

**GUÍA METODOLÓGICA PARA LA GENERACIÓN DE SERVICIOS EN  
LÍNEA A PARTIR DE LOS ESTÁNDARES WFS Y WMS BASADOS EN  
VISUALIZACIÓN CON TRÁFICO LIVIANO Y MANEJO DE SEGURIDAD**



**BIBIANA ALEXANDRA LARA TABARES  
ANA MARÍA RODRÍGUEZ ARÉVALO**

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA DE SISTEMAS  
BOGOTÁ, D.C.  
2004**

**GUÍA METODOLÓGICA PARA LA GENERACIÓN DE SERVICIOS EN  
LÍNEA A PARTIR DE LOS ESTÁNDARES WFS Y WMS BASADOS EN  
VISUALIZACIÓN CON TRÁFICO LIVIANO Y MANEJO DE SEGURIDAD**



**BIBIANA ALEXANDRA LARA TABARES  
ANA MARÍA RODRÍGUEZ ARÉVALO**

Proyecto de grado para optar al título de  
Ingeniero de Sistemas

Director

**JAVIER FRANCISCO LÓPEZ PARRA**  
Ingeniero de Sistemas

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA DE SISTEMAS  
BOGOTÁ, D.C.  
2004**

**NOTA DE ACEPTACIÓN**

---

---

---

---

---

**Director del Proyecto**

---

**Jurado**

---

**Jurado**

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA DE SISTEMAS**

RECTOR MAGNÍFICO:	R.P. GERARDO REMOLINA VARGAS S.J.
DECANO ACADÉMICO:	ING. ROBERTO ENRIQUE MONTOYA VILLA
DECANO DEL MEDIO UNIVERSITARIO:	R.P. ANTONIO JOSÉ SARMIENTO
DIRECTOR DE CARRERA:	ING. HILDA CRISTINA CHAPARRO LÓPEZ
DIRECTOR DE DEPARTAMENTO:	ING. GERMÁN ALBERTO CHAVARRO FLÓREZ
DIRECTOR DE PROYECTO:	ING. JAVIER FRANCISCO LÓPEZ PARRA

## **ARTÍCULO 23 DE LA RESOLUCIÓN NO. 01 DE JUNIO DE 1946**

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.

Solo velará porque no se publique nada contrario al dogma y la moral católica porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la justicia”.

*No puedo parar de trabajar.  
Tendré toda la eternidad para descansar  
Agnes Gonxha Bojaxhiu,  
Madre Teresa de Calcuta.  
Misionera Yugoslava*

## **AGRADECIMIENTOS**

Los autores agradecen cordialmente a:

Ing. Javier Francisco López Parra, Director del Proyecto de Investigación por su apoyo, revisiones y aportes.

Ing. Ricardo González por su valiosa orientación en el desarrollo de la arquitectura del prototipo.

Empresa de Acueducto y Alcantarillado de Bogotá (EAAB), por brindar el apoyo y la información requerida para llevar a cabo el caso de estudio, y el desarrollo del prototipo con información geográfica de dicha entidad.

Dedicamos este trabajo de investigación a Dios, a nuestras familias y a nuestros amigos.

## TABLA DE CONTENIDO

<b>1. INTRODUCCIÓN.....</b>	<b>20</b>
<b>2. MARCO TEÓRICO.....</b>	<b>22</b>
2.1 Sistemas de Información Geográfica (SIG).....	22
2.1.1 Definición.....	22
2.1.2 Componentes de un SIG.....	23
2.1.3 Modelos de Datos Espaciales.....	24
2.1.4 Bases de Datos Geográficas .....	25
2.1.5 Herramientas para generar servicios en línea para SIG .....	27
2.1.5.1. ArcIMS .....	27
2.1.5.2. Otras herramientas .....	29
2.2 Web Services.....	29
2.2.1 Definición.....	29
2.2.2 Estados de un Web Service.....	32
2.2.3 Características de los Web Services .....	32
2.2.4 Componentes de un Web Service .....	33
2.2.5 Objetivos de los Web Services .....	34
2.2.6 Arquitectura .....	34
2.2.7 Protocolos y estándares de los Web Services .....	35
2.2.7.1 Hypertext Transfer Transport Protocol (HTTP) .....	35
2.2.7.1.1 Métodos .....	35
2.2.7.1.2 Request .....	36
2.2.7.2. Universal Discovery Description Integration (UDDI) .....	36
2.2.7.2.1. Paginas Blancas, Amarillas y Verdes .....	38
2.2.7.2.2. Estructura de Datos de UDDI .....	38
2.2.7.2.3. API UDDI.....	39
2.2.7.3. Web Service Description Language (WSDL) .....	40
2.2.7.4. Simple Object Access Protocol (SOAP).....	42
2.2.7.4.1. Ventajas de SOAP.....	43
2.2.7.4.2. Objetivos de SOAP .....	43
2.2.7.4.3. Funcionamiento de SOAP .....	43
2.2.7.4.4. Factores en escenarios distribuidos .....	44
2.2.7.4.5. Mensaje SOAP .....	45
2.2.7.5. Extensible Markup Language (XML).....	45
2.2.7.6. XML Stylesheets Language Transformation (XSLT) .....	46
2.2.7.6.1. Proceso de transformación.....	48
2.3. Servicios en Línea para SIG.....	49

2.3.1.	Seguridad en los Servicios en Línea .....	50
2.3.1.1.	Seguridad a nivel de Web .....	50
2.3.1.2.	Seguridad a nivel de Negocio .....	51
2.3.1.3.	Seguridad a nivel de Datos .....	52
2.4.	Herramientas de Desarrollo para Servicios en Línea.....	52
2.4.1.	Sistemas Administradores de Bases de Datos Relacionales (RDBMS).....	53
2.4.1.1.	MySQL.....	53
2.4.1.2.	Microsoft SQL Server .....	54
2.4.1.3.	Oracle .....	55
2.4.2.	Plataformas de desarrollo .....	56
2.4.2.1.	Microsoft .NET .....	56
2.4.2.2.	Java 2 Enterprise Edition (J2EE) .....	58
2.5.	Estándares Open GIS Consortium (OGC) .....	59
2.5.1.	Web Map Service (WMS).....	59
2.5.1.1.	Operaciones Web Map Service.....	61
2.5.1.1.1.	Get Capabilities.....	61
2.5.1.1.2.	Get Map .....	62
2.5.1.1.3.	GetFeatureInfo .....	63
2.5.2.	Web Feature Service (WFS) .....	64
2.5.2.1.	Requerimientos para un Servicio Web Feature .....	65
2.5.2.2.	Operaciones Web Feature Service .....	67
2.5.2.2.1.	GetCapabilities.....	67
2.5.2.2.2.	DescribeFeatureType .....	68
2.5.2.2.3.	GetFeature .....	69
2.5.2.2.4.	Transaction .....	69
2.5.2.2.5.	LockFeature .....	70
2.5.3.	Geography Markup Language (GML) .....	71
2.5.3.1.	Elementos que permite manipular GML.....	72
2.5.3.2.	Ventajas de GML.....	73
2.5.3.3.	Desventajas de GML.....	76
2.5.3.4.	Esquemas de documentos GML .....	76
2.5.3.5.	Instancias de documentos GML.....	76
2.6.	Visualización de Mapas .....	77
2.6.1.	Scalable Vector Graphic (SVG) .....	77
2.6.1.1.	Ventajas del formato SVG .....	77
2.6.1.2.	Herramientas para crear y ver SVG .....	78
2.6.2.	Shape .....	80
<b>3.</b>	<b>GUÍA METODOLÓGICA.....</b>	<b>81</b>
3.1.	Introducción .....	81
3.2.	Fase de Planeación .....	81
3.2.1.	Identificación de la necesidad de generar servicios en línea .....	81
3.2.2.	Factores a tener en cuenta cuando se planea construir un servicio en línea	83
3.3.	Fase de Análisis.....	84

3.3.1.	Análisis de la situación actual .....	84
3.3.2.	Levantamiento de requerimientos.....	87
3.3.2.1.	Requerimientos Funcionales.....	89
3.3.2.1.1.	Web Map Service (WMS) .....	89
3.3.2.1.2.	Web Feature Service (WFS).....	93
3.3.2.2.	Requerimientos no Funcionales.....	99
3.3.2.2.1.	Alta disponibilidad .....	99
3.3.2.2.2.	Alto desempeño .....	102
3.4.	Fase de Diseño.....	104
3.4.1.	Recomendaciones de diseño de servicios en línea.....	105
3.4.2.	Fuentes de Datos .....	106
3.4.3.	Tipos de Servicios.....	107
3.4.4.	Diseño de Base de Datos .....	108
3.4.5.	Diseño de la Arquitectura.....	111
3.4.5.1.	Factores que debe proveer la arquitectura .....	111
3.4.5.2.	Sistema Distribuido.....	112
3.4.5.3.	Patrones de Software .....	113
3.4.5.4.	Eficiencia y Seguridad.....	115
3.4.5.5.	Modelo Multicapa: .....	118
3.4.5.5.1.	Capa de Datos .....	119
3.4.5.5.2.	Capa de Lógica de Servicios .....	120
3.4.5.5.3.	Capa de Adaptador de Servicios.....	122
3.4.5.5.4.	Capa de Cliente .....	123
3.4.5.6.	Arquitectura para la generación de servicios WMS y WFS .....	123
3.5.	Fase de Implementación .....	125
3.5.1.	Patrones de Implementación .....	125
3.5.2.	Modelo de clases e interfaces para la arquitectura por capas .....	126
3.5.2.1.	Capa de Datos.....	126
3.5.2.2.	Capa de Lógica de Servicio .....	128
3.5.2.3.	Capa de Adaptador de Servicio .....	130
3.5.2.4.	Capa de Cliente .....	131
3.5.3.	Implementación de las operaciones básicas de WMS y WFS .....	133
3.5.4.	Diseño de Interfaces de usuario .....	136
3.5.5.	Despliegue de Información al Cliente .....	136
3.6.	Fase de Soporte y Mantenimiento.....	138
3.6.1.	Pruebas al sistema.....	138
3.6.2.	Despliegue e instalación del sistema.....	140
3.6.3.	Mantenimiento del Sistema.....	140
3.7.	Resultados esperados .....	140
<b>4.</b>	<b>PROTOTIPO .....</b>	<b>141</b>
4.1.	Introducción .....	141
4.2.	Caso de Estudio.....	141
4.2.1.	Análisis de entidades generadoras de Información Geográfica de Bogotá .....	142
4.2.2.	Selección de Caso de Estudio .....	143

4.3.	Desarrollo de la Guía Metodológica en la entidad seleccionada .....	143
4.3.1.	Fase de Planeación .....	144
4.3.1.1.	Identificación de la necesidad de generar servicios en línea....	144
4.3.1.2.	Factores a tener en cuenta cuando se planea construir un servicio en línea.....	144
4.3.2.	Fase de Análisis.....	146
4.3.2.1.	Análisis de la situación actual .....	146
4.3.2.2.	Levantamiento de requerimientos .....	151
4.3.3.	Fase de Diseño .....	151
4.3.3.1.	Recomendaciones de diseño de servicios en línea .....	151
4.3.3.2.	Fuentes de Datos .....	152
4.3.3.3.	Tipos de Servicios .....	153
4.3.3.4.	Diseño de Base de Datos.....	154
4.3.3.5.	Diseño de la Arquitectura .....	155
4.3.4.	Fase de Implementación.....	156
4.3.4.1.	Patrones de Implementación.....	156
4.3.4.2.	Modelo de clases e interfaces para la arquitectura por capas..	157
4.3.4.3.	Implementación de las operaciones básicas de WMS y WFS..	160
4.3.4.4.	Diseño de Interfaces de usuario.....	161
4.3.4.5.	Despliegue de Información al Cliente.....	161
4.3.5.	Fase de Soporte y Mantenimiento .....	162
4.3.5.1.	Pruebas al sistema .....	162
4.3.5.2.	Despliegue e instalación del sistema .....	162
4.3.5.3.	Mantenimiento del Sistema .....	163
4.4.	Análisis de eficiencia en la generación de servicios de Mapas .....	164
4.4.5.	Uso de la herramienta ArcIMS.....	165
4.4.6.	Uso del modelo planteado en la Guía Metodológica .....	166
4.4.7.	Análisis Comparativo de eficiencia .....	166
<b>5.</b>	<b>CONCLUSIONES.....</b>	<b>171</b>
<b>6.</b>	<b>BIBLIOGRAFÍA.....</b>	<b>173</b>

## LISTA DE FIGURAS

Figura 1. Arquitectura de ArcIMS.....	28
Figura 2. Arquitectura de Web Services .....	34
Figura 3. Estructura de Datos UDDI .....	39
Figura 4. Definiciones Abstractas WSDL.....	42
Figura 5. Esquema de funcionamiento de SOAP .....	44
Figura 6. Seguridad de Base de Datos.....	52
Figura 7. Arquitectura Framework de .NET .....	58
Figura 8. Arquitectura J2EE .....	59
Figura 9. Arquitectura Simple WFS .....	65
Figura 10. Arquitectura aplicación SVG.....	79
Figura 11. Levantamiento de la situación actual .....	87
Figura 12. Factores de modelamiento de casos de uso.....	89
Figura 13. Esquema Casos de Uso WMS .....	93
Figura 14. Esquema Casos de Uso WFS.....	99
Figura 15. Modelo Entidad Relación para capacidades de los servidores .....	109
Figura 16. Modelo Entidad Relación para Datos Geográficos .....	110
Figura 17. Modelo Entidad Relación para Seguridad.....	111
Figura 18. Factores para conseguir eficiencia.....	117
Figura 19. Modelo de Seguridad Servicios en Línea.....	118
Figura 20. Modelo de Capas.....	119
Figura 21. Arquitectura para la generación de Servicios en Línea .....	124
WMS y WFS.....	124
Figura 22. Esquema de paquetes y clases de la Capa de Datos .....	128
Figura 23. Esquema de paquetes y clases de la Capa de .....	130
Lógica de Servicio .....	130
Figura 24. Esquema de paquetes y clases de la Capa de .....	131
Adaptador de Servicios.....	131
Figura 25. Esquema de paquetes y clases de la Capa de Cliente.....	132
Figura 26. Esquema de funcionamiento de la operación GetCapabilities .....	133
Figura 27. Esquema de funcionamiento de la operación GetMap .....	134
Figura 28. Esquema de Transformación de GML a SVG.....	135
Figura 29. Modelo de datos geográficos de la EAAB.....	145
Figura 30. Arquitectura de Seguridad de Servicios en línea EAAB .....	148
Figura 31. Arquitectura Servicios en línea EAAB .....	150

Figura 32. Niveles de Información para cada modelo de datos de la EAAB .....	153
Figura 33. Esquema de paquetes por capas del modelo DLAC .....	160
Figura 34. Esquema comparativo DLAC y ArcIMS 1a Solicitud.....	168
Figura 35. Esquema comparativo DLAC y ArcIMS 2a Solicitud.....	168
Figura 36. Esquema comparativo DLAC y ArcIMS 3a Solicitud.....	169
Figura 37. Esquema comparativo DLAC y ArcIMS 4a Solicitud.....	169

## LISTA DE TABLAS

Tabla 1. Componentes de un SIG .....	24
Tabla 2. Comparación entre Modelos de Datos Espaciales .....	25
Tabla 3. Estados de un Web Service .....	32
Tabla 4. Páginas Blancas, Amarillas y Verdes (UDDI) .....	38
Tabla 5. Elementos de un documento WSDL .....	41
Tabla 6. Características de MySQL.....	53
Tabla 7. Características de Microsoft SQL Server .....	54
Tabla 8. Características de Oracle .....	56
Tabla 9. Operaciones WMS .....	60
Tabla 10. Request GetCapabilities (WMS) .....	62
Tabla 11. Funciones WFS .....	66
Tabla 12. Request GetCapabilities (WFS) .....	67
Tabla 13. Request DescribeFeatureType (WFS).....	68
Tabla 14. Niveles de Requerimiento .....	88
Tabla 15. Términos de Alta Disponibilidad.....	100
Tabla 16. Ventajas Grupos lógicos en Capa de Datos .....	107
Tabla 17. Factores que provee la arquitectura.....	112
Tabla 18. Ventajas del formato SVG .....	138
Tabla 19. Grupos lógicos de datos para la EAAB.....	152
Tabla 20. Análisis de tiempos de respuesta ArcIMS y Prototipo .....	167
Tabla 21. Análisis de tiempos de respuesta ArcIMS y Prototipo .....	167

## LISTA DE ANEXOS

- Anexo A. Parámetros del Request para la operación Get Map de Web Map Service (WMS)
- Anexo B. Parámetros del Request para la operación Get Feature Info de Web Map Service (WMS)
- Anexo C. Parámetros del Request para la operación Get Feature de Web Feature Service (WFS)
- Anexo D. Parámetros del Request para la operación Transaction de Web Feature Service (WFS)
- Anexo E. Parámetros del Request para la operación Lock Feature de Web Feature Service (WFS)
- Anexo F. Formato de Cuestionario aplicado a las entidades de Bogotá que producen Información Geográfica
- Anexo G. Cuadro de criterios de selección para el caso de estudio
- Anexo H. Formato de Especificación de Requerimientos del Negocio
- Anexo I. Formato de Especificación de Requerimientos de Usuario en Casos de Uso
- Anexo J. Formato de Especificación de Requerimientos Funcionales
- Anexo K. Formato de Especificación de Requerimientos no Funcionales
- Anexo L. Especificación de Requerimientos del Negocio para la EAAB
- Anexo M. Especificación de Requerimientos Funcionales para la EAAB
- Anexo N. Especificación de Requerimientos no Funcionales para la EAAB
- Anexo O. Esquema de las Capacidades de los servidores WMS y WFS
- Anexo P. Modelos del aplicativo SIGUE de la EAAB
- Anexo Q. Diagrama de Clases del Prototipo Funcional
- Anexo R. Descripción del contenido del CD

## GLOSARIO

- **SIG (Sistemas de Información Geográfica)**

Sistema de hardware, software y procedimientos almacenados para facilitar la obtención, gestión, manipulación, análisis, modelado y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión.

