cubano, el cual tuvo varios meses de prueba, tiempo que aprovechamos para validar las prestaciones del servidor.



4.2 Resultados.

Con la introducción del nuevo servicio se logró satisfacer un conjunto de solicitudes hechas por el cliente, las cuales no era posible realizar sin las nuevas prestaciones del WFS, como son la actualización de la cartografía desde Web, las consultas espaciales, la utilización de filtros temáticos y espaciales en dichas consultas, etc; y se facilitaron las implementaciones que ya estaban en explotación, como son el Sistema de Información Geográfica para la Telefonía Pública y el Diccionario de Nombres Geográficos de Cuba, lo que se revirtió en mejoras del servicio prestado en forma general y mayor satisfacción del cliente.

Otro gran paso que ha demostrado la utilidad y estabilidad del servicio ha sido la publicación del sitio de la Infraestructura de Datos Espaciales de la república de Cuba (IDERC), organización creada por decisión del Consejo de Ministros, quien encomienda a la ONHG y a GEOCUBA la publicación de algunas de bases cartográficas del país y servicios de entidades en un portal [2], el cual fue lanzado oficialmente en el recién concluido evento de Informática 2005. En este portal se pueden explorar actualmente tres aplicaciones de servicio de mapas y de objetos geográficos. Una es un visor genérico de la cartografía disponible en el servidor; otra es el Diccionario de Nombres Geográficos de Cuba para Internet, aplicación que posibilita poner a disposición de un mayor número de personas esta importante obra de obligada consulta para geógrafos, cartógrafos y todo aquel que se interese por la actividad toponímica. Además de recoger la ubicación del topónimo sobre la cartografía, y a parir de este obtener la descripción e imágenes asociadas, podrán realizarse búsquedas de los topónimos por diferentes criterios: nombre, tipo de accidente, tipo de topónimo, área geográfica, etc.; y lo tercera es una aplicación que expone las principales estadísticas poblacionales de nuestro país, a partir del censo del 2002 y brinda iguales posibilidades que las antes expuestas.



Los principales resultados obtenidos en este trabajo pueden resumirse de la manera siguiente:

- Se logró la interpretación teórica de las especificaciones del Consorcio Internacional OpenGIS referente a los Filtros, Lenguaje de Marcas Geográficas, Esquemas y Servidor de Objetos Geográfico, así como sus implementaciones usando el lenguaje de programación Delphi en su versión 7.
- Se desarrolló un Servidor de Objetos geográficos basado en las especificaciones del Consorcio Internacional OpenGIS, que implementa las operaciones GetCapabilities, DescribeFeatureType, GetFeature y Transaction y que da soporte al conjunto de requerimientos de las aplicaciones SIG en Web.
- Se obtuvo una aplicación útil para nuestro país, la cual permite recuperar objetos geográficos y sus características solicitados a través de un Web Feature Server accediendo a los datos primarios, y devolver la petición en el lenguaje extendido de XML, Geographic Markup Language (GML) version 2.1.
- Teniendo en cuenta el punto anterior y que el WFS es un servicio que soporta consultas descriptivas y espaciales, se mejora considerablemente la transmisión de la Información Geográfica y el acceso a los mismos.
- El servicio desarrollado cumple con los requisitos de seguridad y protección de la información que son requeridos para la manipulación de la misma.



4.3 Alcances y limitaciones.

Alcances

1. La aplicación está disponible vía Web, para que cualquier usuario con un equipo conectado a Internet y con un navegador habilitado pueda usar la aplicación.
2. La tecnología lograda facilita la implementación de SIG en Web, garantizando el acceso a los objetos geográficos primarios de forma rápida y con el uso de navegadores comunes.
3. Con el servidor se logra la realización de consultas temáticas y espaciales mostrando los resultados en formato texto (GML).
4. Se facilita la creación de mapas temáticos al proporcionar accesos más rápidos a los atributos de los objetos geográficos.
5. Como el servidor de objetos es compatible con las especificaciones de OpenGIS, sus resultados pueden ser interpretados, y visualizados en el caso de que sean objetos geográficos, por los software profesionales que interactúan con este tipo de servidores como por ejemplo Mapinfo y Geomedia.
6. Se integrará a los resultados obtenidos por nuestro Grupo de Desarrollo de Software del Grupo Empresarial GEOCUBA, como una solución para la consulta de Objetos Geográficos de la cartografía del Grupo Empresarial.

Limitaciones

- a. El servidor implementado es compatible con Windows (NT, Server, 2000 y XP) y sobre *Internet Information Service*.
- b. Interpreta información solo en formato vectorial.



- c. El acceso a la información no toma en consideración las facilidades de almacenamiento de objetos geográficos que poseen algunos gestores de bases de datos en la actualidad.
- d. La cartografía que se consulta, está en la Base de Información Geográfica propia del Grupo Empresarial GEOCUBA, por lo tanto, nuestra aplicación dependerá de la disponibilidad de ésta para funcionar adecuadamente.

4.4 Trabajo Futuro.

Recientemente ha surgido un grupo de trabajo en el OGC cuyo objetivo es analizar las tendencias de software en los entornos de los servicios Web, para decidir las directrices futuras de las especificaciones de los servicios del OGC para proporcionar la interoperabilidad. Este grupo de trabajo ha recibido el nombre de OWS (OpenGIS Web Services), se está apoyando en la norma ISO-19119 para los SIG y a este proceso se conoce como la nueva iniciativa en el OGC para avanzar con la formalización de los servicios Web. Se trata de OpenGIS Web Service 1.2 que está sentando las bases para una futura actuación en OpenGIS Web Service 2. Las características interesantes, desde el punto de vista de los servicios Web, residen principalmente en el modelo de servicio orientado a la interoperabilidad entre programas (IPSM). Este nuevo modelo se adapta a las tecnologías emergentes y se está convirtiendo en el núcleo a través del cual OpenGIS Web Service continuarán creciendo.

Las perspectivas futuras del grupo residen en actualizar todos los servicios en conformidad con la nueva iniciativa, así como la implementación de otros que emane la misma.

4.5 Conclusiones.

1. Se desarrolló un trabajo de investigación teórica fundamental para la creación de un Servidor de Rasgos Geográficos y su publicación en formato GML de acuerdo a las tendencias actuales de desarrollo de los SIG en Web y las características específicas de Cuba.



2. Se obtuvo un WFS transaccional acorde con las exigencias del mercado internacional cumpliendo con las especificaciones OpenGIS.
3. Se logró el acceso a los datos primarios ubicados en las Bases de Información Geográfica.
4. Se mejoró considerablemente la transmisión de la información, en comparación con el caso imagen del servidor de mapas implementado.
5. Se cumple con los requisitos de seguridad y protección para cada nivel informativo, según las exigencias de los organismos rectores de esta actividad en nuestro país.
6. Se incluyó en el WFS la factibilidad técnica de acceder a la información ubicada en diferentes almacenes de datos.
7. Se desarrollaron factibilidades técnicas para presentar los datos al cliente en una aplicación visual, útil y significativa como son los servidores de mapas basados en las especificaciones de OpenGIS.



- [1] OpenGIS Consortium (OGC).(www.opengis.org).
- [2] Geoportal Cubano. (www.iderc.cu).
- [3] ISO/TC 211 Geographic information/Geomatics, 1999. (www.isotc211.org).
- [4] www.ccrs.nrcan.gc.ca/ccrs/homepg.pl?e.
- [5] www.canri.nsw.gov.au.
- [6] Open GIS Consortium, Inc., The OpenGIS Abstract Specification, Versión 4, 1999 (www.opengis.org/techno/implementation.htm).
- [7] Open GIS Consortium, Inc., OpenGIS Web Map Server Interface Implementation Specification Revision 1.0.0, 2000 (www.opengis.org/techno/implementation.htm).
- [8] OpenGIS Consortium, Inc., OpenGIS Simple Features Specification For OLE/COM Revision 1.1, 1999 (www.opengis.org/techno/implementation.htm).
- [9] OpenGIS Consortium, Inc., OpenGIS Catalog Interface Implementation Revision 1.1, 1999 (www.opengis.org/techno/implementation.htm).
- [10] Open GIS Consortium, 1997. The OGC Technical Committee Technology Development Process, Wayland, Massachusetts. (www.opengis.org/techno/development.htm).
- [11] Burrough, P.A. Principles of Geographical Information Systems for Land Resources Assessment. Oxford University Press, New York, U.S.A., 1990.
- [12] The GIS History Project, David M. Mark, Nicholas Chrisman, Andrew U. Frank, Patrick H. McHaffie, John Pickles.(www.usgs.gov/research/gis/title.html 06-27-97).
- [13] [Bra90].
- [14] [NCG90].
- [15] Max J. Egenhofer, Interaction with Geographic Information Systems via Spatial Queries, National Center for Geographic Information and Analysis and



- Department of Surveying Engineering, University of Maine, Orono, U.S.A.
Journal of Visual Languages and Computing (1990).
- [16] U.S. Geological Survey. (www.usgs.gov/research/gis/title.html) 07-01-97.
 - [17] Henry Tom, National Institute of Standards on Technology, Standard View Vo l2. No. 3 September 1994.
 - [18] Niki Pissinou, Kia Makki, E.K. Park, Toward the Design and Development of a New Architecture for Geographic Information System, Center for Advanced Computer Studies University of Southwestern Louisiana, Dept. of Computer Science University of Nevada, Computer Science Dept. United States Naval Academy, Communication of the ACM 1993.
 - [19] Ramesh Subramanian, Nabil R. Adam, The Design and Implementation of an Expert Object-Oriented Geographic Information System, University of Alaska, Rutgers University, Communication of the ACM 1993.
 - [20] Max J. Egenhofer, Geographic Database Systems: Issues and Research Needs, Department of Surveying Engineering Department of Spatial Information Science and Engineering Department of Computer Science, University of Maine, 1996 ACM.
 - [21] Bernard J. Jansen, Failure Analysis in Query Construction: Data and Análisis from A Large Sample of Web Queries, Department of Electrical Engineering & Computer Science United States Military Academy, 1998 ACM.
 - [22] Gísli R. Hjaltason, Distance Browsing in Spatial Databases, University of Maryland, 1999 ACM.
 - [23] Plewe, B., GIS Online, Onword Press, 1997.
 - [24] Dr. Holger Theisel. Scientific Visualization. Compact course. Febr / March 2000.