- **HTTP (Hypertext Transfer Transport Protocol)**

Protocolo genérico sin estado orientado a objetos, que provee la ligereza y la velocidad necesarias para un sistema de información de colaboración distribuido, y el cual se puede utilizar para muchas tareas similares tales como servidores de nombres.

- **XML (Extensible Markup Language)**

Lenguaje de marcas que provee un conjunto de reglas, pautas y convenciones para planificar formatos de texto para datos, de manera que produzcan archivos fácilmente generados y leídos, para proveer extensibilidad, soporte para la internacionalización o localización, y la independencia de plataforma.

- **XSLT (XML Stylesheets Language Transformation)**

Lenguaje de transformación basado en hojas de estilo, el cual es utilizado para convertir documentos XML en otros documentos XML. Los programas XSLT son escritos en XML y requieren un procesador de hojas de estilo para procesarlas aplicándolas a un fichero XML.

- **SVG (Scalable Vector Graphics)**

Formato estándar de fichero vectorial para la web, desarrollado por la W3C, el cual es extensible, lo que permite adicionar definiciones propias sin perder la compatibilidad con las definiciones estándar, y además separa totalmente el contenido de la presentación.

- **OGC (Open GIS Consortium)**

Consortio cuya misión es la estandarización de los servicios relacionados con la geografía en entornos distribuidos, para lo cual han elaborado especificaciones abstractas que satisfacen el primer nivel de las necesidades propias de los SIG, como el renderizado de datos geográficos.

- **WMS (Web Map Service)**

Especificación creada por el OGC que describe el comportamiento de un servicio que produce mapas georeferenciados y las operaciones para recuperar una descripción de los mapas ofrecidos por una instancia del servicio, para recuperar un mapa, y consultar a un servidor acerca de las características mostradas en el mapa.

- **WFS (Web Feature Service)**

Estándar definido por el OGC que describe la especificación de codificación para geodatos en XML. La codificación descrita ha intentado activar el transporte y almacenamiento de información geográfica en XML incluyendo ambas propiedades la geometría y los rasgos geográficos

- **GML (Geography Markup Language)**

Gramática XML definida por el OGC, la cual está escrita en XML Schema para cumplir funciones como modelar, transportar y almacenar información geográfica, que provee varias clases de objetos para describir geografía, tales como rasgos, los sistemas de referencia de coordenadas, la geometría, la topología, el clima, unidades de medida y valores generalizados.

## **OBJETIVO GENERAL**

Crear una guía metodológica para la generación de servicios en línea con base en los estándares del OpenGIS WMS(Web Map Services) y WFS(Web Features Services), GML(Geography Markup Language), así como XML(Extensible Markup Language), para el manejo de información geográfica y visualización de mapas, por medio de la interpretación de la imagen al lado del cliente de forma eficiente y segura.

## OBJETIVOS ESPECÍFICOS

1. Investigar, entender y usar los estándares del Open GIS(WMS, WFS, GML) y XML, para el manejo de servicios en línea basados en información geográfica, así como las herramientas necesarias para proveer servicios de visualización, que permitan la interpretación de la imagen al lado del cliente y el fundamento para soportar seguridad en sitios de éste tipo, esto basado en una arquitectura, que determinará los elementos esenciales que se necesitan para garantizar eficiencia y seguridad.
2. Hacer el levantamiento de requerimientos funcionales y no funcionales, con los usuarios del caso de estudio, quienes interactuarán con los servicios en línea, así como sus roles y permisos asociados y con base en la dimensión de la información que se podría consultar, poder establecer políticas de seguridad adecuadas.
3. Construir una primera aproximación a la guía metodológica, que sea la base para ir refinando la metodología que permitirá la generación de servicios en línea garantizando la seguridad y la rapidez en la transmisión de imágenes.
4. Comprobar la guía metodológica contra un caso de estudio, por medio de un prototipo generado a partir de información georeferenciada de la entidad seleccionada para dicho fin, que ofrezca los servicios planteados para hacer un análisis del diseño en cuanto a eficiencia y seguridad, de esta forma, confirmar la eficiencia del uso de los estándares geográficos, políticas, arquitectura e infraestructura utilizada en el prototipo.

## 1. INTRODUCCIÓN

El presente trabajo de grado tiene como fin, poder aplicar los conocimientos adquiridos en la carrera, haciendo énfasis en los Sistemas de Información Geográfica(SIG), y los servicios que éstos proveen a través de la Web.

Se ofrece al lector un trabajo de tipo investigativo y práctico, en el sentido que se presentan las bases teóricas para la creación de servicios en línea geográficos, basados en los estándares WMS y WFS propuestos por el OGC, y posteriormente se plantea una guía metodológica para la generación de dichos servicios, de modo que sean eficientes y seguros, por medio de la interpretación de los mapas al lado del cliente.

A lo largo de la guía metodológica se resalta el uso de tecnologías como servicios en línea, sistemas distribuidos y estándares especializados para el manejo e intercambio de información geográfica sobre la Web. La guía metodológica está organizada por fases para un desarrollo de software, razón por la cual, es de tipo práctico, en el sentido que se presentan las pautas a seguir en cada una de las fases, lo que determina un correcto desarrollo desde el comienzo.

Una vez planteada la guía metodológica, se demuestra su funcionamiento a través del desarrollo de un prototipo funcional, realizado por medio de un caso de estudio adaptando directamente la guía metodológica creada.

Con el presente trabajo, se contribuye a la construcción de conocimiento y desarrollo de servicios en línea geográficos interoperables, los cuales ofrecen calidad y eficiencia, sin embargo, actualmente los estándares geográficos no son muy usados debido a su desconocimiento y complejidad, por lo cual, quienes requieren manejar información geográfica en la Web no se concientizan de la importancia y las grandes ventajas del uso de los mismos.

Los servicios geográficos en la Web que ofrecen las organizaciones a sus clientes sobre la información que poseen, son elementales cuando se trata de clientes que requieren compartir dicha información, porque se utilizan para realizar análisis de gestión, que permita tomar decisiones adecuadas con base en la información obtenida.

A partir de la importancia de compartir información, la guía metodológica plantea el uso de servicios Web interoperables, por medio de software libre, para llevar a cabo las funciones de generación de servicios geográficos, teniendo en cuenta que actualmente existen herramientas costosas que permiten proveer servicios geográficos a los clientes, y que además proveen mapas estáticos en formato ráster, lo cual implica mayor tráfico en la red. La guía metodológica en este sentido, propone generar mapas dinámicos por medio de formatos y estándares especializados para dicho fin.

El principal aporte del presente trabajo de grado, es proveer a los usuarios interesados en la generación de servicios en línea geográficos, una guía metodológica práctica, la cual presenta una serie de pasos organizados de tal modo que facilite y optimice el proceso de generación de dichos servicios, con el objetivo de garantizar calidad y eficiencia para el usuario, teniendo en cuenta los grandes volúmenes de información que se tienen al generar y manipular mapas geográficos.

## 2. MARCO TEÓRICO

### 2.1 Sistemas de Información Geográfica (SIG)

#### 2.1.1 Definición

Los Sistemas de Información Geográfica (GIS su sigla en inglés: Geographic Information Systems) actualmente son un soporte tecnológico para las entidades o entes que manejan información geográfica.

La definición que más se adapta a este proyecto y a la industria actual basada en información geográfica, es la propuesta por la National Center for Geographic Information and Analysis(NCGIA). Este centro define a un SIG como : “Un sistema de hardware, software y procedimientos almacenados para facilitar la obtención, gestión, manipulación, análisis, modelado y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión.”<sup>1</sup>

Teniendo en cuenta esta definición, se tiene que los SIG abarcan funciones como almacenar y analizar información geográfica, la cual se debe presentar al cliente de la mejor forma, y además permitirle consultarla y manipularla con base en los permisos que tenga sobre los diferentes niveles de información. Los Sistemas de Información Geográfica, así como los sistemas de otras áreas donde se maneja otro tipo de información, pueden tener diferentes tipos de clientes, como es el caso de cliente de aplicación de ventana, cliente Web o cliente móvil (asociado a dispositivos móviles).

Particularmente los SIG en Internet proveen servicios que son llamados servicios en línea, y son la base sobre la cual se enfoca este proyecto, ya que es muy importante el acceso externo a información geográfica interna de cierta entidad o entidades que manejen información geográfica de interés para determinada población que la requiere con diferentes fines.

Un factor elemental en cuanto a facilidad para el acceso a los servicios en línea vía Internet, es que no se requieren instalaciones de software al lado del cliente, sino simplemente acceso a Internet por un navegador Web.

---

<sup>1</sup> [www.ncgia.ucsb.edu](http://www.ncgia.ucsb.edu) Fecha de Consulta: Agosto de 2004.

Un SIG es diseñado para la captura, el almacenamiento, el análisis y la utilización de objetos y fenómenos, donde la locación geográfica es una característica importante o crítica para operaciones de análisis, lo cual lleva a obtener resultados importantes para la toma de decisiones a nivel de gestión y de administración de la información que se maneje.

El poder de los sistemas de información geográfica es notable cuando la cantidad de los datos es muy grande para ser manejados manualmente, es entonces cuando el SIG provee operaciones sobre dichos volúmenes de información para poderla manipular de manera más sencilla, con el objetivo de obtener información útil para al cliente, el cual a su vez puede ser una persona o un grupo de personas, que forman parte de un contexto organizacional en el cual se debe dar un SIG.

La información que es requerida por un cliente provee el contexto fundamental en el cual el SIG debería funcionar. Para que la información pueda ser usada por un cliente, ésta debe ser de alta calidad y debe ser presentada en un formato adecuado para el tipo de cliente. La información en un SIG es presentada en dos formas básicas: como mapas y como tablas, las cuales permiten realizar diferentes tipos de análisis sobre los datos.

Teniendo en cuenta que un SIG es un sistema de información, éste se rige por un ciclo de vida propio el cual consta de 5 fases: Planeación, análisis, diseño, implementación y soporte, las cuales por retroalimentación permiten que el sistema de información lleve un proceso correcto y coherente en todas las fases.

### **2.1.2 Componentes de un SIG**

Los SIG están formados por componentes cuyo elemento principal son los datos y cuyo objetivo esencial es utilizarlos adecuadamente, para que la información que se genere tenga valor para quien la use.

Cada componente del SIG permite manipular los datos con base en sus funciones, para facilitar el manejo de los mismos. En la Tabla 1 se mencionan los componentes de un SIG.

Componente	Descripción
<i>Entrada de Datos</i>	Convierte los datos de su forma existente, a una que pueda ser usada por el SIG.
<i>Administración de Datos</i>	Incluye las funciones necesarias para almacenar y recuperar datos de la base de datos.
<i>Manipulación y Análisis de Datos</i>	Incluye funciones que determinan la información que puede ser generada por el SIG.
<i>Salida de Datos</i>	Representa los reportes que pueden ser generados en forma de mapas, tablas de valores, o archivos de texto, con información de calidad.

**Tabla 1. Componentes de un SIG**

### 2.1.3 Modelos de Datos Espaciales

Existen dos aproximaciones para la representación de componentes espaciales de información geográfica: el modelo vector y el modelo raster<sup>2</sup>. A continuación se describe cada uno de ellos:

- **Modelo Vector**

En este modelo los objetos o condiciones en el mundo real son representados por puntos y líneas que definen sus fronteras o límites. Teniendo en cuenta que éste modelo provee un preciso posicionamiento de rasgos en el espacio, la aproximación usada es precisar específicamente la posición de los *puntos, líneas y polígonos* usados para representar los rasgos de interés. El área del mapa es asumida para ser un espacio coordinado continuo donde una posición puede ser definida como se quiera, para lo cual el modelo asume que las coordenadas de posición son matemáticamente exactas. En efecto, el nivel de precisión está limitado por el número de bits usados para representar un solo valor en el espacio del computador.

- **Modelo Raster**

Consta de un *grid*(reja) regular de celdas cuadradas o rectangulares, en el cual la ubicación de cada celda o píxel está definido por su número de fila y columna, y el valor asignado a la celda indica el valor del atributo que éste representa. Cada celda en un archivo raster se le asigna solo un valor, y los diferentes atributos son almacenados en archivos diferentes.

<sup>2</sup> [http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/vector\\_raster.htm#comparacion](http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/vector_raster.htm#comparacion). Agosto de 2004.

Las operaciones en varios archivos raster cubren la recuperación y el procesamiento de los datos para las posiciones correspondientes de la celda en los diferentes archivos de datos.

La Tabla 2 presenta las ventajas y desventajas de los modelos de datos espaciales vector y raster.

MODELO VECTOR	MODELO RASTER
<b><i>Ventajas</i></b>	
<ul style="list-style-type: none"> <li>▪ Provee una estructura de datos compacta.</li> <li>▪ Provee eficiente codificación de topología y por ende una implementación eficiente de las operaciones que requieren información topológica.</li> <li>▪ Se ajusta más para soportar gráficos que más se aproximan a los mapas dibujados a mano.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Provee una estructura de datos simple.</li> <li>▪ Cubre operaciones que son fácil y eficientemente implementadas.</li> <li>▪ Un formato raster representa de manera eficiente una alta variabilidad espacial.</li> <li>▪ Es un formato requerido para la manipulación y el mejoramiento de imágenes digitales.</li> </ul>
<b><i>Desventajas</i></b>	
<ul style="list-style-type: none"> <li>▪ Tiene una estructura de datos compleja.</li> <li>▪ Cubre operaciones difíciles de implementar.</li> <li>▪ La representación de alta variabilidad espacial es ineficiente.</li> <li>▪ La manipulación y el mejoramiento de imágenes digitales no puede ser realizada efectivamente.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Tiene una estructura de datos menos compacta.</li> <li>▪ Las relaciones topológicas son difíciles de implementar.</li> <li>▪ La salida de gráficos es menos situada estéticamente porque las fronteras tienden a tener una apariencia de bloqueo menor que las líneas lisas de los mapas dibujados a mano.</li> </ul>

**Tabla 2. Comparación entre Modelos de Datos Espaciales**

#### **2.1.4 Bases de Datos Geográficas**

Partiendo de que la información espacial es inherentemente un tipo de dato más complejo para almacenar y manipular, que los datos no espaciales, se tiene que el desarrollo de bases de datos para manejar información espacial es igualmente complejo, para lo cual se debe manejar cierta organización en momento de diseño de la misma.

Actualmente, existe un amplio crecimiento en la cantidad de información geográfica que es recolectada y almacenada en forma digital, para ser adaptada a un análisis y recuperación basadas en computador.

“Una base de datos geográfica es una colección de datos organizados de tal manera que sirvan efectivamente para una o varias aplicaciones SIG.

Esta base de datos comprende la asociación entre sus dos principales componentes: datos espaciales y atributos o datos no espaciales”<sup>3</sup>.

### ***Componentes de una base de datos geográfica***

- Datos espaciales

Llamados atributos, son las características geográficas de los objetos descritos (ubicación, dimensión y forma).

- Datos no espaciales:

También llamados atributos o datos descriptivos, son las características cuantitativas asociadas al objeto que se quiere describir, por lo general son almacenados en tablas.

Para una mayor organización de los datos, se utiliza un Sistema Administrador de Base de Datos (DBMS), el cual permite minimizar la redundancia en el almacenamiento de los datos, por medio de un control centralizado de acceso a los datos, manipulación, integridad y seguridad de la base de datos, para hacer aplicaciones independientes de la forma en la cual se encuentran almacenados los datos.

### ***Aplicación de los conceptos de DBMS en el contexto de SIG***

- Las relaciones entre toda la información espacial y la de atributo son definidas explícitamente.
- Las vistas de los datos son independientes de la forma de almacenamiento de los datos.
- Provee la actualización automatizada de archivos de datos interrelacionados.
- El control central del DBMS provee mayor control de la integridad de la base de datos.

La base de datos geográfica debe almacenar información de las capas geográficas, las cuales son características geográficas de un área que se desea modelar, dichas capas se organizan por temas para facilitar el manejo de la información.

---

<sup>3</sup> [ESRI. 1998]. Fecha de Consulta: Septiembre 2003.

Un mapa puede ser organizado en varias capas tales como división política, hidrología, caminos, contornos o puntos de control, y la idea es que dichas capas sean almacenadas en tablas separadas porque sus atributos son diferentes.

De igual forma, la base de datos almacena entidades, entendiendo por entidad la cosa (objeto, individuo, concepto o evento) que se distingue y sobre el cual se quiere obtener información.