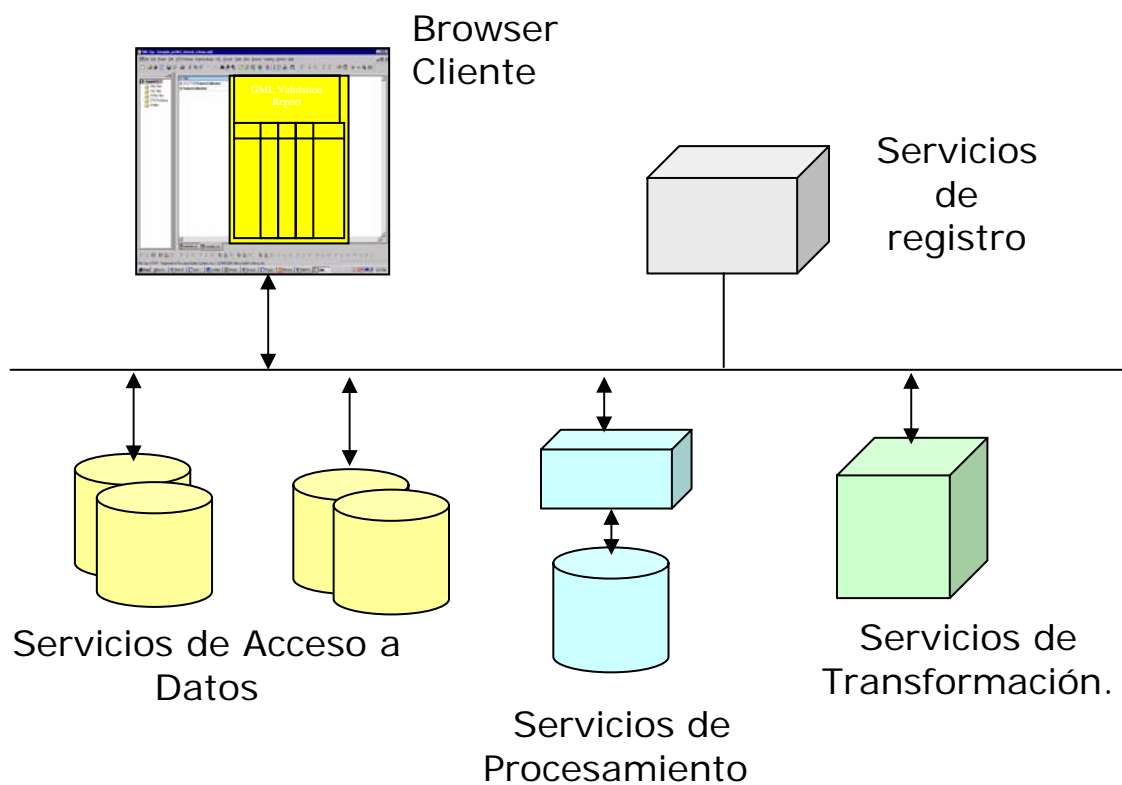
1. Open GIS Consortium Specifications, 1998 – 1999 – 2000.
2. Open GIS Consortium, Inc., The OpenGIS Abstract Specification, Versión 4, 1999.
3. OpenGIS Consortium, Inc., OpenGIS Simple Features Specification For OLE/COM Revision 1.1, 1999.
4. OpenGIS Consortium, Inc., OpenGIS Simple Features Specification For SQL Revision 1.1, 1999.
5. Open GIS Consortium, Inc., OpenGIS Web Map Server Interface Implementation Specification Revision 1.0.0, 2000.
6. OpenGIS Consortium, Inc., OpenGIS Geography Markup Language (GML) Implementation V 3_0. 2003.
7. OpenGIS Consortium, Inc., OpenGIS Web Feature Service Implementation V 1_0_0. 2002.
8. OpenGIS Consortium, Inc., Filter Encoding Implementation Specification. V 1.0. 2001.
9. OpenGIS Consortium, Inc., Styled Layer Descriptor specification (V 1.0).
10. OpenGIS Consortium, Inc., Catalog Services Specification. V 1.0.
11. OpenGIS Consortium, Inc., WMS Cookbook.
12. Microsoft Developer Network, Julio-2002-2003.
13. Borland Delphi 7.0. Programmer Guide.
14. Programación en Visual C++.
15. Inside Windows 2000, Third Edition.
16. GSDI, The SDI CookBook, Version 1, 2000.
17. ADOBE SYSTEMS INCORPORATED (2000): SVG, Scalable Vector Graphics, Release notes, <http://www.adobe.com/svg/indepth/releasenotes.html> (2000.06.25).



18. Web Services XML-RPC, SOAP, PHP y otros conceptos. Bankhacker.com
<http://web-services.bankhacker.com/>
19. Aplicaciones WEB usando XML. Web Studios E-Solutions
http://www.ebstudio.com/src/client/scripts/informacion.php?cat_id=42
20. Creating Internet-Based GIS Applications,
URL:<http://www.ce.utexas.edu/stu/favazzds/GISPaper.html>, 1998
21. Internet für Kartographen; in: Kartographische Nachrichten, 1996, Jg. 46, Heft 5, S. 185-190.
22. Internet-Kartographie: (R)Evolution oder Sackgasse, in: Kartographische Nachrichten, 1999, Jg. 49, Heft 3, S. 98-104.
23. Geographic Information Systems and Cartographic Modelling, Prentice-Hall, Englewood Cliffs, New Jersey.1990.
24. Introduction to GML(Geography Markup Language)
URL:<http://www.opengis.org/GML.html>



Anexo A: Arquitectura de Servicios de OGC.





Anexo B. XML de definición del esquema de FILTER.

B.1 Introducción

Los esquemas contenidos en esta sección son las definiciones de esquemas XML para las expresiones, filtros y capacidades del filtro. El esquema filter.xsd puede ser importado desde cualquier esquema que intente usar la codificación del filtro descrito en esta especificación. El esquema, filterCapabilities.xsd debe importarse en cualquier esquema de capacidades que intente anunciar las capacidades de un módulo particular de proceso de filtrado.

B.2 expr.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema targetNamespace="http://www.opengis.net/ogc"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xsd:element name="Add" type="ogc:BinaryOperatorType"
    substitutionGroup="ogc:expression" />
  <xsd:element name="Sub" type="ogc:BinaryOperatorType"
    substitutionGroup="ogc:expression" />
  <xsd:element name="Mul" type="ogc:BinaryOperatorType"
    substitutionGroup="ogc:expression" />
  <xsd:element name="Div" type="ogc:BinaryOperatorType"
    substitutionGroup="ogc:expression" />
  <xsd:element name="PropertyName" type="ogc:PropertyNameType"
    substitutionGroup="ogc:expression" />
  <xsd:element name="Function" type="ogc:FunctionType"
    substitutionGroup="ogc:expression" />
  <xsd:element name="Literal" type="ogc:LiteralType"
    substitutionGroup="ogc:expression" />
  <xsd:element name="expression" type="ogc:ExpressionType" abstract="true" />
  <xsd:complexType name="ExpressionType" abstract="true" />
  <xsd:complexType name="BinaryOperatorType">
  <xsd:complexContent>
  <xsd:extension base="ogc:ExpressionType">
  <xsd:sequence>
    <xsd:element ref="ogc:expression" minOccurs="2" maxOccurs="2" />
  </xsd:sequence>
  </xsd:extension>
  </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="FunctionType">
  <xsd:complexContent>
  <xsd:extension base="ogc:ExpressionType">
```



```

<xsd:sequence>
  <xsd:element ref="ogc:expression" minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="name" type="xsd:string" use="required" />
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="LiteralType">
<xsd:complexContent mixed="true">
<xsd:extension base="ogc:ExpressionType">
<xsd:sequence>
  <xsd:any minOccurs="0" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PropertyNameType">
<xsd:complexContent mixed="true">
<xsd:extension base="ogc:ExpressionType" />
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

B.3 filter.xsd

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema targetNamespace="http://www.opengis.net/ogc"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xsd:include schemaLocation="expr.xsd" />
  <xsd:import namespace="http://www.opengis.net/gml"
    schemaLocation="geometry.xsd" />
  <!-- ===== -->
  FILTER EXPRESSION
  <!-- ===== -->
  <xsd:element name="FeatureId" type="ogc:FeatureIdType" />
  <xsd:element name="Filter" type="ogc:FilterType" />
  <!-- ===== -->
  COMPARISON OPERATORS
  <!-- ===== -->
  <xsd:element name="comparisonOps" type="ogc:ComparisonOpsType"
    abstract="true" />
  <xsd:element name="PropertyIsEqualTo" type="ogc:BinaryComparisonOpType"
    substitutionGroup="ogc:comparisonOps" />

```



```

<xsd:element name="PropertyIsNotEqualTo"
  type="ogc:BinaryComparisonOpType"
  substitutionGroup="ogc:comparisonOps" />
<xsd:element name="PropertyIsLessThan"
  type="ogc:BinaryComparisonOpType"
  substitutionGroup="ogc:comparisonOps" />
<xsd:element name="PropertyIsGreaterThan"
  type="ogc:BinaryComparisonOpType"
  substitutionGroup="ogc:comparisonOps" />
<xsd:element name="PropertyIsLessThanOrEqualTo"
  type="ogc:BinaryComparisonOpType"
  substitutionGroup="ogc:comparisonOps" />
<xsd:element name="PropertyIsGreaterThanOrEqualTo"
  type="ogc:BinaryComparisonOpType"
  substitutionGroup="ogc:comparisonOps" />
<xsd:element name="PropertyIsLike" type="ogc:PropertyIsLikeType"
  substitutionGroup="ogc:comparisonOps" />
<xsd:element name="PropertyIsNull" type="ogc:PropertyIsNullType"
  substitutionGroup="ogc:comparisonOps" />
<xsd:element name="PropertyIsBetween" type="ogc:PropertyIsBetweenType"
  substitutionGroup="ogc:comparisonOps" />
<xsd:complexType name="ComparisonOpsType" abstract="true" />
<!-- ===== -->
  SPATIAL OPERATORS (sec 3.2.19.2 99-049)
<!-- ===== -->
<xsd:element name="spatialOps" type="ogc:SpatialOpsType" abstract="true" />
<xsd:element name="Equals" type="ogc:BinarySpatialOpType"
  substitutionGroup="ogc:spatialOps" />
<xsd:element name="Disjoint" type="ogc:BinarySpatialOpType"
  substitutionGroup="ogc:spatialOps" />
<xsd:element name="Touches" type="ogc:BinarySpatialOpType"
  substitutionGroup="ogc:spatialOps" />
<xsd:element name="Within" type="ogc:BinarySpatialOpType"
  substitutionGroup="ogc:spatialOps" />
<xsd:element name="Overlaps" type="ogc:BinarySpatialOpType"
  substitutionGroup="ogc:spatialOps" />
<xsd:element name="Crosses" type="ogc:BinarySpatialOpType"
  substitutionGroup="ogc:spatialOps" />
<xsd:element name="Intersects" type="ogc:BinarySpatialOpType"
  substitutionGroup="ogc:spatialOps" />
<xsd:element name="Contains" type="ogc:BinarySpatialOpType"
  substitutionGroup="ogc:spatialOps" />
<!-- ===== -->
<xsd:element name="DWithin" type="ogc:DistanceBufferType"
  substitutionGroup="ogc:spatialOps" />
<xsd:element name="Beyond" type="ogc:DistanceBufferType"
  substitutionGroup="ogc:spatialOps" />

```