### **2.1.5 Herramientas para generar servicios en línea para SIG**

En el mercado existen algunas soluciones que facilitan la publicación de mapas sobre Internet, y que contienen herramientas que permiten realizar consultas sobre los mismos. A continuación se mencionan algunas de las soluciones que ofrece el mercado, y se describe la más destacada actualmente que es ArcIMS de ESRI.

#### **2.1.5.1. ArcIMS**

ArcIMS es una solución para Sistemas de Información Geográfica la cual permite de manera centralizada, la construcción y la entrega de mapas, datos y herramientas sobre Internet, lo que hace posible compartir información alrededor de ésta red.

ArcIMS permite la creación de mapas y la publicación de éstos en un sitio Web propio, así como la navegación sobre el mapa, por medio de herramientas para consulta sobre el mismo, teniendo en cuenta aspectos de seguridad, los cuales permiten regular el acceso a los servicios que se entregan en Internet.

La arquitectura de ArcIMS consta de componentes del cliente y componentes del servidor, que corren en un entorno distribuido.

#### Componentes al lado del cliente:

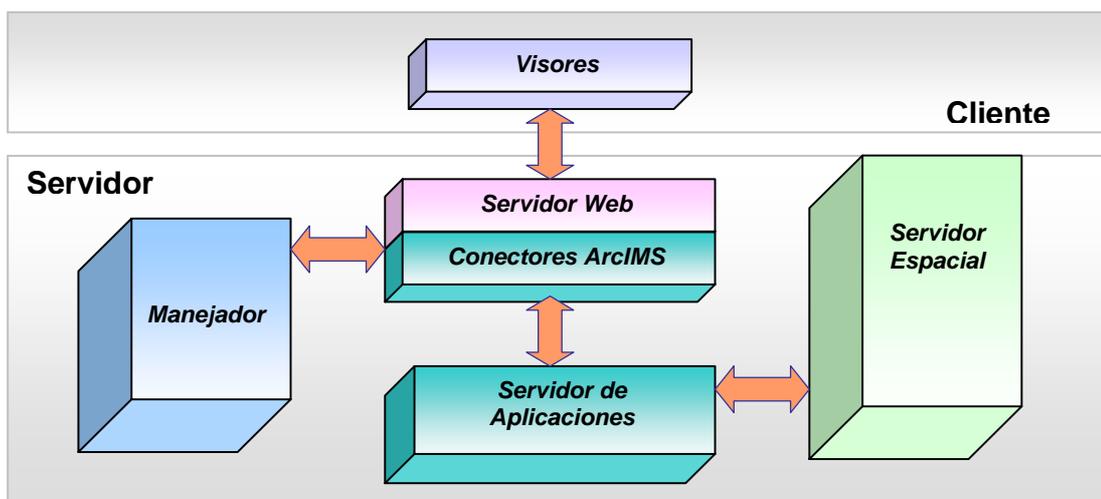
- Visor HTML ArcIMS
- Visores Java™ ArcIMS

#### Componentes al lado del servidor:

- Servidor espacial ArcIMS
- Servidor de aplicaciones ArcIMS

- Conectores del servidor de aplicaciones ArcIMS
- Manejador ArcIMS

La arquitectura también incluye un sistema operativo, un servidor Web, un motor de servlets y navegadores Web al lado del cliente. El procedimiento consiste en que el cliente realiza una solicitud al servidor, quien la procesa y envía la respuesta al cliente, el cual presenta la información que recibe. La Figura 1 presenta un esquema de la arquitectura de ArcIMS.



**Figura 1. Arquitectura de ArcIMS**

- ⇒ El *Servidor Espacial ArcIMS* es el soporte fundamental de ArcIMS, teniendo en cuenta que es éste quien procesa las solicitudes para los mapas y la información referida, soporta 2 procesos en background: El Monitor ArcIMS y el Tasker ArcIMS, los cuales corren como servicios de Windows o como procesos demonio en Unix.
- ⇒ El *Servidor de Aplicaciones ArcIMS* maneja las solicitudes entrantes y rastrea cuáles servicios que están corriendo y en qué Servidores Espaciales ArcIMS, maneja a distancia una solicitud para el Servidor Espacial apropiado, el cual es escrito como una aplicación Java y corre como un servicio de Windows o como un proceso deamon en Unix.
- ⇒ Los *Conectores del Servidor de Aplicaciones ArcIMS* se encargan de conectar el Servidor Web al Servidor de Aplicaciones ArcIMS. El Conector Servlet ArcIMS es un conector estándar usado con ArcIMS, el cual soporta la especificación de la implementación del estándar WMS 1.0.0 del Open GIS Consortium (OGC). La comunicación entre el Servidor Web y el Servidor de Aplicaciones ArcIMS se realiza por medio de ArcXML.

- ⇒ El *Manejador ArcIMS* es una aplicación basada en Web, la cual soporta las 3 tareas principales que se desempeñan en ArcIMS: Creación de mapas, diseño de sitios Web y administración del sitio, las cuales a su vez pueden ser completadas usando respectivamente ArcIMS Author, ArcIMS Designer, y ArcIMS Administrator. El Manejador ArcIMS se debe usar en el caso que se quiera una aplicación basada en Web manejada por wizard para Windows que permita el manejo de sitios de forma remota.
- ⇒ *Los Visores* que incluye ArcIMS son el visor HTML ArcIMS, el visor Java Custom ArcIMS y el visor Estándar de Java ArcIMS. El visor que el cliente utiliza para ver un mapa cuando entra a un sitio Web, se encuentra embebido en el sitio Web, y determina la vista y la funcionalidad del sitio.

### **2.1.5.2. Otras herramientas**

Los programas comerciales de SIG en Internet son desarrollados con el fin de ofrecer mejores herramientas para compartir datos sobre la Web. Se mencionan algunos de los programas que se encuentran disponibles para generar servicios SIG en Internet:

- ⇒ MapObject IMS de ESRI.
- ⇒ MapGuide de AutoDesk.
- ⇒ Geomedia WebMap de Intergraph.
- ⇒ MapXtreme de MapInfo.
- ⇒ Internet Application Server de GE SmallWorld.
- ⇒ Image Web Server de ER Mapper.

## **2.2 Web Services**

Los Web Services son componentes que permiten la comunicación entre aplicaciones ubicadas en diversos puntos geográficos de manera interoperable, por medio del uso de estándares y protocolos abiertos de Internet, los cuales proveen al cliente, acceso a la funcionalidad del servicio sobre la Web de manera eficiente y segura.

### **2.2.1 Definición**

A continuación se mencionan 4 definiciones para Web Services planteadas por las principales organizaciones internacionales que se han enfocado en el tema:

- **W3C**

“Un Web Service es un sistema de software diseñado para la interacción interoperable máquina a máquina sobre una red. Esta tiene una interfaz descrita en un formato procesable por la máquina (específicamente WSDL). Otros sistemas interactúan con el Web Service en una manera preescrita por esta descripción usando mensajes SOAP, típicamente transportado usando HTTP con una serialización XML en conjunto con otros estándares web”<sup>4</sup>.

- **IBM**

“Un Web Service es una interfaz que describe una colección de operaciones que son accesibles en la red a través de mensajería estandarizada XML. Los Web Services satisfacen una tarea específica o un conjunto de tareas. Un Web Service es descrito usando un estándar, noción formal XML, llamado la descripción del servicio, que provee todos los detalles necesarios para interactuar con el servicio, incluyendo formatos de mensaje (que detalla las operaciones), protocolos de transporte y localización.

La naturaleza de la interfaz oculta los detalles de implementación del servicio, así que esta puede ser usada independientemente del hardware o la plataforma de software en la cual esta es implementada, e independientemente del lenguaje de programación en el cual esté escrito. Esto permite y motiva a los Web Services basados en aplicaciones, a que se unan libremente las implementaciones orientadas a componentes por medio de la tecnología. Los Web Services pueden ser usados solos o en conjunto con otros Web Services para realizar una agregación compleja o una transacción de negocios”<sup>5</sup>.

- **Microsoft**

“Un Web Services es una unidad de la lógica de la aplicación que provee datos y servicios a otras aplicaciones. Las aplicaciones acceden a los Web Services vía protocolos web ubicuos y formatos de datos tales como HTTP, HTML y SOAP, sin necesidad de preocuparse acerca de cómo cada servicio está implementado. Los Web Services combinan los mejores aspectos del desarrollo basado en componentes y de la web y son la piedra angular del modelo de programación Microsoft .NET”<sup>6</sup>

---

<sup>4</sup> Web Services. <http://www.w3.org/TR/ws-arch/#whatis> Fecha de Consulta: Octubre 2003

<sup>5</sup> GRAHAM Steve, SIMEINOV Simeon, BOUBEZ Toufic, DAVIS Doug, DANIELS Glen, NAKAMURA Yuichi y NEYAMA Ryo. Building Web Services with Java TM Making sense of XML, SOAP, WSDL, and UDDI. 2002.

<sup>6</sup> Ibidem 5.

“Un Web Service es una aplicación lógica programable, accesible usando protocolos estándares de Internet. Los Web Services combinan los mejores aspectos del desarrollo basado en componentes y de la Web. Como componentes, los web Services representan la caja negra de funcionalidad que puede ser re-usada sin necesidad de conocer como el Web Service es implementado. En contraste con las tecnologías de componentes actuales, los Web Services no son accesibles vía protocolos específicos del modelo de objetos, tal como DCOM, RMI, IIOP. Por su parte ellos son accedidos vía protocolos web y formatos de datos ubicuos, tal como HTTP y XML; además, la interfaz de un Web Service es definida estrictamente en términos de los mensajes que el Web Service acepta y genera.

Los consumidores de un Web Service pueden ser implementados en cualquier plataforma y en cualquier lenguaje de programación tal largos como ellos puedan crear y consumir los mensajes definidos por la interfaz del Web Service”<sup>7</sup>.

- **Sun**

“Web Services son componentes de Software que pueden ser espontáneamente descubiertos, combinados y recombinados para proveer una solución al problema en las solicitudes de los usuarios, el lenguaje Java TM y XML son las tecnologías prominentes para Web Services”<sup>8</sup>

Teniendo en cuenta las anteriores definiciones, un Web Service se puede definir como una unidad lógica que permite la comunicación entre diversas aplicaciones sobre una red, con base en protocolos estándares de Internet como HTTP, SOAP y XML entre otros, que permite ocultar detalles de implementación y con ello proveer seguridad e interoperabilidad.

Los servicios Web están conformados por un conjunto de métodos, los cuales son presentados como una unidad, la cual puede ser accedida por otras aplicaciones, y a su vez formar parte de sistemas distribuidos.

La industria de Tecnologías de Información(IT) por medio de los Web Services, direcciona el desafío fundamental de la computación distribuida, que ha tenido a través de décadas, a través de la localización y acceso a sistemas remotos.

---

<sup>7</sup> Ibidem 5.

<sup>8</sup> Ibidem 3.

### 2.2.2 Estados de un Web Service

Un Web Service puede pasar por diferentes estados, los cuales se explican en la Tabla 3.

Estado	Descripción
<i>Descrito</i>	Por medio de un lenguaje de descripción para el servicio, como el lenguaje WSDL (Web Service Description Language)
<i>Publicado</i>	En un registro de servicios, utilizando el método de registro UDDI (Universal Description, Discovery and Integration).
<i>Descubierto</i>	A través de mecanismos estándar, ya sea en tiempo de diseño o en tiempo de ejecución, al enviar peticiones al Registro y recibir detalles de ligamiento ( <i>binding</i> ) del servicio que se ajusta a los parámetros de la búsqueda.
<i>Invocado</i>	A través de un API declarado usualmente sobre una red, se utiliza la información contenida en los detalles de ligamiento de la descripción del servicio.
<i>Compuesto</i>	Con otros servicios para integrar servicios y aplicaciones nuevas mas integrales.

**Tabla 3. Estados de un Web Service**

### 2.2.3 Características de los Web Services

- ⇒ Forman un ambiente distribuido ya que los objetos pueden ser accedidos remotamente por medio de interfaces y proxies, y las operaciones pueden ser desempeñadas en ellas.
- ⇒ Usan un modelo de 3 capas, en el cual se definen 3 roles: El rol del proveedor del servicio, el rol del consumidor del servicio, y el rol del corredor del servicio. Estos roles funcionan de la siguiente manera: Los proveedores pueden registrar sus servicios con los corredores, y los consumidores pueden preguntar a los corredores por información en los servicios.
- ⇒ Sus componentes están libremente ligados, lo cual permite a un cliente consultar el corredor en tiempo de ejecución, obtener la interfaz para un servicio, y enlazarse a esta, sin tener que ingresar un URL o cualquier nombre de método. Esto hace a una red de Web Services, auto reparable, porque si un servicio se baja, los consumidores pueden consultar los corredores para otros servicios que puedan desempeñar la misma función. La importancia radica en que se tienen varios nodos de corredores que aseguran que un corredor está siempre disponible.

- ⇒ Para la comunicación se rigen por los estándares abiertos de la Web tales como: TCP/IP, HTTP y XML. Los protocolos de alto nivel para llamado a procedimientos, descripción, publicación, búsqueda y unión entre otros, están basados en gramática XML. Lo que asegura la interoperabilidad y la independencia de vendedores, es el uso de estándares abiertos que son ampliamente soportados sobre diversas plataformas y que tienden a ser permitidos a través de firewalls.

#### **2.2.4 Componentes de un Web Service**

##### *Servicio*

Es la función que se ofrece a los solicitantes que cumplen con los requisitos mínimos que especifica el proveedor del servicio. El servicio es descrito por medio de un lenguaje de descripción de servicio, y se implementa sobre una plataforma la cual es accesible en la red. Un servicio publica en un registro las políticas de uso.

##### *Registro de Servicios*

Es un ente donde se encuentran las descripciones de servicios, éste ente puede ser consultado tanto por los proveedores de servicios para publicar sus servicios, como por los solicitantes quienes encuentran los servicios y características para poder utilizar los servicios previamente publicados por el proveedor.

##### *Proveedor*

El proveedor es quien publica un servicio. Se puede ver desde 2 puntos de vista: desde el punto de vista de arquitectura, es la plataforma que provee el servicio y desde el punto de vista comercial, es quien presta el servicio.

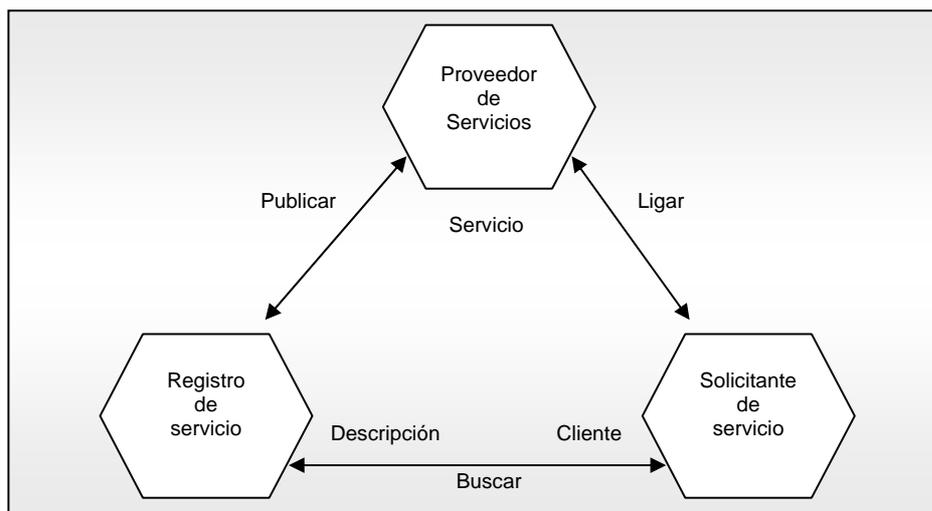
##### *Consumidor*

El consumidor es quien utiliza el servicio. Se puede ver desde 2 puntos de vista: desde el punto de vista de arquitectura, es la aplicación o cliente que busca e invoca un servicio y desde el punto de vista comercial, es la empresa que requiere cierto servicio.

## 2.2.5 Objetivos de los Web Services

- ⇒ *Interoperabilidad* por medio de la utilización de estándares: XML, SOAP, WSDL y UDDI entre otros.
- ⇒ *Independencia* tanto del lenguaje como de la plataforma, por medio de la separación de especificación de la implementación
- ⇒ *Acoplamiento débil* debido al manejo de sistemas basados en mensajes en los cuales existen interacciones síncronas y asíncronas
- ⇒ *Acceso a través de Internet* lo cual permite un control no centralizado, por otro lado se tiene la utilización de protocolos establecidos y las consideraciones de seguridad.
- ⇒ *Modularidad y Reusabilidad* de servicios debido al uso de estándares.
- ⇒ *Escalabilidad* teniendo en cuenta que se trata de aplicaciones uno-a-uno frente a uno-a-muchos.

## 2.2.6 Arquitectura



**Figura 2. Arquitectura de Web Services**

La Figura 2<sup>9</sup> describe la arquitectura de Web Services, teniendo en cuenta las operaciones que desempeñan y los componentes involucrados.

Las operaciones básicas de un Web Service son:

<sup>9</sup> <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/> Fecha de Consulta: Octubre de 2003.

### Publicar/Cancelar:

Los proveedores de servicios publican (publicitan) la disponibilidad de su servicio comercial (e-business) a uno o más registros de servicios, o cancelan la publicación de su servicio.