```

<!-- ===== -->
<xsd:element name="BBOX" type="ogc:BBOXType"
  substitutionGroup="ogc:spatialOps" />
<xsd:complexType name="SpatialOpsType" abstract="true" />
<!-- ===== -->
LOGICAL OPERATORS
===== -->
<xsd:element name="logicOps" type="ogc:LogicOpsType" abstract="true" />
<xsd:element name="And" type="ogc:BinaryLogicOpType"
  substitutionGroup="ogc:logicOps" />
<xsd:element name="Or" type="ogc:BinaryLogicOpType"
  substitutionGroup="ogc:logicOps" />
<xsd:element name="Not" type="ogc:UnaryLogicOpType"
  substitutionGroup="ogc:logicOps" />
<xsd:complexType name="LogicOpsType" abstract="true" />
<!-- ===== -->
COMPLEX TYPES
===== -->
<xsd:complexType name="FilterType">
  <xsd:choice>
    <xsd:element ref="ogc:spatialOps" />
    <xsd:element ref="ogc:comparisonOps" />
    <xsd:element ref="ogc:logicOps" />
    <xsd:element ref="ogc:FeatureId" maxOccurs="unbounded" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="FeatureIdType">
  <xsd:attribute name="fid" type="xsd:anyURI" use="required" />
</xsd:complexType>
<xsd:complexType name="BinaryComparisonOpType">
  <xsd:complexContent>
    <xsd:extension base="ogc:ComparisonOpsType">
      <xsd:sequence>
        <xsd:element ref="ogc:expression" minOccurs="2" maxOccurs="2" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PropertyIsLikeType">
  <xsd:complexContent>
    <xsd:extension base="ogc:ComparisonOpsType">
      <xsd:sequence>
        <xsd:element ref="ogc:PropertyName" />
        <xsd:element ref="ogc:Literal" />
      </xsd:sequence>
      <xsd:attribute name="wildCard" type="xsd:string" use="required" />
      <xsd:attribute name="singleChar" type="xsd:string" use="required" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```



```

<xsd:attribute name="escape" type="xsd:string" use="required" />
  </xsd:extension>
  </xsd:complexContent>
  </xsd:complexType>
<xsd:complexType name="PropertyIsNullType">
  <xsd:complexContent>
  <xsd:extension base="ogc:ComparisonOpsType">
  <xsd:choice>
    <xsd:element ref="ogc:PropertyName" />
    <xsd:element ref="ogc:Literal" />
  </xsd:choice>
  </xsd:extension>
  </xsd:complexContent>
  </xsd:complexType>
<xsd:complexType name="PropertyIsBetweenType">
  <xsd:complexContent>
  <xsd:extension base="ogc:ComparisonOpsType">
  <xsd:sequence>
    <xsd:element ref="ogc:expression" />
    <xsd:element name="LowerBoundary" type="ogc:LowerBoundaryType" />
    <xsd:element name="UpperBoundary" type="ogc:UpperBoundaryType" />
  </xsd:sequence>
  </xsd:extension>
  </xsd:complexContent>
  </xsd:complexType>
<xsd:complexType name="LowerBoundaryType">
  <xsd:choice>
    <xsd:element ref="ogc:expression" />
  </xsd:choice>
  </xsd:complexType>
<xsd:complexType name="UpperBoundaryType">
  <xsd:sequence>
    <xsd:element ref="ogc:expression" />
  </xsd:sequence>
  </xsd:complexType>
<xsd:complexType name="BinarySpatialOpType">
  <xsd:complexContent>
  <xsd:extension base="ogc:SpatialOpsType">
  <xsd:sequence>
    <xsd:element ref="ogc:PropertyName" />
  </xsd:sequence>
  </xsd:extension>
  </xsd:complexContent>
  </xsd:complexType>
  <xsd:choice>
    <xsd:element ref="gml:_Geometry" />
    <xsd:element ref="gml:Box" />
  </xsd:choice>
  </xsd:sequence>
  </xsd:extension>
  </xsd:complexContent>

```



```

    </xsd:complexType>
<xsd:complexType name="BBOXType">
<xsd:complexContent>
<xsd:extension base="ogc:SpatialOpsType">
<xsd:sequence>
  <xsd:element ref="ogc:PropertyName" />
  <xsd:element ref="gml:Box" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="DistanceBufferType">
<xsd:complexContent>
<xsd:extension base="ogc:SpatialOpsType">
<xsd:sequence>
  <xsd:element ref="ogc:PropertyName" />
  <xsd:element ref="gml:_Geometry" />
  <xsd:element name="Distance" type="ogc:DistanceType" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="DistanceType" mixed="true">
  <xsd:attribute name="units" type="xsd:string" use="required" />
</xsd:complexType>
<xsd:complexType name="BinaryLogicOpType">
<xsd:complexContent>
<xsd:extension base="ogc:LogicOpsType">
<xsd:choice minOccurs="2" maxOccurs="unbounded">
  <xsd:element ref="ogc:comparisonOps" />
  <xsd:element ref="ogc:spatialOps" />
  <xsd:element ref="ogc:logicOps" />
</xsd:choice>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="UnaryLogicOpType">
<xsd:complexContent>
<xsd:extension base="ogc:LogicOpsType">
<xsd:sequence>
<xsd:choice>
  <xsd:element ref="ogc:comparisonOps" />
  <xsd:element ref="ogc:spatialOps" />
  <xsd:element ref="ogc:logicOps" />
</xsd:choice>
</xsd:sequence>
</xsd:extension>

```




```

</xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

B.4 filterCapabilities.xsd

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema targetNamespace="http://www.opengis.net/ogc"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ogc="http://www.opengis.net/ogc" elementFormDefault="qualified">
  <xsd:complexType name="Arithmetic_OperatorsType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="ogc:Simple_Arithmetic" />
      <xsd:element name="Functions" type="ogc:FunctionsType" />
    </xsd:choice>
  </xsd:complexType>
  <xsd:element name="BBOX">
    <xsd:complexType />
  </xsd:element>
  <xsd:element name="Between">
    <xsd:complexType />
  </xsd:element>
  <xsd:element name="Beyond">
    <xsd:complexType />
  </xsd:element>
  <xsd:complexType name="Comparison_OperatorsType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="ogc:Simple_Comparisons" />
      <xsd:element ref="ogc:Like" />
      <xsd:element ref="ogc:Between" />
      <xsd:element ref="ogc:NullCheck" />
    </xsd:choice>
  </xsd:complexType>
  <xsd:element name="Contains">
    <xsd:complexType />
  </xsd:element>
  <xsd:element name="Crosses">
    <xsd:complexType />
  </xsd:element>
  <xsd:element name="Disjoint">
    <xsd:complexType />
  </xsd:element>
  <xsd:element name="DWithin">
    <xsd:complexType />
  </xsd:element>
  <xsd:element name="Equals">
    <xsd:complexType />
  </xsd:element>

```



```

    </xsd:element>
<xsd:element name="Filter_Capabilities">
<xsd:complexType>
<xsd:sequence>
  <xsd:element name="Spatial_Capabilities" type="ogc:Spatial_CapabilitiesType"
    />
  <xsd:element name="Scalar_Capabilities" type="ogc:Scalar_CapabilitiesType" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="Function_NameType">
<xsd:simpleContent>
<xsd:extension base="xsd:string">
  <xsd:attribute name="nArgs" type="xsd:string" use="required" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Function_NamesType">
<xsd:sequence maxOccurs="unbounded">
  <xsd:element name="Function_Name" type="ogc:Function_NameType" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FunctionsType">
<xsd:sequence>
  <xsd:element name="Function_Names" type="ogc:Function_NamesType" />
</xsd:sequence>
</xsd:complexType>
<xsd:element name="Intersect">
  <xsd:complexType />
</xsd:element>
<xsd:element name="Like">
  <xsd:complexType />
</xsd:element>
<xsd:element name="Logical_Operators">
  <xsd:complexType />
</xsd:element>
<xsd:element name="NullCheck">
  <xsd:complexType />
</xsd:element>
<xsd:element name="Overlaps">
  <xsd:complexType />
</xsd:element>
<xsd:complexType name="Scalar_CapabilitiesType">
<xsd:choice maxOccurs="unbounded">
  <xsd:element ref="ogc:Logical_Operators" />
  <xsd:element name="Comparison_Operators"
    type="ogc:Comparison_OperatorsType" />

```




```

<xsd:element name="Arithmetic_Operators"
  type="ogc:Arithmetic_OperatorsType" />
</xsd:choice>
</xsd:complexType>
<xsd:element name="Simple_Arithmetic">
  <xsd:complexType />
</xsd:element>
<xsd:element name="Simple_Comparisons">
  <xsd:complexType />
</xsd:element>
<xsd:complexType name="Spatial_CapabilitiesType">
<xsd:sequence>
  <xsd:element name="Spatial_Operators" type="ogc:Spatial_OperatorsType" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Spatial_OperatorsType">
<xsd:choice maxOccurs="unbounded">
  <xsd:element ref="ogc:BBOX" />
  <xsd:element ref="ogc:Equals" />
  <xsd:element ref="ogc:Disjoint" />
  <xsd:element ref="ogc:Intersect" />
  <xsd:element ref="ogc:Touches" />
  <xsd:element ref="ogc:Crosses" />
  <xsd:element ref="ogc:Within" />
  <xsd:element ref="ogc:Contains" />
  <xsd:element ref="ogc:Overlaps" />
  <xsd:element ref="ogc:Beyond" />
  <xsd:element ref="ogc:DWithin" />
</xsd:choice>
</xsd:complexType>
<xsd:element name="Touches">
  <xsd:complexType />
</xsd:element>
<xsd:element name="Within">
  <xsd:complexType />
</xsd:element>
</xsd:schema>

```



Anexo C. XML de definición del esquema de GML.