### Búsqueda:

Los solicitantes de servicios interactúan con uno o más registros de servicios para descubrir un conjunto de servicios comerciales con los que pueden interactuar para encontrar una solución.

### Ligar, Unir (Bind):

Los solicitantes de servicios negocian con los proveedores de servicios para acceder e invocar servicios comerciales (e-business) .

## **2.2.7 Protocolos y estándares de los Web Services**

### **2.2.7.1 Hypertext Transfer Transport Protocol (HTTP)**

HTTP es un protocolo que provee la ligereza y la velocidad necesarias para un sistema de información de colaboración distribuido. Es un protocolo genérico sin estado orientado a objetos, se puede utilizar para muchas tareas similares tales como servidores de nombres, y sistemas orientados a objetos distribuidos, ampliando los comandos, o los "métodos", usados.

Una característica que ofrece HTTP es la negociación de la representación de datos, permitiendo que los sistemas sean construidos independientemente del desarrollo de nuevas representaciones avanzadas.

#### **2.2.7.1.1 Métodos**

El campo método en HTTP indica el método que se realizará en el objeto identificado por el URL. El método GET, es soportado por una lista de otros métodos aceptables por los objetos retornados en el response a cualquiera de estas dos peticiones.

A continuación se presenta la lista de métodos, cuyos nombres son case sensitive: GET, HEAD, CHECKOUT, SHOWMETHOD, PUT, DELETE, POST, LINK, UNLINK, CHECKIN, TEXTSEARCH, SPACEJUMP Y SEARCH de los cuales los más directamente usados son GET y POST:

## GET

Devuelve cualquiera de los datos identificados por el URI, donde el URI se refiere a los datos producidos en el proceso, o un script que puede correr como un proceso, estos serán los datos que serán devueltos, no el texto fuente del script o proceso.

## POST

Crea un nuevo objeto con un link a un objeto especificado. El campo de id del mensaje de un nuevo objeto debe ser dado por el cliente o sino será dado por el servidor. Una URL será asignada por el servidor y devuelta al cliente. El nuevo documento es el elemento de datos de la petición.

### 2.2.7.1.2 Request

El Request se envía con una primera línea que contiene el método que se aplicará al objeto solicitado, el identificador del objeto, y la versión del protocolo en uso, seguido por la información adicional codificada en el estilo de encabezado RFC822. El formato de la petición es el siguiente:

Request	=	SimpleRequest   FullRequest
SimpleRequest	=	GET <uri> CrLf
FullRequest	=	Method URI ProtocolVersion CrLf [*<HTRQ Header>] [<CrLf> <data>]
<Method>	=	<InitialAlpha>
ProtocolVersion	=	HTTP/1.0
uri	=	<as defined in URL spec>
<HTRQ Header>	=	<Fieldname> : <Value> <CrLf>
<data>	=	MIME-conforming-message

### 2.2.7.2. Universal Discovery Description Integration (UDDI)

Es un protocolo de descripción universal, descubrimiento e integración, considerado uno de los principales bloques de construcción requeridos para Web Services exitosos. UDDI crea un estándar de plataforma interoperable

que permite a una compañía el uso rápido, fácil, y dinámico de hallazgo y uso de Web Services sobre Internet.

UDDI también permite registro operacional que se mantiene para diversos propósitos en diversos contextos.

El proyecto de UDDI se aprovecha de estándares mundiales del consorcio (W3C) y los estándares del Internet Engineering Task Force (IETF), tales como Extensible Markup Language (XML), y los protocolos de HTTP y el Domain Name System (DNS). Además, a través de las características de programación de la plataforma son tratadas adoptando versiones tempranas del Simple Object Access Protocol (SOAP) conocidas como las especificaciones de la mensajería del protocolo de XML encontradas en sitio W3C.

La especificación UDDI define una estructura de datos y un API, que describen la manera como se construye conceptualmente un registro de Web Services, cómo se accede, consulta y modifica por los diseñadores del servicio, implementadores y clientes.

Los negocios de todos los tamaños pueden beneficiarse de UDDI, porque la especificación trata los problemas que limitan el crecimiento y las sinergias del comercio de Bussiness to Bussiness (B2B).

Los siguientes son los problemas que la especificación de UDDI puede ayudar a solucionar:

- ✓ Permite que las organizaciones descubran rápidamente las reglas del negocio de millones que están actualmente en línea.
- ✓ Permite definir cómo se activa el comercio lo cual se conduce una vez que se descubra el negocio preferido.

Las ventajas inmediatas de UDDI para los negocios son las siguientes:

- ✓ Alcanzar a nuevos clientes y expandir ofertas.
- ✓ Extender el mercado que se obtiene.
- ✓ Incrementar el acceso de los clientes actuales.
- ✓ Solucionar las necesidades de los clientes quitando barreras que permitan la rápida participación en la economía global de Internet.
- ✓ Describir sus servicios y procesos del negocio programáticamente en un ambiente simple, abierto y seguro.
- ✓ Usar un sistema de los protocolos que permiten a los negocios invocar servicios sobre Internet para adicionar valor a sus clientes preferidos.

### 2.2.7.2.1. Páginas Blancas, Amarillas y Verdes

Se distinguen tres tipos de información que será encontrada en el registro acerca de negocios, estos tipos de información están representados por páginas blancas, amarillas y verdes las cuales se describen en la Tabla 4.

Páginas	Descripción
<i>Blancas</i>	Dan información de los negocios. Si necesita saber el nombre de un negocio, se puede encontrar detalles del contacto que representan datos del mundo real. Con esta información, se puede construir libros de direcciones automáticamente y listas de clientes, entre otros.
<i>Amarillas</i>	Proveen capacidades taxonómicas: un negocio puede ser clasificado de acuerdo a sus actividades. Un negocio simple puede pertenecer a varias categorías, y los usuarios pueden buscar un registro para negocios en una categoría dada.
<i>Verdes</i>	Contienen información técnica acerca de los servicios proveídos por el negocio. Esta información debe incluir interfaces implementadas y puntos de acceso a la red, entre otros.

**Tabla 4. Páginas Blancas, Amarillas y Verdes (UDDI)**

### 2.2.7.2.2. Estructura de Datos de UDDI

#### Universal:

(Universal) el extensible tModel basado en una arquitectura que permite cualquier identificador o taxonomía se adiciona al registro, así todos los tipos de objetos diferentes pueden ser registrados y descritos.

#### Description:

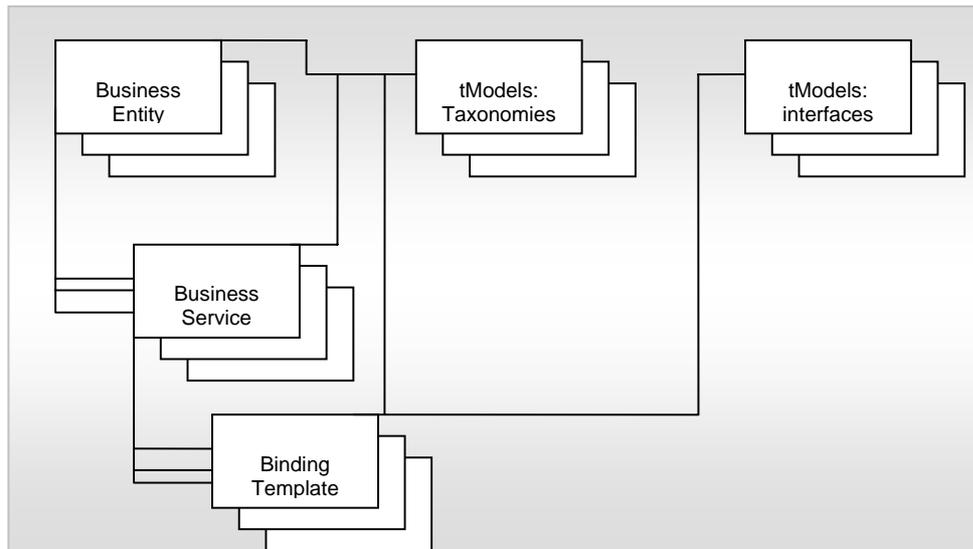
(Descripción) gracias a los mecanismos de taxonomía, clasificaciones y a la posibilidad de agregar URL externas que referencian documentos WSDL, todos los servicios pueden ser descritos completamente y con precisión.

#### Discovery:

(Descubrimiento) Un registro UDDI provee algoritmos de búsqueda y drill-down que permiten a los consumidores del servicio que encuentren negocios, implementaciones e interfaces usando un arreglo de criterio extensible (tModel-based).

Integration:

(Integración) debido a que todos los servicios pueden ser publicados en el registro, totalmente descritos y descubiertos dinámicamente, ellos pueden ser combinados dentro de una aplicación consumidora sin necesidad de información externa. (Figura 3).



**Figura 3. Estructura de Datos UDDI**

**2.2.7.2.3. API UDDI**

El API UDDI se divide en tres secciones principales:

⇒ Búsqueda

El API de búsqueda define métodos que permite al consumidor de un servicio buscar por negocios, servicios o tModels de acuerdo a los siguientes criterios de búsqueda: Categoría taxonómica, especificación tModel o identificadores tModel

⇒ Drill-down

Permite a los consumidores de servicios obtener detalles de una entidad específica.

## ⇒ Publicación

Permite a los diseñadores del servicio y a los implementadores publicar y borrar interfaces, negocios, servicios, bindings , entre otros.

### **2.2.7.3. Web Service Description Language (WSDL)**

WSDL es un estándar planteado por la W3C, el cual se define como un formato XML para describir servicios en la red como un conjunto de puntos finales, los cuales operan en mensajes que contienen cualquier información ya sea orientada a procedimiento o a documento. En WSDL se describen de manera abstracta las operaciones y los mensajes, los cuales a su vez se unen para un protocolo de red y formato de mensaje específicos, con el objetivo de definir un punto final. Por su parte, estos puntos finales se combinan en puntos finales abstractos, los cuales se denominan servicios.

Este estándar fue creado con el propósito de describir la sintaxis para la invocación técnica de un Web Service, y con ello permitir a las aplicaciones describir a otras aplicaciones las reglas para interactuar y dialogar entre ellas. Además WSDL permite automatizar la generación de servidores proxy para los Web Services de forma independiente de lenguaje y de plataforma.

Una descripción WSDL básicamente describe las tres propiedades fundamentales de un Web Service:

- Qué hace un servicio: Las operaciones o métodos que el servicio provee.
- Cómo es accedido un servicio: Los detalles de formatos de datos y de los protocolos necesarios para acceder las operaciones del servicio.
- Dónde es localizado un servicio: Los detalles de la dirección de red del protocolo específico, tal como un URL.

Un documento WSDL es el contrato que existe entre el cliente y el servidor, en el cual se establecen las reglas de comunicación, se definen servicios como colecciones de puntos finales o puertos de una red, por lo cual dicha definición abstracta de puntos finales y mensajes, es separada de su concreto despliegue de red o ligamentos de los formatos de datos, y cuya independencia permite la reutilización de definiciones abstractas tales como los mensajes.

Los elementos que usa un documento WSDL en la definición de servicios de red se presentan en la Tabla 5:

<b>Elemento</b>	<b>Descripción</b>
<i>Tipos</i>	Define una colección de todos los tipos de datos que son usados en el Web Service como referencia para varios elementos que hacen parte del mensaje.
<i>Mensaje</i>	Define un conjunto de parámetros referidos por las firmas de métodos u operaciones. Es una definición abstracta y tipada de los datos que van a ser comunicados.
<i>Operación</i>	Descripción abstracta de una acción soportada por el servicio.
<i>Tipo de Puerto</i>	Conjunto de operaciones abstractas que son soportadas por uno o más puntos finales, donde cada operación se refiere a un mensaje de entrada y los mensajes de salida.
<i>Enlace (Binding)</i>	Especificación de un protocolo concreto y de formato de dato para las operaciones y mensajes definidos por un tipo de puerto dado.
<i>Puerto</i>	Punto final de comunicación definido como una combinación de un binding y una dirección de red.
<i>Servicio</i>	Usado para agregar un conjunto de puertos relacionados.

**Tabla 5. Elementos de un documento WSDL**

Microsoft plantea una división para un documento WSDL: “Este documento se puede dividir en dos grupos de secciones. El grupo superior está compuesto por las definiciones abstractas, mientras que el inferior lo forman descripciones concretas.”<sup>10</sup> (Figura 4).

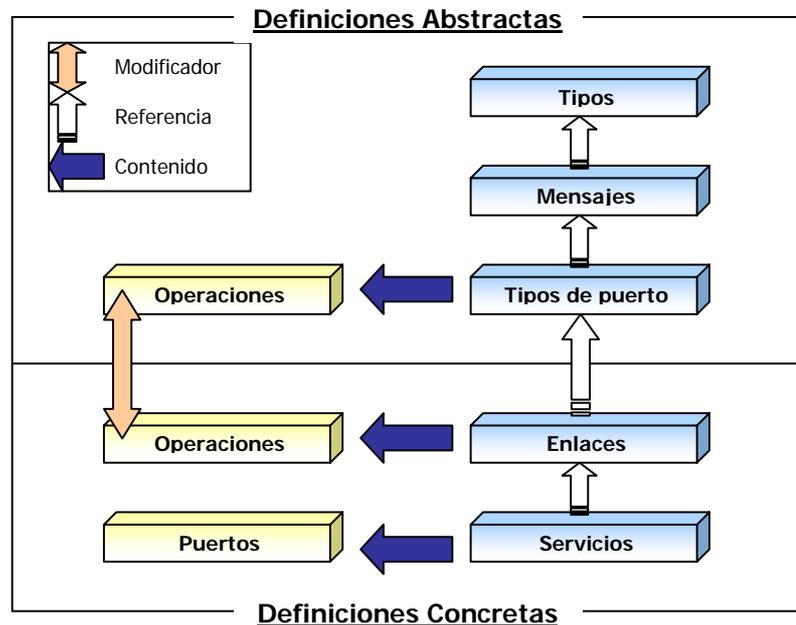
Las secciones abstractas definen los mensajes SOAP independientemente de la plataforma y del lenguaje, porque no contienen elementos específicos de lenguaje y de equipo, lo que permite definir servicios que implementen diferentes sitios Web. Por su parte, en las secciones concretas se definen las cuestiones que son específicas al sitio, tales como la serialización.

La estructura del documento se relaciona de la siguiente manera: la sección de mensajes utiliza definiciones de la sección de tipos; a su vez la de tipos de puerto usa definiciones de la sección de mensajes; la sección de enlaces hace referencia a la sección de tipos de puerto y la de servicios a la sección de enlaces. Tanto la sección de tipos de puerto como la de enlaces, contienen elementos de operación mientras que la sección de servicios incluye elementos de puerto. En la sección de enlaces, los elementos de operación se encargan de describir o modificar a fondo los elementos de operación de la sección de tipos de puerto.

---

<sup>10</sup>

[http://www.microsoft.com/spanish/msdn/articulos/archivo/091101/voices/wsdlexplained.asp#wsdlexplained\\_topic11](http://www.microsoft.com/spanish/msdn/articulos/archivo/091101/voices/wsdlexplained.asp#wsdlexplained_topic11).  
Fecha de consulta: Noviembre de 2003.



**Figura 4. Definiciones Abstractas WSDL**

#### 2.2.7.4. Simple Object Access Protocol (SOAP)

SOAP es un protocolo planteado por la W3C para el intercambio de información estructurada en un ambiente descentralizado y distribuido, el cual utiliza tecnologías XML para definir una estructura de mensajería extensible que provea un mensaje construido de tal manera, que pueda ser intercambiado sobre diferentes protocolos, para lo cual, dicha estructura ha sido diseñada para ser independiente de cualquier modelo de programación específico y de otras semánticas específicas de implementación.

Microsoft plantea la siguiente definición para SOAP: “Es un protocolo elaborado para facilitar la llamada remota de funciones a través de Internet, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web.”<sup>11</sup>

Las solicitudes SOAP se pueden hacer en tres estándares: GET, POST y SOAP. Las respuestas siempre son en XML sin importar la manera como se haya realizado la solicitud.

<sup>11</sup> <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art51.asp> Fecha de Consulta: Febrero de 2004.

#### **2.2.7.4.1. Ventajas de SOAP**

El protocolo SOAP posee varias ventajas en comparación con otras formas utilizadas para el llamado de funciones de manera remota como CORBA, DCOM, o TCP/IP, tales como:

- Es un estándar de la industria, el cual es adoptado por varias empresas.
- Es independiente del sistema operativo y del procesador.
- Es fácil de implementar, probar y usar.
- Los datos y las funciones se describen en XML, lo que provee solidez al protocolo.
- Utiliza los mismos estándares de la Web para la mayoría de las funciones, tales como: la comunicación que se hace mediante HTTP con paquetes virtualmente iguales; los protocolos de autenticación y encriptación son los mismos; el mantenimiento de estado se hace de la misma forma; se implementa normalmente por el propio servidor Web.
- Permite utilizarse en forma anónima o con autenticación (user/password).

#### **2.2.7.4.2. Objetivos de SOAP**

- Instaurar un protocolo de invocación de servicios remotos estándar, basado en protocolos estándares de Internet tales como HTTP para la transmisión y XML para la codificación de datos.
- Conseguir la independencia tanto de plataforma y lenguaje de desarrollo, como de implementación es decir de modelo de objetos.