C.1 Geometry.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<schema targetNamespace="http://www.opengis.net/gml"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified"
  version="2.06">
<annotation>
  <appinfo>geometry.xsd v2.06 2001-02</appinfo>
  <documentation xml:lang="en">GML Geometry schema. Copyright (c) 2001 OGC,
    All Rights Reserved.</documentation>
</annotation>
<!-- bring in the XLink attributes -->
<import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlinks.xsd"
  />
  <!-- global declarations -->
  <element name="_Geometry" type="gml:AbstractGeometryType" abstract="true" />
  <element name="_GeometryCollection" type="gml:GeometryCollectionType"
    abstract="true" />
  <element name="geometryMember" type="gml:GeometryAssociationType" />
  <!-- primitive geometry elements -->
  <element name="Point" type="gml:PointType" substitutionGroup="gml:_Geometry"
    />
  <element name="LineString" type="gml:LineStringType"
    substitutionGroup="gml:_Geometry" />
  <element name="LinearRing" type="gml:LinearRingType"
    substitutionGroup="gml:_Geometry" />
  <element name="Polygon" type="gml:PolygonType"
    substitutionGroup="gml:_Geometry" />
  <element name="Box" type="gml:BoxType" />
  <!-- aggregate geometry elements -->
  <element name="MultiGeometry" type="gml:GeometryCollectionType" />
  <element name="MultiPoint" type="gml:MultiPointType"
    substitutionGroup="gml:_Geometry" />
  <element name="MultiLineString" type="gml:MultiLineStringType"
    substitutionGroup="gml:_Geometry" />
  <element name="MultiPolygon" type="gml:MultiPolygonType"
    substitutionGroup="gml:_Geometry" />
  <!-- coordinate elements -->
  <element name="coord" type="gml:CoordType" />
  <element name="coordinates" type="gml:CoordinatesType" />
  <!-- this attribute gives the location where an element is defined -->
  <attribute name="remoteSchema" type="anyURI" />
  <!-- abstract supertypes -->
  <complexType name="AbstractGeometryType" abstract="true">
```



```

<annotation>
  <documentation>All geometry elements are derived from this abstract supertype;
    a geometry element may have an identifying attribute ('gid'). It may be
    associated with a spatial reference system.</documentation>
</annotation>
<attribute name="gid" type="ID" use="optional" />
<attribute name="srsName" type="anyURI" use="optional" />
</complexType>
<complexType name="AbstractGeometryCollectionBaseType" abstract="true">
<annotation>
  <documentation>This abstract base type for geometry collections just makes the
    srsName attribute mandatory.</documentation>
</annotation>
<complexContent>
<restriction base="gml:AbstractGeometryType">
  <attribute name="gid" type="ID" use="optional" />
  <attribute name="srsName" type="anyURI" use="required" />
</restriction>
</complexContent>
</complexType>
<attributeGroup name="AssociationAttributeGroup">
<annotation>
  <documentation>These attributes can be attached to any element, thus allowing it
    to act as a pointer. The 'remoteSchema' attribute allows an element that
    carries link attributes to indicate that the element is declared in a remote
    schema rather than by the schema that constrains the current document
    instance.</documentation>
</annotation>
<attributeGroup ref="xlink:simpleLink" />
<attribute ref="gml:remoteSchema" use="optional" />
</attributeGroup>
<complexType name="GeometryAssociationType">
<annotation>
  <documentation>An instance of this type (e.g. a geometryMember) can either
    enclose or point to a primitive geometry element. When serving as a simple
    link that references a remote geometry instance, the value of the
    gml:remoteSchema attribute can be used to locate a schema fragment that
    constrains the target instance.</documentation>
</annotation>
<sequence>
  <element ref="gml:_Geometry" minOccurs="0" />
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
<!-- primitive geometry types -->
<complexType name="PointType">
<annotation>

```



```

<documentation>A Point is defined by a single coordinate tuple.</documentation>
  </annotation>
<complexContent>
<extension base="gml:AbstractGeometryType">
<sequence>
<choice>
  <element ref="gml:coord" />
  <element ref="gml:coordinates" />
  </choice>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="LineStringType">
<annotation>
  <documentation>A LineString is defined by two or more coordinate tuples, with
    linear interpolation between them.</documentation>
  </annotation>
<complexContent>
<extension base="gml:AbstractGeometryType">
<sequence>
<choice>
  <element ref="gml:coord" minOccurs="2" maxOccurs="unbounded" />
  <element ref="gml:coordinates" />
  </choice>
</sequence>
</extension>
</complexContent>
</complexType>
<annotation>
  <documentation>A LinearRing is defined by four or more coordinate tuples, with
    linear interpolation between them; the first and last coordinates must be
    coincident.</documentation>
  </annotation>
<complexContent>
<extension base="gml:AbstractGeometryType">
<sequence>
<choice>
  <element ref="gml:coord" minOccurs="4" maxOccurs="unbounded" />
  <element ref="gml:coordinates" />
  </choice>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="BoxType">
<annotation>

```



```

<documentation>The Box structure defines an extent using a pair of coordinate
  tuples.</documentation>
</annotation>
<complexContent>
<extension base="gml:AbstractGeometryType">
<sequence>
<choice>
  <element ref="gml:coord" minOccurs="2" maxOccurs="2" />
  <element ref="gml:coordinates" />
</choice>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="PolygonType">
<annotation>
  <documentation>A Polygon is defined by an outer boundary and zero or more
    inner boundaries which are in turn defined by LinearRings.</documentation>
</annotation>
<complexContent>
<extension base="gml:AbstractGeometryType">
<sequence>
<element name="outerBoundaryIs">
<complexType>
<sequence>
  <element ref="gml:LinearRing" />
</sequence>
</complexType>
</element>
<element name="innerBoundaryIs" minOccurs="0" maxOccurs="unbounded">
<complexType>
<sequence>
  <element ref="gml:LinearRing" />
</sequence>
</complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
  <!-- aggregate geometry types -->
<complexType name="GeometryCollectionType">
<annotation>
  <documentation>A geometry collection must include one or more geometries,
    referenced through geometryMember elements. User-defined geometry
    collections that accept GML geometry classes as members must instantiate--
    or derive from--this type.</documentation>

```



```

    </annotation>
<complexContent>
<extension base="gml:AbstractGeometryCollectionBaseType">
<sequence>
    <element ref="gml:geometryMember" maxOccurs="unbounded" />
  </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="MultiPointType">
<annotation>
    <documentation>A MultiPoint is defined by one or more Points, referenced
      through pointMember elements.</documentation>
</annotation>
<complexContent>
<restriction base="gml:GeometryCollectionType">
<sequence>
    <element name="pointMember" maxOccurs="unbounded">
<complexType>
<sequence>
    <element ref="gml:Point" />
  </sequence>
</complexType>
</element>
</sequence>
</restriction>
</complexContent>
</complexType>
<complexType name="MultiLineStringType">
<annotation>
    <documentation>A MultiLineString is defined by one or more LineStrings,
      referenced through lineStringMember elements.</documentation>
</annotation>
<complexContent>
<restriction base="gml:GeometryCollectionType">
<sequence>
    <element name="lineStringMember" maxOccurs="unbounded">
<complexType>
<sequence>
    <element ref="gml:LineString" />
  </sequence>
</complexType>
</element>
</sequence>
</restriction>
</complexContent>
</complexType>

```



```

<complexType name="MultiPolygonType">
<annotation>
  <documentation>A MultiPolygon is defined by one or more Polygons, referenced
    through polygonMember elements.</documentation>
</annotation>
<complexContent>
<restriction base="gml:GeometryCollectionType">
<sequence>
<element name="polygonMember" maxOccurs="unbounded">
<complexType>
<sequence>
  <element ref="gml:Polygon" />
</sequence>
</complexType>
</element>
</sequence>
</restriction>
</complexContent>
</complexType>
<complexType name="CoordType">
<annotation>
  <documentation>Represents a coordinate tuple in one, two, or three
    dimensions.</documentation>
</annotation>
<sequence>
  <element name="X" type="decimal" />
  <element name="Y" type="decimal" minOccurs="0" />
  <element name="Z" type="decimal" minOccurs="0" />
</sequence>
</complexType>
<complexType name="CoordinatesType">
<annotation>
  <documentation>Coordinates can be included in a single string, but there is no
    facility for validating string content. The value of the 'cs' attribute is the
    separator for coordinate values, and the value of the 'ts' attribute gives the
    tuple separator (a single space by default); the default values may be
    changed to reflect local usage.</documentation>
</annotation>
<simpleContent>
<extension base="string">
  <attribute name="decimal" type="string" default="." />
  <attribute name="cs" type="string" default="," />
  <attribute name="ts" type="string" default="" />
</extension>
</simpleContent>
</complexType>
</schema>

```




C.2 Feature.xsd

```

<?xml version="1.0" encoding="UTF-8" ?>
<schema targetNamespace="http://www.opengis.net/gml"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  version="2.06">
<annotation>
  <appinfo>feature.xsd v2.06 2001-02</appinfo>
  <documentation xml:lang="en">GML Feature schema. Copyright (c) 2001 OGC, All
    Rights Reserved.</documentation>
</annotation>
<!-- include constructs from the GML Geometry schema -->
<include schemaLocation="geometry.xsd" />
  <!-- global declarations -->
  <element name="_Feature" type="gml:AbstractFeatureType" abstract="true" />
  <element name="_FeatureCollection" type="gml:AbstractFeatureCollectionType"
    abstract="true" substitutionGroup="gml:_Feature" />
  <element name="featureMember" type="gml:FeatureAssociationType" />
  <!-- some basic geometric properties of features -->
  <element name="_geometryProperty" type="gml:GeometryPropertyType"
    abstract="true" />
  <element name="geometryProperty" type="gml:GeometryPropertyType" />
  <element name="boundedBy" type="gml:BoundingShapeType" />
  <element name="pointProperty" type="gml:PointPropertyType"
    substitutionGroup="gml:_geometryProperty" />
  <element name="polygonProperty" type="gml:PolygonPropertyType"
    substitutionGroup="gml:_geometryProperty" />
  <element name="lineStringProperty" type="gml:LineStringPropertyType"
    substitutionGroup="gml:_geometryProperty" />
  <element name="multiPointProperty" type="gml:MultiPointPropertyType"
    substitutionGroup="gml:_geometryProperty" />
  <element name="multiLineStringProperty"
    type="gml:MultiLineStringPropertyType"
    substitutionGroup="gml:_geometryProperty" />
  <element name="multiPolygonProperty" type="gml:MultiPolygonPropertyType"
    substitutionGroup="gml:_geometryProperty" />
  <element name="multiGeometryProperty" type="gml:MultiGeometryPropertyType"
    substitutionGroup="gml:_geometryProperty" />
  <!-- common aliases for geometry properties -->
  <element name="location" type="gml:PointPropertyType"
    substitutionGroup="gml:pointProperty" />
  <element name="centerOf" type="gml:PointPropertyType"
    substitutionGroup="gml:pointProperty" />
  <element name="position" type="gml:PointPropertyType"
    substitutionGroup="gml:pointProperty" />

```




```

<element name="extentOf" type="gml:PolygonPropertyType"
  substitutionGroup="gml:polygonProperty" />
<element name="coverage" type="gml:PolygonPropertyType"
  substitutionGroup="gml:polygonProperty" />
<element name="edgeOf" type="gml:LineStringPropertyType"
  substitutionGroup="gml:lineStringProperty" />
<element name="centerLineOf" type="gml:LineStringPropertyType"
  substitutionGroup="gml:lineStringProperty" />
<element name="multiLocation" type="gml:MultiPointPropertyType"
  substitutionGroup="gml:multiPointProperty" />
<element name="multiCenterOf" type="gml:MultiPointPropertyType"
  substitutionGroup="gml:multiPointProperty" />
<element name="multiPosition" type="gml:MultiPointPropertyType"
  substitutionGroup="gml:multiPointProperty" />
<element name="multiCenterLineOf" type="gml:MultiLineStringPropertyType"
  substitutionGroup="gml:multiLineStringProperty" />
<element name="multiEdgeOf" type="gml:MultiLineStringPropertyType"
  substitutionGroup="gml:multiLineStringProperty" />
<element name="multiCoverage" type="gml:MultiPolygonPropertyType"
  substitutionGroup="gml:multiPolygonProperty" />
<element name="multiExtentOf" type="gml:MultiPolygonPropertyType"
  substitutionGroup="gml:multiPolygonProperty" />
<!-- common feature descriptors -->
<element name="description" type="string" />
<element name="name" type="string" />
<!-- abstract supertypes -->
<complexType name="AbstractFeatureType" abstract="true">
<annotation>
  <documentation>An abstract feature provides a set of common properties. A
    concrete feature type must derive from this type and specify additional
    properties in an application schema. A feature may optionally possess an
    identifying attribute ('fid').</documentation>
</annotation>
<sequence>
  <element ref="gml:description" minOccurs="0" />
  <element ref="gml:name" minOccurs="0" />
  <element ref="gml:boundedBy" minOccurs="0" />
  <!-- additional properties must be specified in an application schema -->
</sequence>
  <attribute name="fid" type="ID" use="optional" />
</complexType>
<complexType name="AbstractFeatureCollectionBaseType" abstract="true">
<annotation>
  <documentation>This abstract base type just makes the boundedBy element
    mandatory for a feature collection.</documentation>
</annotation>
<complexContent>

```



```

<restriction base="gml:AbstractFeatureType">
<sequence>
  <element ref="gml:description" minOccurs="0" />
  <element ref="gml:name" minOccurs="0" />
  <element ref="gml:boundedBy" />
</sequence>
<attribute name="fid" type="ID" use="optional" />
</restriction>
</complexContent>
</complexType>
<complexType name="AbstractFeatureCollectionType" abstract="true">
<annotation>
  <documentation>A feature collection contains zero or more featureMember
    elements.</documentation>
</annotation>
<complexContent>
<extension base="gml:AbstractFeatureCollectionBaseType">
<sequence>
  <element ref="gml:featureMember" minOccurs="0" maxOccurs="unbounded" />
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="GeometryPropertyType">
<annotation>
  <documentation>A simple geometry property encapsulates a geometry element.
    Alternatively, it can function as a pointer (simple-type link) that refers to a
    remote geometry element.</documentation>
</annotation>
<sequence minOccurs="0">
  <element ref="gml:_Geometry" />
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
<complexType name="FeatureAssociationType">
<annotation>
  <documentation>An instance of this type (e.g. a featureMember) can either
    enclose or point to a feature (or feature collection); this type can be restricted
    in an application schema to allow only specified features as valid participants
    in the association. When serving as a simple link that references a remote
    feature instance, the value of the gml:remoteSchema attribute can be used to
    locate a schema fragment that constrains the target
    instance.</documentation>
</annotation>
<sequence minOccurs="0">
  <element ref="gml:_Feature" />
</sequence>

```



```

<attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
<complexType name="BoundingShapeType">
<annotation>
<documentation>Bounding shapes--a Box or a null element are currently
    allowed.</documentation>
</annotation>
<sequence>
<choice>
<element ref="gml:Box" />
<element name="null" type="gml:NullType" />
</choice>
</sequence>
</complexType>
<!-- geometry properties -->
<complexType name="PointPropertyType">
<annotation>
<documentation>Encapsulates a single point to represent position, location, or
    centerOf properties.</documentation>
</annotation>
<complexContent>
<restriction base="gml:GeometryPropertyType">
<sequence minOccurs="0">
<element ref="gml:Point" />
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup" />
</restriction>
</complexContent>
</complexType>
<complexType name="PolygonPropertyType">
<annotation>
<documentation>Encapsulates a single polygon to represent coverage or
    extentOf properties.</documentation>
</annotation>
<complexContent>
<restriction base="gml:GeometryPropertyType">
<sequence minOccurs="0">
<element ref="gml:Polygon" />
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup" />
</restriction>
</complexContent>
</complexType>
<complexType name="LineStringPropertyType">
<annotation>
<documentation>Encapsulates a single LineString to represent centerLineOf or
    edgeOf properties.</documentation>

```



```

    </annotation>
<complexContent>
<restriction base="gml:GeometryPropertyType">
<sequence minOccurs="0">
    <element ref="gml:LineString" />
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
    </restriction>
</complexContent>
</complexType>
<complexType name="MultiPointPropertyType">
<annotation>
    <documentation>Encapsulates a MultiPoint element to represent the following
        discontinuous geometric properties: multiLocation, multiPosition,
        multiCenterOf.</documentation>
    </annotation>
<complexContent>
<restriction base="gml:GeometryPropertyType">
<sequence minOccurs="0">
    <element ref="gml:MultiPoint" />
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
    </restriction>
</complexContent>
</complexType>
<complexType name="MultiLineStringPropertyType">
<annotation>
    <documentation>Encapsulates a MultiLineString element to represent the
        following discontinuous geometric properties: multiEdgeOf,
        multiCenterLineOf.</documentation>
    </annotation>
<complexContent>
<restriction base="gml:GeometryPropertyType">
<sequence minOccurs="0">
    <element ref="gml:MultiLineString" />
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
    </restriction>
</complexContent>
</complexType>
<complexType name="MultiPolygonPropertyType">
<annotation>
    <documentation>Encapsulates a MultiPolygon to represent the following
        discontinuous geometric properties: multiCoverage,
        multiExtentOf.</documentation>
    </annotation>
<complexContent>

```



```

<restriction base="gml:GeometryPropertyType">
<sequence minOccurs="0">
  <element ref="gml:MultiPolygon" />
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup" />
</restriction>
</complexContent>
</complexType>
<complexType name="MultiGeometryPropertyType">
<annotation>
  <documentation>Encapsulates a MultiGeometry element.</documentation>
</annotation>
<complexContent>
<restriction base="gml:GeometryPropertyType">
<sequence minOccurs="0">
  <element ref="gml:MultiGeometry" />
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup" />
</restriction>
</complexContent>
</complexType>
<simpleType name="NullType">
<annotation>
  <documentation>If a bounding shape is not provided for a feature collection,
    explain why. Allowable values are: innapplicable - the features do not have
    geometry unknown - the boundingBox cannot be computed unavailable -
    there may be a boundingBox but it is not divulged missing - there are no
    features</documentation>
</annotation>
<restriction base="string">
  <enumeration value="innapplicable" />
  <enumeration value="unknown" />
  <enumeration value="unavailable" />
  <enumeration value="missing" />
</restriction>
</simpleType>
</schema>
C.3 Xlinks.xsd
  <?xml version="1.0" encoding="UTF-8" ?>
  <schema targetNamespace="http://www.w3.org/1999/xlink"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns="http://www.w3.org/2001/XMLSchema" version="2.01">
  <annotation>
    <appinfo>xlinks.xsd v2.01 2001-02</appinfo>
    <documentation xml:lang="en">This schema provides the XLink attributes for
      general use.</documentation>
  </annotation>

```



```

    <!-- global declarations -->
    <!-- locator attribute -->
    <attribute name="href" type="anyURI" />
    <!-- semantic attributes -->
    <attribute name="role" type="anyURI" />
    <attribute name="arcrole" type="anyURI" />
    <attribute name="title" type="string" />
    <!-- behavior attributes -->
    <attribute name="show">
    <annotation>
    <documentation>The 'show' attribute is used to communicate the desired
    presentation of the ending resource on traversal from the starting resource;
    it's value should be treated as follows: new - load ending resource in a new
    window, frame, pane, or other presentation context replace - load the
    resource in the same window, frame, pane, or other presentation context
    embed - load ending resource in place of the presentation of the starting
    resource other - behavior is unconstrained; examine other markup in the link
    for hints none - behavior is unconstrained</documentation>
    </annotation>
    <simpleType>
    <restriction base="string">
    <enumeration value="new" />
    <enumeration value="replace" />
    <enumeration value="embed" />
    <enumeration value="other" />
    <enumeration value="none" />
    </restriction>
    </simpleType>
    </attribute>
    <attribute name="actuate">
    <annotation>
    <documentation>The 'actuate' attribute is used to communicate the desired
    timing of traversal from the starting resource to the ending resource; it's
    value should be treated as follows: onLoad - traverse to the ending resource
    immediately on loading the starting resource onRequest - traverse from the
    starting resource to the ending resource only on a post-loading event
    triggered for this purpose other - behavior is unconstrained; examine other
    markup in link for hints none - behavior is unconstrained</documentation>
    </annotation>
    <simpleType>
    <restriction base="string">
    <enumeration value="onLoad" />
    <enumeration value="onRequest" />
    <enumeration value="other" />
    <enumeration value="none" />
    </restriction>
    </simpleType>

```




```

</attribute>
<!-- traversal attributes -->
<attribute name="label" type="string" />
<attribute name="from" type="string" />
<attribute name="to" type="string" />
<!--
    Attributes grouped by XLink type, as specified in the W3C
    Proposed Recommendation (dated 2000-12-20)
-->
<attributeGroup name="simpleLink">
  <attribute name="type" type="string" fixed="simple" form="qualified" />
  <attribute ref="xlink:href" use="optional" />
  <attribute ref="xlink:role" use="optional" />
  <attribute ref="xlink:arcrole" use="optional" />
  <attribute ref="xlink:title" use="optional" />
  <attribute ref="xlink:show" use="optional" />
  <attribute ref="xlink:actuate" use="optional" />
</attributeGroup>
<attributeGroup name="extendedLink">
  <attribute name="type" type="string" fixed="extended" form="qualified" />
  <attribute ref="xlink:role" use="optional" />
  <attribute ref="xlink:title" use="optional" />
</attributeGroup>
<attributeGroup name="locatorLink">
  <attribute name="type" type="string" fixed="locator" form="qualified" />
  <attribute ref="xlink:href" use="required" />
  <attribute ref="xlink:role" use="optional" />
  <attribute ref="xlink:title" use="optional" />
  <attribute ref="xlink:label" use="optional" />
</attributeGroup>
<attributeGroup name="arcLink">
  <attribute name="type" type="string" fixed="arc" form="qualified" />
  <attribute ref="xlink:arcrole" use="optional" />
  <attribute ref="xlink:title" use="optional" />
  <attribute ref="xlink:show" use="optional" />
  <attribute ref="xlink:actuate" use="optional" />
  <attribute ref="xlink:from" use="optional" />
  <attribute ref="xlink:to" use="optional" />
</attributeGroup>
<attributeGroup name="resourceLink">
  <attribute name="type" type="string" fixed="resource" form="qualified" />
  <attribute ref="xlink:role" use="optional" />
  <attribute ref="xlink:title" use="optional" />
  <attribute ref="xlink:label" use="optional" />
</attributeGroup>
<attributeGroup name="titleLink">
  <attribute name="type" type="string" fixed="title" form="qualified" />

```