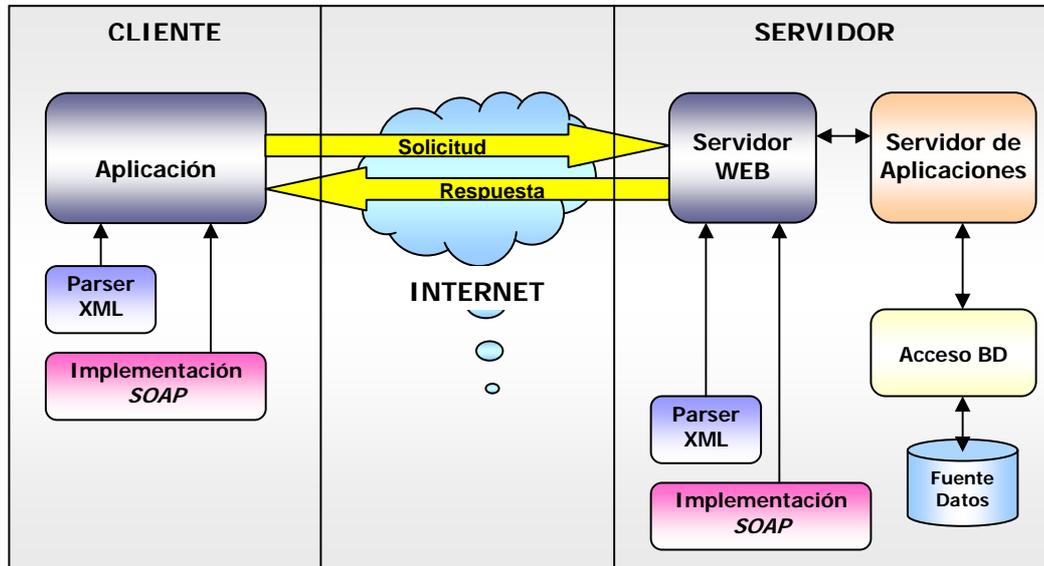
#### **2.2.7.4.3. Funcionamiento de SOAP**

Con base en la especificación de SOAP, las aplicaciones deben ser independientes del lenguaje de desarrollo, por lo cual las aplicaciones cliente y servidor pueden estar escritas en lenguajes como Java, Visual Basic, HTML, DHTML, entre otros disponibles. La importancia radica en que se recomienda tener alguna implementación de SOAP y enlazar sus librerías con la aplicación, con el objetivo de la reutilización de código. Por medio del protocolo HTTP, las peticiones emplean el método POST para la comunicación entre el cliente y el servidor.

SOAP define el conjunto formal de convenciones que rigen el formato y las reglas de procesamiento de un mensaje SOAP. Dichas convenciones incluyen las interacciones entre los nodos SOAP, generando y aceptando mensajes SOAP con el propósito de intercambiar información por un path de un mensaje SOAP, el cual se define como el conjunto de nodos SOAP a

través de los cuales pasa un solo mensaje SOAP, éste incluye el emisor SOAP inicial, cero o más intermediarios SOAP y un último receptor SOAP.

El esquema de funcionamiento de SOAP se presenta en la Figura 5.<sup>12</sup>



**Figura 5. Esquema de funcionamiento de SOAP**

#### 2.2.7.4.4. Factores en escenarios distribuidos

SOAP provee los siguientes factores en los escenarios de computación distribuida:

- Un mecanismo para definir la unidad de comunicación.
- Un mecanismo para el manejo de errores.
- Un mecanismo de extensibilidad en el cual la evolución no está bloqueada.
- Un mecanismo flexible para la representación de datos que permite el intercambio de datos serializados en algún formato.
- Una convención para representar Remote Procedure Calls (RPCs) y respuestas como mensajes SOAP.
- Un mecanismo de "binding" para mensajes SOAP a HTTP.
- Una aproximación a un documento central para reflejar modelos de intercambio de documentos de manera más natural para interacciones de negocios.

<sup>12</sup> <http://www.revista.unam.mx/vol.3/num1/art3/img01.html> Fecha de Consulta: Marzo de 2004.

#### **2.2.7.4.5. Mensaje SOAP**

Un mensaje SOAP es una manera fundamental de transmisión entre nodos (Personificación de la lógica de procesamiento necesaria para transmitir, recibir, procesar y/o retransmitir un mensaje, responsable de hacer cumplir las reglas que rige el intercambio de mensajes), de un emisor (Nodo que transmite un mensaje) a un receptor(Nodo que acepta un mensaje).

Los mensajes SOAP fueron diseñados para ser combinados por aplicaciones, para implementar patrones de interacción mas complejos y extenderse de request/response a múltiples intercambios conversacionales *back-and-forth*.

Un mensaje SOAP consta de un envelope (sobre) obligatorio que envuelve cualquier número de headers opcionales y un body obligatorio.

En SOAP toda la información es empaquetada en un mensaje claramente identificable. Teniendo en cuenta que este protocolo provee un modelo de procesamiento distribuido, un mensaje originado en un emisor inicial, envía éste mensaje a un receptor por medio de cero o más intermediarios. En el procesamiento de un mensaje, un nodo es enviado para actuar en uno o más roles, cada uno identificado por un nombre de rol.

#### **2.2.7.5. Extensible Markup Language (XML)**

XML consiste en una serie de reglas, pautas, convenciones, para planificar formatos texto para tales datos, de manera que produzcan archivos que sean fácilmente generados y leídos (por un ordenador) que son inequívocos, y que evitan dificultades comunes como la falta de extensibilidad, falta de soporte para la internacionalización o localización, y la dependencia de una determinada plataforma.

Al igual que HTML, XML utiliza *tags* -etiquetas- (palabras entre corchetes agudos: '<' y '>') y *atributos* (de la forma name="valor"), mientras que HTML especifica lo que cada etiqueta y atributo significan (y frecuentemente la apariencia que presentará en un navegador el texto que hay entre ellos) XML usa las etiquetas sólo para delimitar piezas de datos, y deja la interpretación de los datos, completamente, a la aplicación que los lee.

Los archivos XML son archivos de texto, pero son más difíciles de leer por los humanos que los archivos HTML. Son archivos de texto, porque permiten a los expertos depurar errores en las aplicaciones, más fácilmente, y en casos de emergencia, pueden usar un simple editor de textos para arreglar un archivo XML estropeado.

Pero las reglas para los archivos XML son más estrictas que para los archivos HTML. El olvido de una etiqueta, o un atributo sin comillas, hacen que el archivo sea inservible; mientras que en HTML, a menudo, tales prácticas son explícitamente permitidas o al menos toleradas. Está escrito en la especificación oficial de XML: No le está permitido a las aplicaciones intentar justificar al creador de un archivo XML dañado; si el archivo está dañado, la aplicación debe detenerse inmediatamente y emitir un error.

Existe *XML 1.0*, la especificación que define cuales son las "tags" (etiquetas) y "atributos", pero alrededor de XML 1.0 hay una creciente serie de módulos opcionales que ofrecen colecciones de etiquetas y atributos, o pautas para especificar tareas.

Existe, por ejemplo, *Xlink* que describe una manera estándar de añadir hiper-enlaces a un archivo XML. *XPointer* y *XFragments* son sintaxis para apuntar a partes de un documento XML. (Un *Xpointer* es parecido a una URL, pero en vez de apuntar a documentos en la Web, apunta a fragmentos de datos en un archivo XML.) CSS, el lenguaje de hojas de estilo, se puede aplicar a XML igual que a HTML. XSL es el lenguaje avanzado para explicitar hojas de estilo. Está basado en XSLT, un lenguaje de transformación a menudo útil también fuera de XSL, para reordenar, añadir o borrar etiquetas y atributos. El *DOM* es una serie de funciones estándar llamadas para manipular archivos XML (y HTML) desde un lenguaje de programación.

*XML Namespaces* es una especificación que describe como se puede asociar una URL (dirección en la Web) con cada etiqueta y atributo en un documento XML, si bien, para qué se utiliza la URL depende de la aplicación que lea la URL. (RDF, el estándar del W3C para metadatos, lo usa para enlazar cada metadato a un archivo definiendo el tipo de ese metadato.) *XML Schemas 1 y 2* ayuda a los desarrolladores a definir precisamente sus propios formatos basados en XML. Hay muchos más módulos y herramientas disponibles o en desarrollo.

Elijiendo XML como base para algún proyecto se tiene a disposición una gran y creciente comunidad de herramientas. Puesto que XML es una tecnología W3C, es gratis, se puede construir *software* propio para ello y sin pagar nada a nadie. La amplia y creciente distribución significa que se está ligado a un único vendedor.

#### **2.2.7.6. XML Stylesheets Language Transformation (XSLT)**

XSLT es un lenguaje de transformación basado en hojas de estilo, definido por la W3C, el cual es utilizado para convertir documentos XML en otros

documentos XML con base en unos DTD(Data Transformation Definition) definidos. Los programas XSLT son escritos en XML y por lo general requiere un procesador de hojas de estilo para procesarlas aplicándolas a un fichero XML.

El estilo de programación está basado en 2 factores básicos:

- Programación soportada por reglas, lo que indica que si ocurre algo en la entrada, se genera algo en la salida.
- No hay situaciones secundarias, lo que indica que una instrucción debe realizar la misma acción sin importar la ruta para llegar a ella, lo cual implica también que no existen variables globales.

Las hojas de estilo son la única manera para adaptar un contenido descrito con XML a diferentes clientes, y también es la mejor forma para procesar documentos XML, ya que lo que permiten es la independencia de la información de su manera de presentación, por medio de las transformaciones que se requieren para mostrar de la mejor manera el contenido al lado del cliente, dependiendo de los requerimientos del mismo.

Una transformación en el lenguaje XSLT es expresado como un documento XML bien formado, conforme a los Namespaces que recomienda XML, los cuales deben incluir los elementos definidos y no definidos por XSLT.

Una transformación expresada en XSLT define reglas para transformar un árbol fuente en un árbol resultado, dicha transformación es alcanzada por la asociación de patrones con plantillas. Un patrón es igualado contra elementos en el árbol fuente y una plantilla es instanciada para crear parte del árbol resultante, el cual a su vez es separado del árbol fuente.

Hay que tener en cuenta que la estructura del árbol resultante puede ser totalmente diferente a la estructura del árbol fuente que lo genera, ya que en el proceso de construcción del árbol resultante, los elementos del árbol fuente pueden ser filtrados y recuperados, y además pueden ser adicionadas estructuras arbitrarias.

Una transformación expresada en XSLT es llamada XSLT stylesheet, debido a que en el caso cuando XSLT es transformado en vocabulario formateado en XSL, dicha transformación funciona como una hoja de estilo.

XSLT es considerada la parte de XML que permite transformar archivos XML a HTML, a TEXT o a cualquier otro lenguaje de marcas, para lo cual se requiere un motor XSLT. Un motor XSLT se define como el elemento que realmente realiza la transformación, los cuales pueden estar: dentro de los navegadores, en los servidores o independientes.

XSLT provee dos mecanismos para combinar hojas de estilos:

- Un mecanismo de importación que permite a las hojas de estilos no hacerle caso el uno al otro.
- Un mecanismo de inclusión que permite a las hojas de estilos ser combinadas sin cambiar la semántica.

XSLT utiliza expresiones con los siguientes propósitos:

- Seleccionar los nodos a procesar.
- Especificar las condiciones para las diferentes maneras de procesar un nodo.
- Generar el texto a ser insertado en el árbol resultante.

#### **2.2.7.6.1. Proceso de transformación**

Teniendo en cuenta que las hojas de estilo XSLT realizan la transformación del documento utilizando una o varias reglas de plantilla, se tiene que dichas reglas ligadas al documento a transformar, son las que alimentan a un procesador XSLT, el cual es quien realiza las transformaciones colocando el resultado ya sea en un archivo de salida o en algún dispositivo de presentación.

El modelo de procesamiento de XSLT consiste en que se tiene una lista de nodos fuente, que es procesada para un fragmento de un árbol resultado. Un árbol resultado es construido por el procesamiento de una lista que contiene solamente el nodo raíz.

Una lista de nodos fuente es procesada por adición de la estructura del árbol resultante creada por el procesamiento de cada uno de los miembros de la lista en orden. Un nodo es procesado por el descubrimiento de todas las reglas de plantilla con patrones que igualan al mismo, y escoge la mejor entre ellas, luego la regla de plantilla seleccionada es instanciada con dicho nodo como el actual y con la lista de los fuente como la lista de actuales. Una plantilla típicamente contiene instrucciones que seleccionan una lista adicional de nodos fuente para procesar.

El proceso de igualación, instalación y selección es continuado recursivamente hasta que no haya nuevos nodos seleccionados para procesamiento.

Es importante resaltar que la asociación de XML con XSLT, permite separar el contenido de la presentación, lo cual se ve reflejado en un aumento de la productividad, ya que dicha independencia trae consigo factores esenciales como modularidad y reutilización.

Por su parte, los navegadores son quienes permiten finalmente visualizar de una manera adecuada páginas XML, para lo cual se debe indicar a éste la ubicación de la XSLT.

Por lo general se suele usar las hojas de estilo dentro de un entorno de publicación tal como Cocoon, IBM Transcoding Publisher y AxKit entre otros. Dicho entorno de publicación permite mantener sitios completamente basados en XML, y con ello generar páginas en diversos formatos a partir de ellos. Por su parte, los procesadores de hojas de estilo, también pueden usarse como librerías de clases o desde otros lenguajes.

La funcionalidad de XML/XSLT puede generar contenido dinámico, en el sentido que se evalúan las entradas y con base en ellas y/o en los atributos, se genera el contenido adecuado, para lo cual se le pasan parámetros a las hojas por medio de *xsl:param*.

Un documento XML puede tener varias hojas de estilo asociadas, pero es el sistema de publicación quien selecciona la hoja de estilo, con base en el navegador que tenga el cliente.

### **2.3. Servicios en Línea para SIG**

Los servicios en línea, independientemente del área de enfoque deben tener una característica esencial y es la interoperabilidad, la cual se logra a partir del uso de estándares para todas las fases del servicio. Se entiende por *interoperabilidad* “la habilidad de un sistema, o componentes de un sistema para proveer portabilidad de la información y control en el proceso cooperativo entre aplicaciones”(Bordie, 1992).

Se distinguen 2 clases de interoperabilidad: Para un programa la interoperabilidad significa la habilidad para utilizar un rango de formatos de datos; mientras que para un conjunto de datos, la interoperabilidad de un programa significa que dicho conjunto puede ser usado por diferentes tipos de programas. Una base de datos interoperable se refiere al nivel de interoperabilidad de los datos. Esto puede ser usado por diferentes tipos de programas y aplicaciones. Las bases de datos interoperables permiten a los usuarios solicitar e integrar datos de manera sencilla independientemente del lugar de almacenamiento de la base de datos: localmente o remotamente.

Particularmente, en el contexto de las aplicaciones geográficas, la interoperabilidad de los datos de fuentes heterogéneas es muy importante, debido a la existencia de grandes volúmenes de datos especiales de

diferentes formatos geográficos y además porque la demanda para reutilizar dichos datos espaciales crece considerablemente.

### **2.3.1. Seguridad en los Servicios en Línea**

#### **2.3.1.1. Seguridad a nivel de Web**

Teniendo en cuenta la importancia de la estandarización en todos los factores que influyen en la generación de servicios Web, se mencionan a continuación los objetivos de seguridad descritos por el World Wide Web Consortium en su documento de especificación de requisitos:

La Arquitectura Web Service (Web Service Architecture o WSA) debe proporcionar un entorno seguro para los procesos en línea (onlines processes).

Los factores críticos de éxito identificados son:

- Construcción de un modelo de amenazas para los Web services (Web services Threat Model) basado en un análisis exhaustivo de las amenazas actuales y futuras a las que estarán sujetos tanto los Puntos Finales (EndPoints) como los canales de comunicación.
- Establecimiento de un conjunto de políticas de seguridad para los Web services (Web services Security Policy) que neutralice o mitigue las amenazas de seguridad.
- Construcción de un modelo de seguridad (Web services Security Model) que integre las políticas de seguridad, debe proporcionar mecanismos de confidencialidad.
- Marco de referencia de seguridad que proporcione mecanismos:
  - De integridad de datos
  - De no repudio en origen y recepción
  - Para expresar la política de seguridad
  - Que permitan acceder a la política de seguridad definida por un Web service
  - De auditoría/trazabilidad (auditing)

Cuando el Web service proporcione mecanismos de seguridad acordes a lo expresado anteriormente, proporcionará medios que habiliten la administración anonimato de una o varias partes que participen en la comunicación

Como conclusión de esta parte de seguridad en Web Services, se determina que existen varias alternativas para garantizar un entorno seguro y confiable, basado en la federación de servicios, en el cual las relaciones de

confianza, las políticas de seguridad, la identidad digital, los procesos de negocio y los beneficios se gestionarán, administrarán y operarán de forma automática.

### **2.3.1.2. Seguridad a nivel de Negocio**

Los dos conceptos básicos cuando se habla de seguridad en servicios Web son Autenticación y Autorización, los cuales se definen a continuación.<sup>13</sup>

#### **AUTENTICACIÓN:**

Es el proceso de obtener credenciales de identificación, tales como un nombre y una contraseña de un usuario, y validar estas credenciales contra alguna autoridad tal como una base de datos. Si las credenciales son válidas, la identidad que presentó las credenciales es considerada una identidad autenticada.