```
</attributeGroup>  
<attributeGroup name="emptyLink">  
  <attribute name="type" type="string" fixed="none" form="qualified" />  
</attributeGroup>  
</schema>
```




Anexo D. XML de definición del esquema de WFS.

D.1 Introducción.

Este anexo contiene las definiciones de esquemas XML de las operaciones de un WFS.

D.2 WFS-basic.xsd

```
<?xml version="1.0" ?>
<xsd:schema targetNamespace="http://www.opengis.net/wfs"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <!-- Includes and Imports -->
  <xsd:import namespace="http://www.opengis.net/gml"
    schemaLocation="../../gml/2.1.2/feature.xsd" />
  <xsd:import namespace="http://www.opengis.net/ogc"
    schemaLocation="../../filter/1.0.0/filter.xsd" />
  <!-- REQUEST MESSAGES -->
  <xsd:element name="GetCapabilities" type="wfs:GetCapabilitiesType">
  <xsd:annotation>
    <xsd:documentation>The GetCapabilities element is used to request that a Web
      Feature Service generate an XML document describing the organization
      providing the service, the WFS operations that the service supports, a list of
      feature types that the service can operate on and list of filtering capabilities
      that the service support. Such an XML document is called a capabilities
      document.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="DescribeFeatureType" type="wfs:DescribeFeatureTypeType">
  <xsd:annotation>
    <xsd:documentation>The DescribeFeatureType element is used to request that a
      Web Feature Service generate a document describing one or more feature
      types.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="GetFeature" type="wfs:GetFeatureType">
  <xsd:annotation>
    <xsd:documentation>The GetFeature element is used to request that a Web
      Feature Service return feature instances of one or more feature
      types.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
```



```

<!-- RESPONSE MESSAGES -->
<xsd:element name="FeatureCollection" type="wfs:FeatureCollectionType"
  substitutionGroup="gml:_FeatureCollection">
<xsd:annotation>
  <xsd:documentation>This element is a container for the response to a GetFeature
    or GetFeatureWithLock (WFS-transaction.xsd) request.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<!-- TYPES -->
<!-- GETCAPABILITIES -->
<xsd:complexType name="GetCapabilitiesType">
<xsd:annotation>
  <xsd:documentation>This type defines the GetCapabilities operation. In response
    to a GetCapabilities request, a Web Feature Service must generate a
    capabilities XML document that validates against the schemas defined in
    WFS-capabilities.xsd.</xsd:documentation>
</xsd:annotation>
  <xsd:attribute name="version" type="xsd:string" use="optional" fixed="1.0.0" />
  <xsd:attribute name="service" type="xsd:string" use="required" fixed="WFS" />
</xsd:complexType>
<!-- DESCRIBEFEATURETYPE -->
<xsd:complexType name="DescribeFeatureTypeType">
<xsd:annotation>
  <xsd:documentation>The DescribeFeatureType operation allows a client
    application to request that a Web Feature Service describe one or more
    feature types. A Web Feature Service must be able to generate feature
    descriptions as valid GML2 application schemas. The schemas generated by
    the DescribeFeatureType operation can be used by a client application to
    validate the output. Feature instances within the WFS interface must be
    specified using GML2. The schema of feature instances specified within the
    WFS interface must validate against the feature schemas generated by the
    DescribeFeatureType request.</xsd:documentation>
</xsd:annotation>
  <xsd:sequence>
    <xsd:element name="TypeName" type="xsd:QName" minOccurs="0"
      maxOccurs="unbounded">
    <xsd:annotation>
      <xsd:documentation>The TypeName element is used to enumerate the feature
        types to be described. If no TypeName elements are specified then all
        features should be described.</xsd:documentation>
    </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="version" type="xsd:string" use="required" fixed="1.0.0" />
  <xsd:attribute name="service" type="xsd:string" use="required" fixed="WFS" />
  <xsd:attribute name="outputFormat" type="xsd:string" use="optional"
    default="XMLSCHEMA">

```



```

<xsd:annotation>
  <xsd:documentation>The outputFormat attribute is used to specify what schema
    description language should be used to describe features. The default value
    of XMLSCHEMA means that the Web Feature Service must generate a GML2
    application schema that can be used to validate the GML2 output of a
    GetFeature request or feature instances specified in Transaction
    operations.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<!-- GETFEATURE -->
<xsd:complexType name="GetFeatureType">
<xsd:annotation>
  <xsd:documentation>A GetFeature element contains one or more Query elements
    that describe a query operation on one feature type. In response to a
    GetFeature request, a Web Feature Service must be able to generate a GML2
    response that validates using a schema generated by the
    DescribeFeatureType request. A Web Feature Service may support other
    possibly non-XML (and even binary) output formats as long as those formats
    are advertised in the capabilities document.</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
  <xsd:element ref="wfs:Query" maxOccurs="unbounded" />
</xsd:sequence>
  <xsd:attribute name="version" type="xsd:string" use="required" fixed="1.0.0" />
  <xsd:attribute name="service" type="xsd:string" use="required" fixed="WFS" />
  <xsd:attribute name="handle" type="xsd:string" use="optional" />
  <xsd:attribute name="outputFormat" type="xsd:string" use="optional"
    default="GML2">
<xsd:annotation>
  <xsd:documentation>The outputFormat attribute is used to specify the output
    format that the Web Feature Service should generate in response to a
    GetFeature or GetFeatureWithLock element. The default value of GML2
    indicates that the output is an XML document that conforms to the
    Geography Markup Language (GML) Implementation Specification V2.0.
    Other values may be used to specify other formats as long as those values
    are advertised in the capabilities document. For example, the value WKB may
    be used to indicate that a Well Known Binary format be used to encode the
    output.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
  <xsd:attribute name="maxFeatures" type="xsd:positiveInteger" use="optional">
<xsd:annotation>
  <xsd:documentation>The maxFeatures attribute is used to specify the maximum
    number of features that a GetFeature operation should generate (regardless
    of the actual number of query hits).</xsd:documentation>
</xsd:annotation>

```



```

</xsd:attribute>
</xsd:complexType>
<xsd:element name="Query" type="wfs:QueryType">
<xsd:annotation>
<xsd:documentation>The Query element is used to describe a single query. One
or more Query elements can be specified inside a GetFeature element so that
multiple queries can be executed in one request. The output from the various
queries are combined in a wfs:FeatureCollection element to form the
response to the request.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:complexType name="QueryType">
<xsd:annotation>
<xsd:documentation>The Query element is of type
QueryType.</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element ref="ogc:PropertyName" minOccurs="0" maxOccurs="unbounded">
<xsd:annotation>
<xsd:documentation>The PropertyName element is used to specify one or more
properties of a feature whose values are to be retrieved by a Web Feature
Service. While a Web Feature Service should endeavour to satisfy the exact
request specified, in some instance this may not be possible. Specifically, a
Web Feature Service must generate a valid GML2 response to a Query
operation. The schema used to generate the output may include properties
that are mandatory. In order that the output validates, these mandatory
properties must be specified in the request. If they are not, a Web Feature
Service may add them automatically to the Query before processing it. Thus
a client application should, in general, be prepared to receive more properties
than it requested. Of course, using the DescribeFeatureType request, a client
application can determine which properties are mandatory and request them
in the first place.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1">
<xsd:annotation>
<xsd:documentation>The Filter element is used to define spatial and/or non-
spatial constraints on query. Spatial constraints use GML2 to specify the
constraining geometry. A full description of the Filter element can be found in
the Filter Encoding Implementation Specification.</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="handle" type="xsd:string" use="optional" />
<xsd:attribute name="typeName" type="xsd:QName" use="required" />
<xsd:attribute name="featureVersion" type="xsd:string" use="optional">
<xsd:annotation>

```



```

<xsd:documentation>For systems that implement versioning, the featureVersion
  attribute is used to specify which version of a particular feature instance is to
  be retrieved. A value of ALL means that all versions should be retrieved. An
  integer value 'i', means that the ith version should be retrieve if it exists or the
  most recent version otherwise.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<!-- RESPONSE TYPE -->
<xsd:complexType name="FeatureCollectionType">
<xsd:annotation>
<xsd:documentation>This type defines a container for the response to a
  GetFeature or GetFeatureWithLock request. If the request is
  GetFeatureWithLock, the lockId attribute must be populated. The lockId
  attribute can otherwise be safely ignored.</xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
<xsd:extension base="gml:AbstractFeatureCollectionType">
<xsd:attribute name="lockId" type="xsd:string" use="optional">
<xsd:annotation>
<xsd:documentation>The value of the lockId attribute is an identifier that a Web
  Feature Service generates and which a client application can use in
  subsequent operations (such as a Transaction request) to reference the set
  of locked features.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

D.3 WFS-transaction.xsd

```

<?xml version="1.0" ?>
<xsd:schema targetNamespace="http://www.opengis.net/wfs"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wfs="http://www.opengis.net/wfs" elementFormDefault="qualified">
  <!-- Includes and Imports -->
  <xsd:include schemaLocation="WFS-basic.xsd" />
  <xsd:import namespace="http://www.opengis.net/gml"
    schemaLocation="../gml/2.1.2/feature.xsd" />
  <xsd:import namespace="http://www.opengis.net/ogc"
    schemaLocation="../filter/1.0.0/filter.xsd" />
  <!-- REQUEST MESSAGES -->
  <xsd:element name="GetFeatureWithLock" type="wfs:GetFeatureWithLockType">

```



```

<xsd:annotation>
  <xsd:documentation>This is the root element for the GetFeatureWithLock
    request. The GetFeatureWithLock operation performs identically to a
    GetFeature request except that the GetFeatureWithLock request locks all
    the feature instances in the result set and returns a lock identifier to a
    client application in the response.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="LockFeature" type="wfs:LockFeatureType">
<xsd:annotation>
  <xsd:documentation>This is the root element for a LockFeature request. The
    LockFeature request can be used to lock one or more feature
    instances.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Transaction" type="wfs:TransactionType">
<xsd:annotation>
  <xsd:documentation>This is the root element for a Transaction request. A
    transaction request allows insert, update and delete operations to be
    performed to create, change or remove feature
    instances.</xsd:documentation>
</xsd:annotation>
</xsd:element>
  <!-- RESPONSE MESSAGES -->
<xsd:element name="WFS_LockFeatureResponse"
  type="wfs:WFS_LockFeatureResponseType">
<xsd:annotation>
  <xsd:documentation>The WFS_LockFeatureResponse element contains a
    report about the completion status of a LockFeature
    request.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="WFS_TransactionResponse"
  type="wfs:WFS_TransactionResponseType">
<xsd:annotation>
  <xsd:documentation>The WFS_TransactionResponse element contains a report
    about the completion status of a Transaction
    operation.</xsd:documentation>
</xsd:annotation>
</xsd:element>
  <!-- GETFEATUREWITHLOCK -->
<xsd:complexType name="GetFeatureWithLockType">
<xsd:annotation>
  <xsd:documentation>A GetFeatureWithLock request operates identically to a
    GetFeature request expect that it attempts to lock the feature instances in
    the result set and includes a lock identifier in its response to a client. A
    lock identifier is an identifier generated by a Web Feature Service that a