##### *Mecanismos de Autenticación*

Cuando el usuario ingresa un nombre de usuario y la contraseña en un sitio Web, hay diferentes mecanismos de autenticación que pueden ser implementados para verificar el login. La mayoría de servidores Web soportan autenticación Basic y Digest, otros soportan solamente autenticación Basic. Si el servidor Web permite el uso de Digest Authentication.

Basic Authentication codifica el nombre del usuario y la contraseña usando una técnica que hace éstos ilegibles, pero es trivial su decodificación. Por su parte Digest Authentication codifica el nombre del usuario y la contraseña por medio de una técnica que los hace virtualmente imposible de decodificar.

#### **AUTORIZACIÓN:**

Después de que una identidad ha sido autenticada, el proceso de autorización determina si dicha identidad tiene acceso a un recurso específico. El proceso de autorización limita los derechos de acceso ya sea por otorgar o denegar permisos específicos a una identidad autenticada.

---

<sup>13</sup> MSDNtraining. 2310B: Developing Microsoft ASP.NET Web Applications Using Visual Studio .NET

### 2.3.1.3. Seguridad a nivel de Datos

Seguridad a nivel de base de datos se determina a partir del proceso de autorización que se lleve a cabo, en el cual se cuida que quienes tengan acceso a la información sean quienes deban acceder a ella, con el objetivo de mantener la integridad de los datos.

Los conceptos claves en la seguridad de los datos son roles, usuarios y privilegios, los cuales se relacionan como se presenta en la Figura 6, cuyo esquema indica que los usuarios tienen asignados uno o varios roles, los cuales a su vez tienen ciertos privilegios CRUD (Crear, Recuperar, Actualizar o Eliminar) sobre los datos de la base de datos, en el nivel información que se otorguen o se denieguen los privilegios.

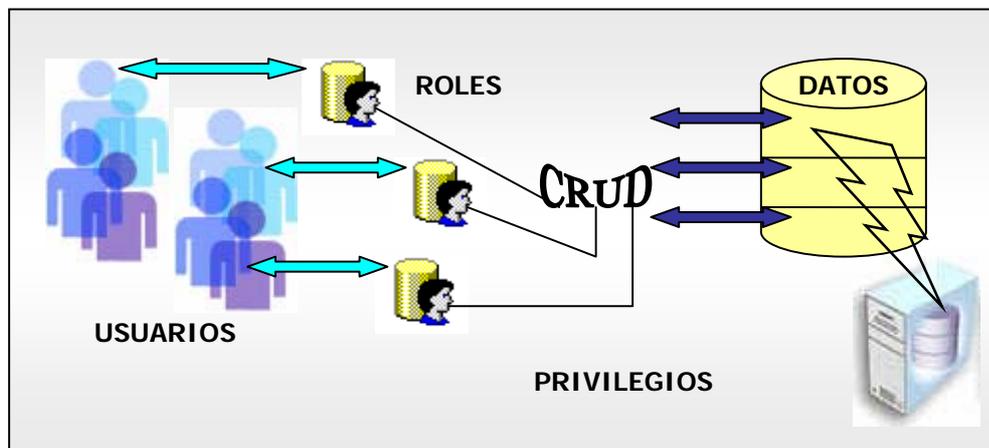


Figura 6. Seguridad de Base de Datos

## 2.4. Herramientas de Desarrollo para Servicios en Línea

A continuación se describen las herramientas de desarrollo que se consideran las más importantes y las más utilizadas actualmente en la industria del software en la Web. Se presentan los Sistemas de Administración de Bases de Datos, y las Plataformas sobre las cuales se desarrollan los sitios Web basados en Web Services.

## 2.4.1. Sistemas Administradores de Bases de Datos Relacionales (RDBMS)

### 2.4.1.1. MySQL

MySQL es un sistema manejador de base de datos Open Source SQL desarrollado, distribuido y soportado por MySQL AB, la cual es una compañía comercial fundada por desarrolladores MySQL, que construye sus negocios para proveer servicios alrededor de la base de datos MySQL. Las características esenciales de MySQL se presentan en la Tabla 6.

Característica	Descripción
<i>Es un sistema administrador de base de datos</i>	Para agregar, acceder y procesar datos almacenados en una base de datos, se requiere un sistema administrador de base de datos tal como MySQL Server, el cual facilita el manejo de grandes volúmenes de datos jugando el rol central en la computación como utilidades stand-alone o como partes de otras aplicaciones.
<i>Es un sistema administrador de base de datos relacional</i>	Teniendo en cuenta que una base de datos relacional almacena datos en tablas separadas, lo cual adiciona flexibilidad, MySQL usa SQL(Structured Query Language) para acceder a la base de datos.
<i>Es Open Source</i>	Open Source significa que es posible para cualquiera que descargue el software de Internet, usarlo y modificar el código según sus necesidades. El software de MySQL usa GLP(GNU General Public Licence).

**Tabla 6. Características de MySQL**

Las razones por las cuales usar el Servidor de Base de Datos de MySQL son las siguientes:

- ⇒ El Servidor de Base de Datos de MySQL es liviano, formal, seguro, recomendable y fácil de usar.
- ⇒ Fue originalmente desarrollado para manejar bases de datos grandes mas rápido que las soluciones existentes y por ello ha sido usado exitosamente con una gran demanda por ambientes de producción por varios años.
- ⇒ El servidor MySQL debido a su constante desarrollo ofrece actualmente un rico y usado conjunto de funciones, que proveen conectividad, velocidad y seguridad, lo cual lo hace altamente apropiado para acceder a bases de datos en Internet.
- ⇒ Para el caso de aplicaciones que manejan información geográfica, MySQL permite el uso de tipos de datos que soportan los tipos de datos geográficos, los cuales por lo regular son de gran tamaño.

### 2.4.1.2. Microsoft SQL Server

Microsoft SQL Server provee una Base de Datos escalable que combina el fácil uso, con análisis complejos y herramientas de data warehousing. Incluye una rica interface gráfica de usuario con un completo ambiente de desarrollo para crear aplicaciones que manejan datos.

Microsoft SQL Server es una familia de productos que satisface los requerimientos de almacenamiento de datos, de los sistemas de procesamiento de gran cantidad de datos y de sitios Web comerciales, a la vez que puede proveer servicios de almacenamiento de datos de fácil uso para un individuo o negocios pequeños.

La Tabla 7 presenta las características esenciales de SQL Server.

<b>Característica</b>	<b>Descripción</b>
<i>Integración con Internet</i>	El motor de base de datos SQL Server incluye soporte XML integrado. Tiene rasgos de escalabilidad, disponibilidad y seguridad requeridos para operar como componente de almacenamiento de datos en sitios Web grandes.
<i>Escalabilidad y Disponibilidad</i>	El mismo motor de base de datos puede ser usado a través de varias plataformas en equipos donde corra Microsoft Windows. SQL Server 2000 Enterprise Edition soporta rasgos tales como servidores federados, vistas indexadas y gran soporte a memoria que permite escalar a niveles de desempeño requeridos por los sitios Web de gran tamaño.
<i>Rasgos de Base de Datos a nivel Empresarial</i>	Soporta los rasgos requeridos para dar soporte a demandas de ambientes de procesamiento de datos. Protege la integridad de datos mientras minimiza la sobrecarga del manejo de usuarios concurrentes modificando la base de datos. La replicación permite mantener múltiples copias de los datos sincronizadas.
<i>Fácil de instalar, desplegar y usar</i>	Incluye un conjunto de herramientas administrativas y de desarrollo que mejoran el proceso de instalación, despliegue, manejo y uso de SQL Server a través de varios sitios. Soporta un modelo de programación integrado con Windows DNA, lo cual usa bases de datos SQL Server y data warehouses.
<i>Data warehousing</i>	SQL Server 2000 incluye herramientas para extraer y analizar resúmenes de datos para procesamiento analítico en línea. De igual forma incluye herramientas para visualizar diseño de base de datos y análisis de datos usando preguntas basadas en Inglés.

**Tabla 7. Características de Microsoft SQL Server**

### 2.4.1.3. Oracle

Oracle es una solución integrada para administrar y monitorear empresas globalmente. Es un manejador de base de datos relacional el cual usa los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes llenos de información.

Oracle desde la perspectiva de datos, se considera como el conjunto de datos que busca proporcionar la capacidad de almacenar, y acudir a los datos de forma consecuente con un modelo relacional. Desde la perspectiva empresarial, se considera como un conjunto de productos que ofrece una gran variedad de herramientas.

La Corporación Oracle ofrece RDBMS como un producto incorporado a la línea de producción, que también incluye 4 generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios. Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo.

Soporta aproximadamente 17 idiomas, corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar código. Esto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

Enterprise Manager está basado en una arquitectura liviana de 3 capas que ofrece opciones flexibles de despliegue, seguridad y escalabilidad. La arquitectura de tres capas para el producto está compuesta por los siguientes elementos:

- ⇒ Consolas, aplicaciones integradas y paquetes de administración.
- ⇒ Servidor(es) de Administración y repositorio de base de datos.
- ⇒ Agentes Inteligentes.

La Tabla 8 presenta las características esenciales de Oracle.

<b>Característica</b>	<b>Descripción</b>
<i>Es un sistema administrador de base de datos relacional</i>	Posee estructuras (objetos que contengan datos y que son accesibles a los usuarios), operaciones (acciones que manipulen datos u objetos) y reglas (leyes para gobernar la información, como y quien manipular).
<i>Partes principales</i>	Un RDBMS Oracle esta compuesto por tres partes principales: el Kernel de Oracle, las instancias del Sistema de Base de Datos y los archivos relacionados al sistema de Base de Datos.
<i>Ambiente Cliente Servidor</i>	Oracle soporta un verdadero ambiente cliente servidor, que establece un proceso entre bases de datos del servidor y el cliente para la aplicación de programas.
<i>Soporta Bases Datos grandes</i>	Oracle soporta bases de datos de todos los tamaños, desde severas cantidades de bytes y gigabytes en tamaño.
<i>Estructura física y lógica</i>	Las estructura física tal como los archivos del sistema operativo, son almacenados tangibles como cintas magnéticas, discos y otros. Oracle requiere varios archivos para su funcionamiento, que conforman su estructura física. A la estructura lógica le corresponde un espacio por unidad.

**Tabla 8. Características de Oracle**

## **2.4.2. Plataformas de desarrollo**

### **2.4.2.1. Microsoft .NET**

.NET Framework es una plataforma de computación que simplifica el desarrollo de aplicaciones en el ambiente altamente distribuido de Internet.

El framework de .NET está diseñado para cumplir los siguientes objetivos:

- ⇒ Proveer un ambiente de programación orientada a objetos consistente, en el cual el código del objeto es almacenado y ejecutado localmente, pero distribuido en Internet o ejecutado remotamente.
- ⇒ Proveer un ambiente de ejecución de código que minimice el despliegue del software y los conflictos entre versiones.
- ⇒ Proveer un ambiente de ejecución de código que garantice seguridad en la ejecución de código, incluyendo código creado por una tercera parte desconocida o de baja confianza.
- ⇒ Proveer un ambiente de ejecución de código que elimine los problemas de desempeño de escritura o ambientes interpretados.
- ⇒ Proveer experiencia al desarrollador a través de la ancha variedad de tipos de aplicaciones, tales como aplicaciones basadas en Windows y en Web.

- ⇒ Construir toda la comunicación basada en los estándares de la industria, para asegurar que el código basado en el Framework de .NET se pueda integrar con cualquier otro código que siga dichos estándares.

El Framework de .NET tiene dos componentes principales:

- ⇒ CLR (Common Language Runtime)

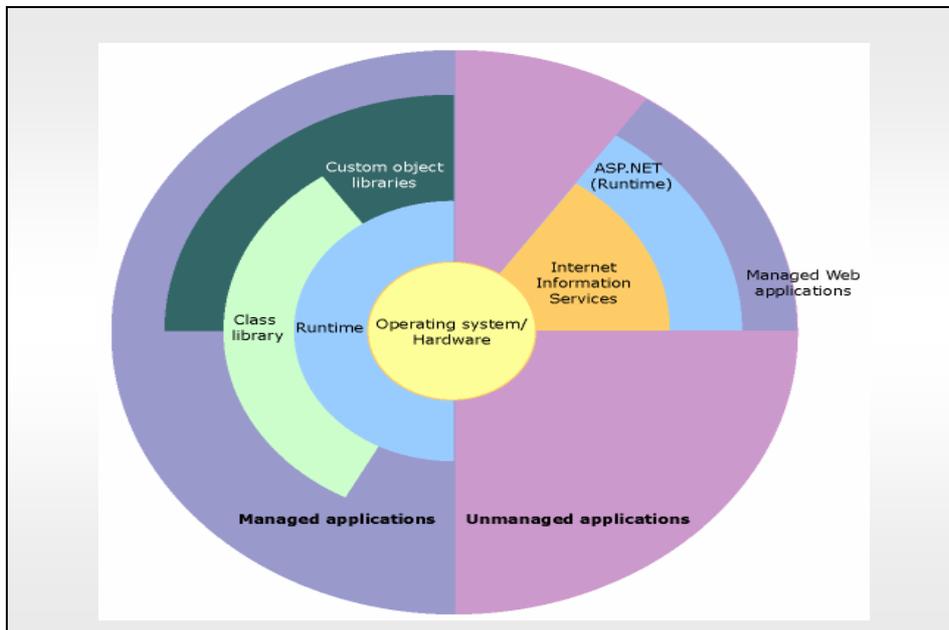
El CLR es el fundamento del Framework de .NET. Se entiende como un agente que maneja código en tiempo de ejecución, por medio de servicios centrales como administración de memoria, manejo de hilos y remoting (objetos remotos), mientras que también hace cumplir estrictamente la seguridad de tipos, y otras formas de código exacto que garantiza la seguridad y la robustez. En efecto, el concepto de código administrado, es un principio fundamental del runtime. El código que maneja el runtime es conocido como código administrado, mientras que el código no manejado por el runtime es conocido como código no administrado.

- ⇒ .NET Framework Class Library

La librería de clases es una colección orientada a objetos de tipos reusables, que sirven para desarrollar aplicaciones, pasando de aplicaciones de comando tradicionales o de interface gráfica de usuario(GUI), a aplicaciones basadas en las últimas innovaciones proveídas por ASP.NET, tales como Web Forms y XML Web services.

La visión del Framework de .NET es garantizar la interoperabilidad entre los sistemas sin importar la ubicación de los recursos físicos. La tecnología central para habilitar la interoperabilidad son los XML Web Services, los cuales proveen una metodología y una capa de transporte para el paso de información entre componentes de diferentes máquinas, diferentes redes y diferentes sistemas operativos.

La Figura 7 muestra la arquitectura del Framework de .NET.



**Figura 7. Arquitectura Framework de .NET**

#### 2.4.2.2. Java 2 Enterprise Edition (J2EE)

La definición que proporciona Sun de J2EE es la siguiente:

“J2EE define un estándar para el desarrollo de aplicaciones empresariales multicapa. J2EE simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchas de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja”<sup>14</sup>.

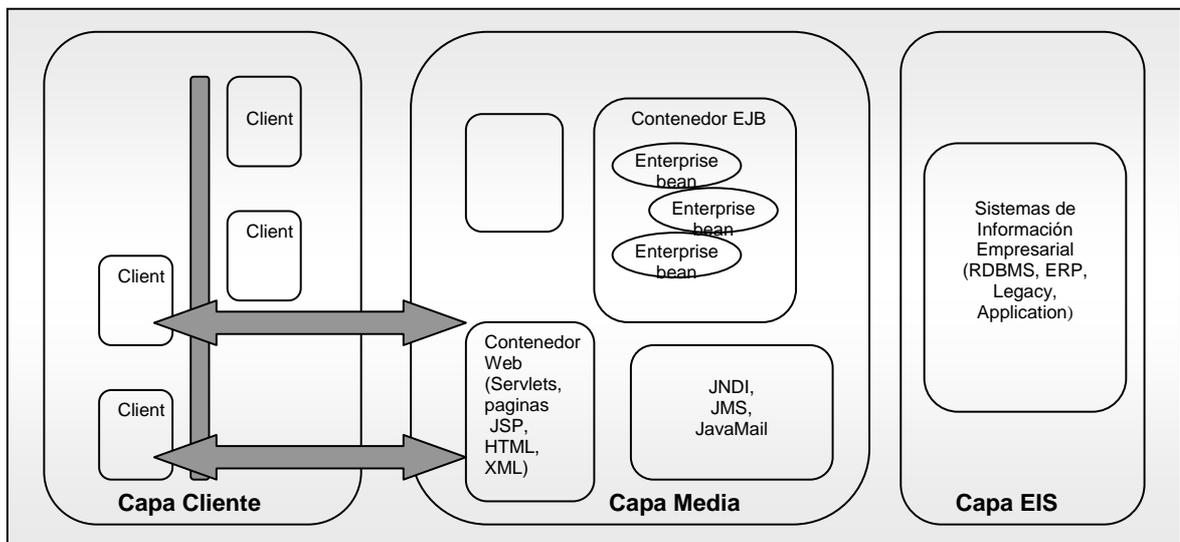
La plataforma J2EE es considerada como un conjunto de herramientas que crean un escenario ideal para el desarrollo y despliegue de aplicaciones escalables en la Web, las cuales poseen las siguientes características:

- ⇒ *Escalabilidad*: Por medio de la adición de componentes J2EE a una aplicación Web para soportar nuevos requerimientos, sin necesidad de reescribir todo el código de nuevo.
- ⇒ *Portabilidad*: Por medio de la creación de una aplicación en Windows o Linux, la cual se puede utilizar en cualquier plataforma que tenga disponible una Máquina Virtual Java (JVM).