```




```

    client application can use, in subsequent operations, to reference the
    locked set of feature instances.</xsd:documentation>
  </xsd:annotation>
</xsd:sequence>
<xsd:element ref="wfs:Query" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="version" type="xsd:string" use="required" fixed="1.0.0" />
<xsd:attribute name="service" type="xsd:string" use="required" fixed="WFS" />
<xsd:attribute name="handle" type="xsd:string" use="optional" />
<xsd:attribute name="expiry" type="xsd:positiveInteger" use="optional" />
<xsd:attribute name="outputFormat" type="xsd:string" use="optional"
  default="GML2" />
<xsd:attribute name="maxFeatures" type="xsd:positiveInteger" use="optional" />
</xsd:complexType>
<!-- LOCKFEATURE -->
<xsd:complexType name="LockFeatureType">
<xsd:annotation>
  <xsd:documentation>This type defines the LockFeature operation. The
    LockFeature element contains one or more Lock elements that define
    which features of a particular type should be locked. A lock identifier
    (lockId) is returned to the client application which can be used by
    subsequent operations to reference the locked
    features.</xsd:documentation>
</xsd:annotation>
</xsd:sequence>
<xsd:element name="Lock" type="wfs:LockType" maxOccurs="unbounded">
<xsd:annotation>
  <xsd:documentation>The lock element is used to indicate which feature
    instances of particular type are to be locked.</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="version" type="xsd:string" use="required" fixed="1.0.0" />
<xsd:attribute name="service" type="xsd:string" use="required" fixed="WFS" />
<xsd:attribute name="expiry" type="xsd:positiveInteger" use="optional" />
<xsd:attribute name="lockAction" type="wfs:AllSomeType" use="optional">
<xsd:annotation>
  <xsd:documentation>The lockAction attribute is used to indicate what a Web
    Feature Service should do when it encounters a feature instance that has
    already been locked by another client application. Valid values are ALL or
    SOME. ALL means that the Web Feature Service must acquire locks on all
    the requested feature instances. If it cannot acquire those locks then the
    request should fail. In this instance, all locks acquired by the operation
    should be released. SOME means that the Web Feature Service should lock
    as many of the requested features as it can.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>

```



```

</xsd:complexType>
<xsd:complexType name="LockType">
<xsd:annotation>
<xsd:documentation>This type defines the Lock element. The Lock element
defines a locking operation on feature instances of a single type. An OGC
Filter is used to constrain the scope of the operation. Features to be locked
can be identified individually by using their feature identifier or they can be
locked by satisfying the spatial and non-spatial constraints defined in the
filter.</xsd:documentation>
</xsd:annotation>
<xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="handle" type="xsd:string" use="optional" />
<xsd:attribute name="typeName" type="xsd:QName" use="required" />
</xsd:complexType>
<!-- TRANSACTION -->
<xsd:complexType name="TransactionType">
<xsd:annotation>
<xsd:documentation>The TransactionType defines the Transaction operation. A
Transaction element contains one or more Insert, Update Delete and Native
elements that allow a client application to create, modify or remove feature
instances from the feature repository that a Web Feature Service
controls.</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element ref="wfs:LockId" minOccurs="0">
<xsd:annotation>
<xsd:documentation>In order for a client application to operate upon locked
feature instances, the Transaction request must include the LockId
element. The content of this element must be the lock identifier the client
application obtained from a previous GetFeatureWithLock or LockFeature
operation. If the correct lock identifier is specified the Web Feature Service
knows that the client application may operate upon the locked feature
instances. No LockId element needs to be specified to operate upon
unlocked features.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<xsd:element ref="wfs:Insert" />
<xsd:element ref="wfs:Update" />
<xsd:element ref="wfs:Delete" />
<xsd:element ref="wfs:Native" />
</xsd:choice>
</xsd:sequence>
<xsd:attribute name="version" type="xsd:string" use="required" fixed="1.0.0" />
<xsd:attribute name="service" type="xsd:string" use="required" fixed="WFS" />
<xsd:attribute name="handle" type="xsd:string" use="optional" />

```




```

<xsd:attribute name="releaseAction" type="wfs:AllSomeType" use="optional">
<xsd:annotation>
  <xsd:documentation>The releaseAction attribute is used to control how a Web
    Feature service releases locks on feature instances after a Transaction
    request has been processed. Valid values are ALL or SOME. A value of ALL
    means that the Web Feature Service should release the locks of all feature
    instances locked with the specified lockId, regardless or whether or not the
    features were actually modified. A value of SOME means that the Web
    Feature Service will only release the locks held on feature instances that
    were actually operated upon by the transaction. The lockId that the client
    application obtained shall remain valid and the other, unmodified, feature
    instances shall remain locked. If the expiry attribute was specified in the
    original operation that locked the feature instances, then the expiry counter
    will be reset to give the client application that same amount of time to post
    subsequent transactions against the locked features.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<xsd:element name="LockId" type="xsd:string">
<xsd:annotation>
  <xsd:documentation>The LockId element contains the value of the lock
    identifier obtained by a client application from a previous
    GetFeatureWithLock or LockFeature request.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Insert" type="wfs:InsertElementType">
<xsd:annotation>
  <xsd:documentation>The Insert element is used to indicate that the Web
    Feature Service should create a new instance of a feature type. The feature
    instance is specified using GML2 and one or more feature instances to be
    created can be contained inside the Insert element.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="InsertElementType">
<xsd:sequence>
  <xsd:element ref="gml:_Feature" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="handle" type="xsd:string" use="optional" />
</xsd:complexType>
<xsd:element name="Update" type="wfs:UpdateElementType">
<xsd:annotation>
  <xsd:documentation>One or more existing feature instances can be changed by
    using the Update element. Changing a feature instance means that the
    current value of one or more properties of the feature are replaced with
    new values. The Update element contains one or more Property elements.
    A Property element contains the name or a feature property who's value is

```



```

    to be changed and the replacement value for that
    property.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="UpdateElementType">
  <xsd:sequence>
    <xsd:element ref="wfs:Property" maxOccurs="unbounded" />
    <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>The Filter element is used to constrain the scope of the
          update operation to those features identified by the filter. Feature instances
          can be specified explicitly and individually using the identifier of each
          feature instance OR a set of features to be operated on can be identified by
          specifying spatial and non-spatial constraints in the filter. If no filter is
          specified, then the update operation applies to all feature
          instances.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="handle" type="xsd:string" use="optional" />
    <xsd:attribute name="typeName" type="xsd:QName" use="required" />
  </xsd:complexType>
<xsd:element name="Delete" type="wfs:DeleteElementType">
  <xsd:annotation>
    <xsd:documentation>The Delete element is used to indicate that one or more
      feature instances should be removed from the feature
      repository.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
<xsd:complexType name="DeleteElementType">
  <xsd:sequence>
    <xsd:element ref="ogc:Filter" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>The Filter element is used to constrain the scope of the
          delete operation to those features identified by the filter. Feature instances
          can be specified explicitly and individually using the identifier of each
          feature instance OR a set of features to be operated on can be identified by
          specifying spatial and non-spatial constraints in the filter. If no filter is
          specified then an exception should be raised since it is unlikely that a
          client application intends to delete all feature
          instances.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="handle" type="xsd:string" use="optional" />
    <xsd:attribute name="typeName" type="xsd:QName" use="required" />
  </xsd:complexType>

```



```

<xsd:element name="Native" type="wfs:NativeType">
<xsd:annotation>
  <xsd:documentation>Many times, a Web Feature Service interacts with a
    repository that may have special vendor specific capabilities. The native
    element allows vendor specific command to be passed to the repository via
    the Web Feature Service.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:complexType name="NativeType">
<xsd:attribute name="vendorId" type="xsd:string" use="required">
<xsd:annotation>
  <xsd:documentation>The vendorId attribute is used to specify the name of
    vendor who's vendor specific command the client application wishes to
    execute.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="safeToIgnore" type="xsd:boolean" use="required">
<xsd:annotation>
  <xsd:documentation>In the event that a Web Feature Service does not
    recognize the vendorId or does not recognize the vendor specific
    command, the safeToIgnore attribute is used to indicate whether the
    exception can be safely ignored. A value of TRUE means that the Web
    Feature Service may ignore the command. A value of FALSE means that a
    Web Feature Service cannot ignore the command and an exception should
    be raised if a problem is encountered.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<!-- define structure to specify a property value -->
<xsd:element name="Property" type="wfs:PropertyType">
<xsd:annotation>
  <xsd:documentation>The Property element is used to specify the new value of a
    feature property inside an Update element.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:complexType name="PropertyType">
<xsd:sequence>
<xsd:element name="Name" type="xsd:string">
<xsd:annotation>
  <xsd:documentation>The Name element contains the name of a feature
    property to be updated.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Value" minOccurs="0">
<xsd:annotation>
  <xsd:documentation>The Value element contains the replacement value for the
    named property.</xsd:documentation>

```



```

</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<!-- RESPONSE TYPES -->
<xsd:complexType name="WFS_LockFeatureResponseType">
<xsd:annotation>
  <xsd:documentation>The WFS_LockFeatureResponseType is used to define an
    element to contains the response to a LockFeature
    operation.</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element ref="wfs:LockId">
<xsd:annotation>
  <xsd:documentation>The WFS_LockFeatureResponse includes a LockId
    element that contains a lock identifier. The lock identifier can be used by a
    client, in subsequent operations, to operate upon the locked feature
    instances.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="FeaturesLocked" type="wfs:FeaturesLockedType"
  minOccurs="0">
<xsd:annotation>
  <xsd:documentation>The LockFeature or GetFeatureWithLock operations
    identify and attempt to lock a set of feature instances that satisfy the
    constraints specified in the request. In the event that the lockAction
    attribute (on the LockFeature or GetFeatureWithLock elements) is set to
    SOME, a Web Feature Service will attempt to lock as many of the feature
    instances from the result set as possible. The FeaturesLocked element
    contains list of ogc:FeatureId elements enumerating the feature instances
    that a WFS actually managed to lock.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="FeaturesNotLocked" type="wfs:FeaturesNotLockedType"
  minOccurs="0">
<xsd:annotation>
  <xsd:documentation>In contrast to the FeaturesLocked element, the
    FeaturesNotLocked element contains a list of ogc:Filter elements
    identifying feature instances that a WFS did not manage to lock because
    they were already locked by another process.</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeaturesLockedType">
<xsd:sequence maxOccurs="unbounded">
  <xsd:element ref="ogc:FeatureId" />