<sup>14</sup> <http://java.sun.com> Fecha de Consulta: Marzo de 2004.

- ⇒ **Seguridad:** El entorno de seguridad de la plataforma J2EE permite que se definan restricciones de seguridad en el momento de despliegue de la aplicación, aislando así las aplicaciones de la complejidad de las implementaciones de seguridad.
- ⇒ **Soporte:** Por medio de la alta probabilidad de encontrar un contenedor de componentes (o servidor de aplicaciones) Web compatibles con J2EE, entre ellas IBM (Websphere), BEA (WebLogic), Apache (Tomcat), Sun (iPlanet), MacroMedia (JRun), etc.

La Figura 8 muestra la arquitectura de J2EE<sup>15</sup>.



**Figura 8. Arquitectura J2EE**

## 2.5. Estándares Open GIS Consortium (OGC)

### 2.5.1. Web Map Service (WMS)

WMS es un estándar definido por la OGC (Open GIS Consortium), la cual define a los mapas como la representación visual de geodatos, es decir que no son datos en si mismos. Con base en este concepto, un servicio de mapas web produce mapas de datos georeferenciados. Estos mapas por lo general son renderizados en formatos de imagen tales como PNG, GIF o JPEG u ocasionalmente en elementos gráficos basados en vectores en SVG (Scalable Vector Graphics) o en formatos Web Computer Graphics Metafile (WebCGM).

<sup>15</sup> Designing Enterprise Applications with the J2EE Platform, Segunda Edición

La especificación WMS busca la estandarización de la forma en la cual los mapas son solicitados por los clientes, y la forma en la que los servidores describen estas adquisiciones de los datos.

Este estándar especifica el comportamiento de un servicio que produce mapas georeferenciados, así como las operaciones para recuperar una descripción de los mapas ofrecidos por una instancia del servicio, para recuperar un mapa, y consultar a un servidor acerca de las características mostradas en el mapa.

Este estándar es aplicable para renderizaciones ilustradas de mapas en un formato gráfico, pero no es aplicable para recuperar datos de rasgos actuales o valores de datos de coberturas.

La Tabla 9 describe de manera general las 3 operaciones de la especificación de WMS.

<b>Operación</b>	<b>Descripción</b>
<i>GetCapabilities</i>	Retorna metadatos a nivel de servicio, el cual es una descripción de la información que contiene el servicio y los parámetros aceptables para las solicitudes.
<i>GetMap</i>	Retorna una imagen de un mapa cuyos parámetros geoespaciales y dimensionales fueron definidos.
<i>GetFeatureInfo</i>	Retorna la información acerca de rasgos particulares que se muestran en el mapa.

**Tabla 9. Operaciones WMS**

Un navegador web estándar puede pedir un Web Map Service para realizar estas operaciones simplemente realizando peticiones en forma de URLs. El contenido de tales URLs depende de cuáles de las tareas se solicitan. Todos los URLs incluyen una especificación del número de versión y del tipo de parámetro solicitado.

Cuando se invoca un GetMap, un cliente de WMS puede especificar la información que se demostrará en el mapa (una o más capas), posiblemente los estilos de éstas capas, la porción de tierra que está siendo mapeada es decir la caja de limitación("Bounding Box"), el sistema de referencia de coordenadas geográficas y de proyección para lo cual se utiliza el SRS(Spatial Reference System), el formato deseado para la salida, el tamaño de la salida (anchura y altura), la transparencia y el color del fondo.

Al invocar GetFeatureInfo el cliente indica el mapa que va a consultar y la localización en el mapa que le interesa. Cuando dos o más mapas se producen con la misma caja de limitación ("Bounding Box"), con el mismo

Sistema Espacial de Referencia, y con el mismo tamaño de la salida, entonces los resultados pueden ser acordados exactamente para producir un mapa compuesto. El uso de formatos de la imagen que apoyan fondos transparentes (GIF o png) permite que las capas más bajas sean visibles.

Las capas individuales del mapa se pueden solicitar de diversos servidores. La operación de WMS GetMap permite así la creación de una red distribuida de servidores de mapas, que para que cada cliente pueda construir mapas compuestos. Un proveedor particular de WMS en una red distribuida de WMS necesita solamente ser el administrador de su propia colección de datos. Esto está en contraste con la integración vertical que manejan los Web Mapping Sites, los cuales recolectan en un solo lugar todos los datos que serán asequibles por su propia interface privada.

Porque cada WMS es independiente, un WMS debe poder proporcionar una descripción de sus capacidades entendible por la máquina. Este servicio de metadatos permite a los clientes formular peticiones válidas y construir catálogos que puedan direccionar al cliente a particulares WMS's.

Opcionalmente un WMS puede permitir la operación de GetFeatureInfo. Si lo hace, sus mapas pueden ser consultados, y un cliente puede solicitar la información sobre características en un mapa agregando los parámetros adicionales al URL del mapa, especificando una localización y el número de facciones cercanas acerca de las cuales se retornará la información.

#### **2.5.1.1. Operaciones Web Map Service**

Las tres operaciones definidas por el Web Map Service son GetCapabilities, GetMap, y GetFeatureInfo, y ahora se describen:

##### **2.5.1.1.1. Get Capabilities**

(Requerido) Consiste en la solicitud al servidor de un documento en formato XML de contenido estandarizado por una DTD donde se detallan la capacidades de los dos comandos restantes y el contenido del servidor mapas(nombre y características de cada capa).

En el caso particular de un Web Map Service, la respuesta a la solicitud GetCapabilities es información general acerca del servicio mismo, e información específica acerca de los mapas disponibles.

### GetCapabilities Request

Para solicitar la operación Get Capabilities se debe armar un URL Request con ciertos parámetros como se muestra en la Tabla 10.

Parámetro Request	Requerido/ Opcional	Descripción
VERSION=versión	O	Versión del Request.
SERVICE=WMS	R	Tipo de Servicio
REQUEST=GetCapabilities	R	Nombre del Request
UPDATESEQUENCE=string	O	Número de secuencia o cadena para control de cache.

**Tabla 10. Request GetCapabilities (WMS)**

### GetCapabilities Response

En un Web Map Service la respuesta del lenguaje Extensible Markup Language (XML), tiene que ser válida de acuerdo con el Document Type Definition (DTD) del XML. El DTD especifica el contenido requerido y opcional de la respuesta, y cómo el contenido es formado.

Las Capabilities XML de un servidor pueden referenciar una copia exacta del DTD en lugar de una copia maestra en el URL. La copia DTD tiene que estar localizada en un URL completamente calificado y accesible que permita al XML validar el software para recuperarlo.

#### **2.5.1.1.2. Get Map**

(Requerido) La operación GetMap es diseñada para producir un mapa, el cual es definido para ser cualquier imagen ilustrada o un conjunto de elementos gráficos. Sobre la recepción de una solicitud Map, un servidor de mapas tiene que satisfacer la solicitud o lanzar una excepción en el formato solicitado.

Consta básicamente de la solicitud al servidor de una imagen que represente una o diversas capas para un ámbito, número de filas y columnas concretas. Se usa para obtener una imagen de una sola capa cada vez, del número de filas y columnas del ámbito de la vista. De igual forma permite solicitar una imagen transparente para las zonas sin objetos permitiendo superposición con otras capas en el área de la vista.

### GetMap Request

La solicitud es típicamente incrustada en el URL que es invocado en el WMS usando la operación HTTP GET. Los parámetros de un Request GetMap se encuentran en el Anexo A.

### GetMap Response

La respuesta para un request válido GetMap, tiene que ser un mapa de información georeferenciada de la capa solicitada, en el estilo deseado y teniendo el sistema de referenciación espacial, bounding box, tamaño, formato y transparencia.

Una solicitud válida GetMap tiene que producir un error de salida en el formato de excepciones solicitado(o una respuesta de error en un protocolo de red en casos extremos).

En un ambiente HTTP, el tipo MIME de valor retornado de la cabecera de la entidad Content-type tiene que igualar el formato del valor retornado.

#### **2.5.1.1.3. GetFeatureInfo**

(Opcional) Consta básicamente de la solicitud de los datos alfanuméricos asociados al objeto que ocupa una posición geográfica concreta (x,y), por ello es una solicitud del resultado de una consulta por localización. El resultado es un texto, preferiblemente en XML pero también posible en HTML o TXT.

GetFeatureInfo es una operación opcional. Esta es solamente soportada por aquellas capas que han sido definidas como consultables. Un cliente no tiene que distribuir una solicitud GetFeatureInfo para otras capas.

La operación GetFeatureInfo es diseñada para proveer a los clientes de un WMS con más información acerca de rasgos en las ilustraciones de los mapas que fueron retornados por anteriores solicitudes de mapas.

El caso de uso canónico para GetFeatureInfo es que un usuario mira la respuesta de un mapa solicitado y escoge un punto en tal mapa, para el cual obtener más información. La operación básica provee la habilidad para que un cliente especifique cuál pixel ha sido preguntado, cuál capa debería ser investigada, y cuál formato de información debería ser retornado.

Teniendo en cuenta que el protocolo WMS es sin estado, el request GetFeatureInfo indica el WMS que el mapa que el usuario está viendo tiene

incluido más de un parámetro original de la solicitud GetMap(todos menos VERSION y REQUEST). De la información del contexto espacial(BBOX, SRS, WIDTH, HEIGHT) en tal request GetMap, junto con la posición X,Y que el usuario escogió , el WMS puede (posiblemente) retornar información adicional acerca de tal posición. El comportamiento es empacado sobre el picture case. En el caso graphic element, la semántica de GetFeatureInfo es menos definida. La actual semántica de cómo WMS decide qué información retornar, o cuál exactamente retornar es parte de la labor del proveedor de WMS.

#### GetFeatureInfo Request

Los parámetros de un Request GetFeatureInfo se encuentran en el Anexo B.

#### GetFeatureInfo Response

El WMS tiene que retornar una respuesta acorde con el solicitado INFO\_FORMAT si el request es válido, o distribuir una excepción de otra manera. La naturaleza de la respuesta está en la discreción del proveedor WMS, pero éste tiene que pertenecer al los rasgos cercanos a (X,Y).

Para el caso del proyecto, la especificación WMS permite la estandarización del proceso de generación de los servicios, porque el servidor de web debe responder a cada consulta compatible con las especificaciones OGC, que provenga del navegador de mapas, y enviar los datos oportunos. Dicha respuesta puede ser: una imagen (un mapa - petición de tipo GetMap), una información textual, como en el caso de la lista de atributos o características de un elemento (petición GetFeatureInfo) o capa. Para permitir una respuesta rápida del servidor a cada acción del usuario remoto, lo que se plantea es un proceso para preparar los datos. Para los elementos REQUEST y RESPONSE, existen parámetros que son obligatorios y otros que son opcionales, dependiendo de la implementación particular.

### **2.5.2. Web Feature Service (WFS)**

WFS es un estándar definido por la OGC(Open GIS Consortium) que describe la especificación de codificación para geodatos en XML. La codificación descrita ha intentado activar el transporte y almacenamiento de información geográfica en XML incluyendo ambas propiedades la geometría y los rasgos geográficos.

Un Web Feature request consiste en una descripción de consultas o transformación de datos de operación que han sido aplicados a uno o más rasgos.

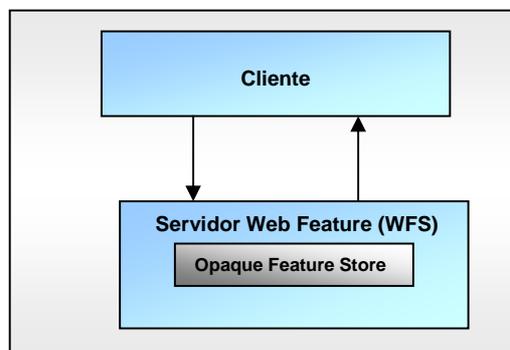
El request es generado al lado del cliente y es puesto en el servidor web de Features por medio de HTTP, entonces este último lee y ejecuta la operación.

El OGC Web Feature Service permite a un cliente devolver datos geoespaciales codificados en Geography Markup Language (GML) de múltiples Servidores web de Features.

### 2.5.2.1. Requerimientos para un Servicio Web Feature

- ⇒ Las interfaces deben ser definidas en XML
- ⇒ GML debe ser usado para expresar features dentro de la interface.
- ⇒ Un WFS debe ser capaz mínimo de presentar features usando GML.
- ⇒ Los predicados de filtros de lenguaje deben ser definidos en XML y ser derivados de CQL como se define en la especificación de Implementación de Interfaces de Catalogo del OpenGIS
- ⇒ Los Datasource usados para almacenar los features geográfico deben ser opacos a las aplicaciones cliente y sus únicas vistas de los datos deberán ser a través de una interface de WFS.
- ⇒ El uso de un subconjunto de expresiones de XPath para referencia de propiedades.

La arquitectura simple de un WFS se muestra en la Figura 9.



**Figura 9. Arquitectura Simple WFS**

Para soportar transacciones y consultas de proceso, son definidas las operaciones que muestra la Tabla 11.

Operación	Descripción
<i>GetCapabilities</i>	Un servicio web feature debe ser capaz de definir estas capacidades. Específicamente, debe indicar cuales tipos de rasgos puede servir y que operaciones son soportadas en cada tipo de rasgo.
<i>DescribeFeatureType</i>	Un servicio de web feature debe ser capaz de responder a un request, describir la estructura de cualquier tipo de rasgo que este pueda servir
<i>GetFeature</i>	Un servicio web feature debe ser capaz de servir Un request y devolver instancias de rasgos. En adición, el cliente deberá ser capaz de especificar cuales propiedades del rasgo buscará y podrá mostrar de acuerdo a los constraint de la consulta espacial y no espacial.
<i>Transaction</i>	Un servicio web feature debe ser capaz de servir un request de transacción. Una transacción está compuesta de operaciones que modifican rasgos; esta puede crear, actualizar y borrar operaciones en un rasgo geográfico.
<i>LockFeature</i>	Un servicio web feature debe ser capaz de procesar un request de candado en una o más instancias de un rasgo para la duración de la transacción. Esto asegura que las transacciones serializables sean soportadas

**Tabla 11. Funciones WFS**

Basadas en las operaciones descritas anteriormente, son definidos dos clases de servicios web feature:

*Basic WFS:*

Un WFS básico podría implementar las operaciones *GetCapabilities*, *DescribeFeatureType* y *GetFeature*. Estas se podrían considerar como servicios web feature de lectura únicamente (READ-ONLY).

*Transaction WFS:*

Una transacción de un servicio web feature podría soportar todas las operaciones de un servicio web feature básico y en adición este podría implementar operaciones de transacción.

Opcionalmente, una transacción WFS podría implementar la operación *LockFeature*.

## 2.5.2.2. Operaciones Web Feature Service

Las cinco operaciones definidas por el Web Map Service son GetCapabilities, DescribeFeatureType, GetFeature, Transaction y LockFeature, las cuales se describen a continuación.

### 2.5.2.2.1. GetCapabilities

(Obligatorio) Un servicio web feature debe tener la habilidad de describir sus capacidades. La sección de capacidades es usada para definir específicamente la lista de operaciones del WFS implementa. Un WFS básico podría incluir las operaciones GetCapabilities, DescribeFeatureType y GetFeature. Un WFS transaccional podría también incluir la operación Transaction, posiblemente la operación LockFeature y/o la operación GetFeatureWithLock.

#### GetCapabilities Request

Los parámetros de un Request GetCapabilities se describen en la Tabla 12.

Parámetro Request	Requerido/Opcional	Descripción
VERSION=version	O	Request version
SERVICE=WFS	R	Provee información acerca del servicio.
<GetCapabilities>	R	Incluido para definir la plataforma distribuida de cómputo disponible para este servicio.

**Tabla 12. Request GetCapabilities (WFS)**

#### GetCapabilities Response

El documento de capacidades de la operación GetCapabilities está compuesto por cuatro secciones principales:

- ⇒ *Sección de servicios:* La sección de servicios provee información acerca del mismo servicio.
- ⇒ *Sección de capabilities:* La sección de capacidades especifica la lista de request que el WFS puede manejar. Dos clases de servicios web feature, basados en las capacidades soportan.
- ⇒ *Sección de FeatureTypeList:* Esta sección define la lista de tipos de rasgo (y operaciones en cada tipo de rasgo) que están disponibles en un servicio web feature. Información adicional, así como SRS, acerca de cada rasgo también es proveída.

⇒ *Sección de filtro de capacidades:* El esquema de capacidades de filtro es definido en la sección Filter Encoding Implementation Specification. Esta es una sección opcional. Si existe, entonces es WFS soportará las operaciones advertidas dentro. Si la sección de capacidades de filtro no es definida, entonces el cliente asume que el servidor únicamente soporta el mínimo por defecto de operaciones asignadas por la especificación del Filter Encoding Implementation.