```



```

    </xsd:sequence>
  </xsd:complexType>
<xsd:complexType name="FeaturesNotLockedType">
<xsd:sequence maxOccurs="unbounded">
  <xsd:element ref="ogc:FeatureId" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="WFS_TransactionResponseType">
<xsd:annotation>
  <xsd:documentation>The WFS_TransactionResponseType defines the format of
    the XML document that a Web Feature Service generates in response to a
    Transaction request. The response includes the completion status of the
    transaction and the feature identifiers of any newly created feature
    instances.</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="InsertResult" type="wfs:InsertResultType" minOccurs="0"
  maxOccurs="unbounded">
<xsd:annotation>
  <xsd:documentation>The InsertResult element contains a list of ogc:FeatureId
    elements that identify any newly created feature
    instances.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TransactionResult" type="wfs:TransactionResultType">
<xsd:annotation>
  <xsd:documentation>The TransactionResult element contains a Status element
    indicating the completion status of a transaction. In the event that the
    transaction fails, additional element may be included to help locate which
    part of the transaction failed and why.</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="version" type="xsd:string" use="required" fixed="1.0.0" />
</xsd:complexType>
<xsd:complexType name="TransactionResultType">
<xsd:sequence>
<xsd:element name="Status" type="wfs:StatusType">
<xsd:annotation>
  <xsd:documentation>The Status element contains an element indicating the
    completion status of a transaction. The SUCCESS element is used to
    indicate successful completion. The FAILED element is used to indicate
    that an exception was encountered.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Locator" type="xsd:string" minOccurs="0">
<xsd:annotation>

```



```

<xsd:documentation>In the event that an exception was encountered while
    processing a transaction, a Web Feature Service may use the Locator
    element to try and identify the part of the transaction that failed. If the
    element(s) contained in a Transaction element included a handle attribute,
    then a Web Feature Service may report the handle to identify the offending
    element.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Message" type="xsd:string" minOccurs="0">
<xsd:annotation>
<xsd:documentation>The Message element may contain an exception report
    generated by a Web Feature Service when an exception is
    encountered.</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="handle" type="xsd:string" use="optional" />
</xsd:complexType>
<xsd:complexType name="InsertResultType">
<xsd:sequence>
<xsd:element ref="ogc:FeatureId" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="handle" type="xsd:string" use="optional" />
</xsd:complexType>
<xsd:complexType name="StatusType">
<xsd:choice>
<xsd:element ref="wfs:SUCCESS" />
<xsd:element ref="wfs:FAILED" />
<xsd:element ref="wfs:PARTIAL" />
</xsd:choice>
</xsd:complexType>
<xsd:element name="SUCCESS" type="wfs:EmptyType" />
<xsd:element name="FAILED" type="wfs:EmptyType" />
<xsd:element name="PARTIAL" type="wfs:EmptyType" />
<!-- MISC TYPES -->
<xsd:complexType name="EmptyType" />
<xsd:simpleType name="AllSomeType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="ALL" />
<xsd:enumeration value="SOME" />
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

D.4 WFS-capabilities.xsd



```

<?xml version="1.0" ?>
<xsd:schema targetNamespace="http://www.opengis.net/wfs"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <!-- Imports ... -->
  <xsd:import namespace="http://www.opengis.net/ogc"
    schemaLocation="../../filter/1.0.0/filterCapabilities.xsd" />
  <!-- Global elements and attributes -->
  <xsd:element name="Abstract" type="xsd:string" />
  <xsd:element name="AccessConstraints" type="xsd:string" />
  <xsd:element name="Fees" type="xsd:string" />
  <xsd:element name="Keywords" type="xsd:string" />
  <xsd:element name="OnlineResource" />
  <xsd:element name="SRS" type="xsd:string" />
  <xsd:element name="Title" type="xsd:string" />
  <xsd:element name="Query" type="wfs:EmptyType" />
  <xsd:element name="Insert" type="wfs:EmptyType" />
  <xsd:element name="Update" type="wfs:EmptyType" />
  <xsd:element name="Delete" type="wfs:EmptyType" />
  <xsd:element name="Lock" type="wfs:EmptyType" />
  <!-- REDEFINE THIS ELEMENT AS NEEDED IN YOUR XML -->
  <xsd:element name="VendorSpecificCapabilities" type="xsd:string" />
  <!-- Root element -->
  <xsd:element name="WFS_Capabilities" type="wfs:WFS_CapabilitiesType" />
  <!-- Types -->
  <xsd:complexType name="WFS_CapabilitiesType">
  <xsd:sequence>
    <xsd:element name="Service" type="wfs:ServiceType" />
    <xsd:element name="Capability" type="wfs:CapabilityType" />
    <xsd:element name="FeatureTypeList" type="wfs:FeatureTypeListType" />
    <xsd:element ref="ogc:Filter_Capabilities" />
  </xsd:sequence>
  <xsd:attribute name="version" type="xsd:string" fixed="1.0.0" />
  <xsd:attribute name="updateSequence" type="xsd:nonNegativeInteger"
    default="0" />
  </xsd:complexType>
  <xsd:complexType name="ServiceType">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string" />
    <xsd:element ref="wfs:Title" />
    <xsd:element ref="wfs:Abstract" minOccurs="0" />
    <xsd:element ref="wfs:Keywords" minOccurs="0" />
    <xsd:element ref="wfs:OnlineResource" />
    <xsd:element ref="wfs:Fees" minOccurs="0" />

```



```

<xsd:element ref="wfs:AccessConstraints" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CapabilityType">
<xsd:sequence>
  <xsd:element name="Request" type="wfs:RequestType" />
  <xsd:element ref="wfs:VendorSpecificCapabilities" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeatureTypeListType">
<xsd:sequence>
  <xsd:element name="Operations" type="wfs:OperationsType" minOccurs="0" />
  <xsd:element name="FeatureType" type="wfs:FeatureTypeType"
    maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RequestType">
<xsd:choice maxOccurs="unbounded">
  <xsd:element name="GetCapabilities" type="wfs:GetCapabilitiesType" />
  <xsd:element name="DescribeFeatureType"
    type="wfs:DescribeFeatureTypeType" />
  <xsd:element name="Transaction" type="wfs:TransactionType" />
  <xsd:element name="GetFeature" type="wfs:GetFeatureTypeType" />
  <xsd:element name="GetFeatureWithLock" type="wfs:GetFeatureTypeType" />
  <xsd:element name="LockFeature" type="wfs:LockFeatureTypeType" />
</xsd:choice>
</xsd:complexType>
<xsd:complexType name="GetCapabilitiesType">
<xsd:sequence>
  <xsd:element name="DCPType" type="wfs:DCPTypeType"
    maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DescribeFeatureTypeType">
<xsd:sequence>
  <xsd:element name="SchemaDescriptionLanguage"
    type="wfs:SchemaDescriptionLanguageType" />
  <xsd:element name="DCPType" type="wfs:DCPTypeType"
    maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TransactionType">
<xsd:sequence>
  <xsd:element name="DCPType" type="wfs:DCPTypeType"
    maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>

```




```

<xsd:complexType name="GetFeatureTypeType">
<xsd:sequence>
  <xsd:element name="ResultFormat" type="wfs:ResultFormatType" />
  <xsd:element name="DCPType" type="wfs:DCPTypeType"
    maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="LockFeatureTypeType">
<xsd:sequence>
  <xsd:element name="DCPType" type="wfs:DCPTypeType"
    maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DCPTypeType">
<xsd:sequence>
  <xsd:element name="HTTP" type="wfs:HTTPType" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeatureTypeType">
  <xsd:element name="Name" type="xsd:QName" />
  <xsd:element ref="wfs:Title" minOccurs="0" />
  <xsd:element ref="wfs:Abstract" minOccurs="0" />
  <xsd:element ref="wfs:Keywords" minOccurs="0" />
  <xsd:element ref="wfs:SRS" />
  <xsd:element name="Operations" type="wfs:OperationsType" minOccurs="0" />
  <xsd:element name="LatLongBoundingBox"
    type="wfs:LatLongBoundingBoxType" minOccurs="0"
    maxOccurs="unbounded" />
  <xsd:element name="MetadataURL" type="wfs:MetadataURLType"
    minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="GetType">
  <xsd:attribute name="onlineResource" type="xsd:string" use="required" />
</xsd:complexType>

<xsd:complexType name="HTTPType">
<xsd:choice maxOccurs="unbounded">
  <xsd:element name="Get" type="wfs:GetType" />
  <xsd:element name="Post" type="wfs:PostType" />
</xsd:choice>
</xsd:complexType>
<xsd:complexType name="LatLongBoundingBoxType">
  <xsd:attribute name="minx" type="xsd:string" use="required" />
  <xsd:attribute name="miny" type="xsd:string" use="required" />
  <xsd:attribute name="maxx" type="xsd:string" use="required" />
  <xsd:attribute name="maxy" type="xsd:string" use="required" />

```



```

    </xsd:complexType>
<xsd:complexType name="MetadataURLType">
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute name="type" use="required">
<xsd:simpleType>
<xsd:restriction base="xsd:NMTOKEN">
  <xsd:enumeration value="TC211" />
  <xsd:enumeration value="FGDC" />
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="format" use="required">
<xsd:simpleType>
<xsd:restriction base="xsd:NMTOKEN">
  <xsd:enumeration value="XML" />
  <xsd:enumeration value="SGML" />
  <xsd:enumeration value="TXT" />
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="OperationsType">
<xsd:choice maxOccurs="unbounded">
  <xsd:element ref="wfs:Insert" />
  <xsd:element ref="wfs:Update" />
  <xsd:element ref="wfs:Delete" />
  <xsd:element ref="wfs:Query" />
  <xsd:element ref="wfs:Lock" />
</xsd:choice>
</xsd:complexType>
<xsd:complexType name="PostType">
  <xsd:attribute name="onlineResource" type="xsd:string" use="required" />
  </xsd:complexType>
  <xsd:element name="GML2" type="wfs:EmptyType" />
<xsd:complexType name="ResultFormatType">
<xsd:sequence maxOccurs="unbounded">
  <xsd:element ref="wfs:GML2" />
</xsd:sequence>
</xsd:complexType>
  <xsd:element name="XMLSCHEMA" type="wfs:EmptyType" />
  <xsd:complexType name="SchemaDescriptionLanguageType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="wfs:XMLSCHEMA" />
  </xsd:sequence>

```



```
</xsd:complexType>  
<xsd:complexType name="EmptyType" />  
</xsd:schema>
```