#### 2.5.2.2.2. DescribeFeatureType

(Obligatorio) La función de esta operación, es generar un esquema de descripción de servicios de tipos de rasgos por la implementación de un WFS. La descripción del esquema define la forma en la cual la implementación de un WFS espera instancias de rasgos para ser codificados en la entrada, así como la manera en la que las instancias de rasgos serán generadas en la salida.

##### Request DescribeFeatureType

Un elemento DescribeFeatureType contiene cero o más elementos TypeName que codifican los nombres de los tipos de rasgos que son descritos. Si el contenido del elemento DescribeFeatureType es vacío, entonces será interpretado como requesting una descripción de todos los tipos de rasgos que puede servir un WFS.

La Tabla 13 presenta los parámetros de un URL Request Describe Feature Type.

Parámetro Request	Requerido/Opcional	Descripción
VERSION=versión	O	Request version
SERVICE=WFS	R	Service type
OUTPUTFORMAT=GetCapabilities	O	Esquema de descripción del lenguaje

**Tabla 13. Request DescribeFeatureType (WFS)**

##### Response DescribeFeatureType

En el response de un request DescribeFeatureType, donde el valor del atributo outputFormat ha sido asignado como XMLSCHEMA, una implementación WFS debe presentar un documento XML que tenga una aplicación de esquema GML válido y defina el tipo de rasgo listado en el request.

El documento(s) presentado por el request DescribeFeatureType debe ser usado para validar las instancias de rasgos generadas por el WFS en forma de colecciones de rasgos en la salida o las instancias especificadas como entrada para operaciones de transacción. Las descripciones usan otro esquema de lenguaje de descripción, como DTD.

#### **2.5.2.2.3. GetFeature**

Esta operación permite devolver rasgos de un servicio web feature. Un request GetFeature es procesado por un WFS y un documento XML, que contiene el conjunto de resultados que es retornado al cliente.

##### *GetFeature Request*

Los parámetros de un URL Request GetFeature se describen en el Anexo C.

##### *GetFeature Response*

El formato response para un request GetFeature es controlado por el atributo outputFormat. El valor por defecto del atributo outputFormat será GML2. Esto indicará que un WFS debe generar un documento GML del conjunto de resultados que conforma la especificación version 2.1.1 del OpenGIS. Geography Markup Language Implementation, y más específicamente, la salida debe validarse una vez sea generado el esquema GML por la operación DescribeFeatureType.

Cualquier documento GML generado por una implementación WFS, en respuesta a una consulta donde el outputFormat es GML2, debe referenciar un apropiado documento de esquema de aplicación GML tal que la salida pueda ser validada.

#### **2.5.2.2.4. Transaction**

(Opcional) La operación Transaction es usada para describir operaciones de transformación de datos que son aplicadas a instancias accesibles de rasgo. Un servicio web feature debe procesar una operación Transaction directamente o posiblemente la traslade dentro de un data store el cual es conectado y se encargará de ejecutar la transacción. Cuando la transacción se ha completado, un servicio web feature generará un documento response XML indicando que se ha completado el estado de la transacción.

##### *Transaction Request*

Los parámetros de un URL Request Transaction se describen en el Anexo D

### Transaction Response

En el response a un request de transacción, un servicio web feature generará un documento XML indicando el estado de terminación de la transacción. Adicionalmente, si el request de la transacción incluye la operación <Insert>, entonces el servicio web feature service debe reportar los identificadores de rasgo de todos los rasgos nuevos creados. En el evento en que la transacción falle al ejecutarse, el servicio web feature también indicará esto en el response.

#### **2.5.2.2.5. LockFeature**

(Opcional)Las conexiones web son inherentemente sin estado. Como consecuencia de esto, la semántica de las transacciones serializables no son preservadas. Para entender el tema, se considerará una operación update. El cliente obtiene una instancia de rasgo. El rasgo es modificado en el lado del cliente y hace submit hacia la base de datos vía request Transaction para actualizar. La serialización se pierde porque no hay ninguna garantía de que mientras el rasgo fue modificado al lado del cliente, otro cliente no haya actualizado el mismo rasgo en la base de datos.

Una vez se asegure la serialización es requerido que el acceso a los datos sea exclusivo; que mientras haya acceso a un dato de una transacción, no pueda otra transacción modificar el mismo dato. Esto se puede cumplir por medio del uso de candados que controlen el acceso a los datos.

El propósito de la operación *LockFeature* es exponer un largo plazo al candado de un rasgo, mecanismo que asegura consistencia. El candado es considerado a largo plazo porque la latencia de la red podría hacer que el candado reciente de un rasgo sea relativamente más grande que el candado nativo de la base de datos comercial.

La operación *LockFeature* es opcional y no necesita ser implementada por un WFS para conformar su especificación. Si un WFS implementa la operación LockFeature, este hecho debe ser advertido en el documento de capacidades.

### LockFeature Request

Los parámetros de URL Request LockFeature se describen en el Anexo E.

### LockFeature Response

En respuesta al request LockFeature, un servicio web feature generará un documento XML. Este documento contendrá un identificador de candado que una aplicación cliente podrá usar en subsecuentes operaciones WFS para operar sobre el conjunto de instancias de candado del rasgo. El response debe contener también el elemento opcional <FeaturesLocked> y <FeaturesNotLocked> dependiendo del valor del atributo lockAction.

### **2.5.3. Geography Markup Language (GML)**

GML es una gramática XML definida por el OGC, la cual está escrita en XML Schema para cumplir funciones como modelar, transportar y almacenar información geográfica. GML provee varias clases de objetos para describir geografía, entre los cuales se encuentran los rasgos (features), los sistemas de referencia de coordenadas, la geometría, la topología, el clima, unidades de medida y valores generalizados.

GML es extensible, lo cual indica que cada persona tiene la posibilidad de crear sus propias entidades de dos formas: partiendo de las que ya existen por extensión ó tomar aquellas que define la especificación y modificarlas según como se requiera, sin dejar de lado los mecanismos de herencia y la estructura de los objetos.

Se entiende por rasgo geográfico “la abstracción de un fenómeno del mundo real; esto es un rasgo geográfico está asociado con una ubicación relativa a la Tierra”<sup>16</sup>. Teniendo en cuenta este concepto, se determina que una representación digital del mundo real se puede describir a través de un conjunto de rasgos. El estado de un rasgo está definido un conjunto de propiedades, donde cada propiedad se describe a través de un conjunto de tripletas {*nombre, tipo, valor*}. El número de propiedades de un rasgo podría junto con sus nombres y tipos, ser determinado por su tipo de definición. Los rasgos geográficos con geometría poseen características que pueden ser geográficamente evaluadas.

Una colección de rasgos tiene su tipo de entidad y por tanto se distinguen sus propias propiedades sumadas a las propiedades de las entidades que la componen.

Cuando se refiere a datos Geográficos, lo que se busca es información sobre las propiedades y la geometría de los objetos del mundo, lo cual difiere de la forma en la cual se simbolizan dicha información en un mapa, es

---

<sup>16</sup> <http://www.OpenGIS.org/docs/02-023r4.pdf> Fecha de Consulta: Noviembre de 2003.

decir factores como color y grosor de líneas entre otros. GML como codificación representa la información geográfica en formato texto, lo cual implica que sea fácil inspeccionar y modificar.

De manera similar a XML, GML ayuda a separar el contenido de la representación de información geográfica. A GML le interesa básicamente la representación del contenido de los datos geográficos, y por ello puede ser utilizado para hacer mapas, por medio del desarrollo de herramientas que interpreten GML, para lo cual se debe dar un estilo a los elementos GML en un formato adecuado y entendible para una pantalla gráfica con un navegador Web. Se utiliza un mapa de estilos para localizar elementos GML e interpretarlos mediante la utilización de estilos gráficos particulares.

Cuando se hace una solicitud de datos GML de un servidor GML los datos son devueltos en conjuntos de rasgos, para lo cual no existe limitante para la cantidad de rasgos que pueden ser contenidos en un conjunto de rasgos.

#### **2.5.3.1. Elementos que permite manipular GML**

##### ✓ Coberturas

Las entidades geográficas en GML contienen como subtipo coberturas, las cuales tienen una función definida para un dominio espacial y con un rango de valores homogéneos de 2 a n tuplas de dimensión.

##### ✓ Observaciones

Las entidades geográficas en GML contienen como subtipo observaciones, las cuales se definen como entidades GML con un tiempo en el que la observación tuvo lugar, y con un valor para la observación.

##### ✓ Sistemas de Referencia

Los sistemas de referencia proveen una escala de medida para asignar valores a un tiempo, localización, u otra cantidad descriptiva o cualitativa. Un sistema de referencia de coordenadas consta en un sistema de ejes coordinados, los cuales se encuentran enlazados a la tierra por medio de un dato que representa el tamaño y forma de la tierra.

GML agrega un sistema de referencia espacial basado en la tierra, el cual es extensible e incluye las principales proyecciones y sistemas de referencia geocéntricos.

✓ Tiempo

Un sistema de referencia temporal ofrece unidades estándar para la medida del tiempo y representa la longitud o duración temporal. Siguiendo la ISO 8601, el calendario Gregoriano se utiliza con el tiempo UTC en GML, como sistema de referencia temporal por defecto.

✓ Unidades de Medida

Un diccionario de Unidades de Medida (Units of Measure, UOM) provee las definiciones de las medidas numéricas de cantidades físicas, tales como longitud, temperatura, y presión, además de su conversión entre unidades.

✓ Calidad en los datos

Existe un esquema (dataQuality.xsd) destinado a definir la calidad de los datos, para lo cual éste define conceptos como exactitud posicional, exactitud posicional externa absoluta, exactitud posicional interna relativa, covarianza, matriz de covarianza, filas y columnas de los elementos de la matriz de covarianzas.

✓ Valores generalizados

GML permite al usuario definir sus propios tipos de valores. Estos tipos de valores pueden ser utilizados para las propiedades de las entidades y para otros tipos de objetos GML.

### **2.5.3.2. Ventajas de GML**

Basado en un modelo común de geografía:

Especificación Abstracta del OGC que ha sido desarrollado y acordada por la mayoría de los fabricantes SIG en el mundo.

Basado en XML:

Es más fácil integrar datos GML con datos no espaciales. XML es fácil de transformar. Utilizando XSLT o cualquier otro lenguaje de programación (VB, VBScript, Java, C++, Javascript, .NET, PHP, Perl, Python...) se puede fácilmente transformar XML de una forma a otra.

### De fácil edición:

Por medio de un editor XML estándar, se puede adicionar una nueva entidad o cambiar el valor de una propiedad, o si se requiere ajustar las entidades de geometría. El Schema de GML puede ser empleado para asegurar la integridad de los datos.

### Mapas de mayor calidad:

GML codifica información sobre entidades geográficas y objetos, y éstos pueden visualizarse con tanta resolución como se quiera. Los mapas pueden ser guardados en local, enviados por correo o imprimirlos. Se suele utilizar SVG para la visualización.

### Estilos de mapa personalizados:

GML almacena el "contenido" del mapa, no provee información de cómo esos datos del mapa tienen que ser representados. Se pueden aplicar diversas hojas de estilo a los datos geográficos de manera que se muestren como el usuario quiera.

### Sirven en los navegadores:

Cuando el cliente recibe un fichero GML, se convierte en un conjunto de objetos de dibujo que son representados como un mapa en el navegador. Normalmente se utiliza Scalable Vector Graphics (SVG) como lenguaje para los objetos de dibujo. No se requiere comprar software al lado del cliente.

### Verificación Automática de Integridad de Datos:

XML permite verificar la integridad de los datos, por medio de los Schemas, los cuales especifican la estructura de un documento XML de una forma que permita que una herramienta de validación pueda verificar que la instancia del documento dado cumple con este Schema. GML está especificado por su correspondiente Schema.

### Puede ser leído por Herramientas Públicas/Genéricas:

GML es texto, estructurado, y cualquiera de los variados editores XML pueden ser utilizados para mostrar esta estructura. Esto hace que sea sencillo visualizar y navegar por datos GML.

### *Puede ser integrado con datos no-espaciales:*

Con GML es sencillo aportar enlaces a otros elementos de datos XML y esto mejorará drásticamente con la introducción de XLink y XPointer. Incluso enlaces a elementos que no son XML pueden ser fácilmente manejados utilizando la sintaxis URI ya establecida.

### *Mejores capacidades de consulta:*

En las entidades basadas en GML, cuando se hace click sobre una entidad, siempre se identifica sin la entidad deseada, se pueden apagar y encender los distintos temas, con lo cual resulta fácil identificar entidades que están dentro de otras.

### *No se tiene que pensar sólo en un navegador Web:*

GML se puede utilizar como un formato de intercambio de propósito general para información geográfica. Los datos geográficos en GML se pueden enviar a cualquier dispositivo que esté preparado para procesar XML.

### *Encadenamiento de servicios:*

Si cada sitio provee un servicio discreto, por medio de GML se pueden encadenar porque es un formato general, extensible y basado en XML, lo cual lo hace fácil de manipular, cambiar, y de añadir otros datos a su contenido.

### *Control sobre los contenidos:*

Puesto que GML está basado en las entidades, resulta totalmente sencillo aplicar funciones de filtrado que permitan a los usuarios descargar sólo los tipos de entidades que quieran que aparezcan en sus mapas.

### *Entidades animadas:*

Objetos y entidades que cambian en el tiempo tienen cabida en GML, y pueden ser representados como gráficos animados utilizando SVG.

### *Mapas editables:*

Se pueden hacer anotaciones sobre los mapas basados en GML que han sido descargados y visualizados utilizando el navegador. Cuando se ha convertido el GML a SVG, el usuario puede aplicar las herramientas de edición gráfica en el cliente y añadir texto, resaltar entidades y dibujar cualquier tipo de forma sobre el mapa.

### **2.5.3.3. Desventajas de GML**

#### Costo de formación y adaptación a GML:

No existe el conocimiento de utilizar GML, las personas se deben capacitar y sobre todo adaptar a esta nueva tecnología, teniendo en cuenta que es la más eficiente para el manejo de información geográfica

#### Gran tamaño de los ficheros:

El almacenamiento de la información no está optimizado ya que GML es texto. Para ello se comprimen los archivos con *gzip*, para el transporte y el almacenamiento

#### Lenta implantación:

Debido a la falta conocimiento de ésta nueva tecnología, se dice que es de lenta implementación, ya que provoca un cambio de infraestructura para enfocar la nueva tecnología de manera adecuada.

### **2.5.3.4. Esquemas de documentos GML**

Los esquemas de los documentos GML son esquemas XML que definen tipos XML y elementos XML para codificar objetos GML con identidad. De igual forma definen elementos para codificar propiedades GML de aquellos objetos, y atributos XML cualificando dichas propiedades.

Un objeto GML es un elemento XML de un tipo derivado directamente o indirectamente del tipo AbstractGML. De esta derivación, un objeto GML puede tener un atributo gml:id. Por su parte, una propiedad GML puede no ser derivada del tipo AbstractGML, puede no tener un atributo gml:id o cualquier otro atributo de tipo ID de XML. Un elemento es una propiedad GML si y sólo si es un elemento hijo de un objeto GML.

Una regla importante es que objetos GML no pueden aparecer como los hijos inmediatos de un objeto GML y por consecuencia, los elementos no pueden ser a la vez un objeto GML y una propiedad GML.

### **2.5.3.5. Instancias de documentos GML**

GML usa una sintaxis explícita para instanciar el modelo Feature definido en la especificación abstracta del OGC en una instancia de documento XML. Un Feature es codificado como un elemento XML con el nombre del tipo de

Feature. Por su parte, una propiedad de un Feature también es codificada como un elemento XML, cuyo nombre es el de la propiedad. Se obtiene como resultado un documento XML *layered*, en el cual los elementos XML corresponden a los rasgos(features), objetos o valores que ocurren intercalando con elementos XML correspondientes a las propiedades que se han mencionado. La función de un feature, objeto o valor en un contexto, siempre puede ser determinado por la inspección del nombre del elemento de propiedad el cual a su vez lo contiene directamente a éste, o el cual lleva la referencia a éste.

## **2.6. Visualización de Mapas**

### **2.6.1. Scalable Vector Graphic (SVG)**

Es un formato estándar de fichero vectorial para la web, desarrollado por la W3C. En las páginas Web, el contenido gráfico está representado por lo regular en formatos como GIF y JPG, los cuales son mapas de bits y por ello tienen grandes limitaciones.

#### *Funcionalidad de SVG*

Solucionar las limitaciones de los formatos GIF y JPG, al igual que formatos vectoriales como SWF (formato para aplicaciones multimedia) que no están diseñados para Web.

SVG es extensible, lo que permite adicionar definiciones propias, sin perder la compatibilidad con las definiciones estándar. De igual forma SVG separa totalmente el contenido de la presentación.

#### **2.6.1.1. Ventajas del formato SVG**

- ⇒ Ventajas asociadas a todo formato vectorial tales como: escalable, compacto, con contornos suavizados, transparencias, y capaz de incluir bitmaps.
- ⇒ La calidad de color es excelente; el color del gráfico se puede calibrar con los sistemas estándar de gestión de color.
- ⇒ Puede incluir código (scripts) que modifican el gráfico dinámicamente en función de las necesidades.
- ⇒ El texto que incluyen es editable: admite las fuentes escalables más comunes, como TrueType o Type 1. Esto supone una diferencia con los actuales GIF o JPG: el texto que contienen se puede editar, seleccionar, ser indexado por los buscadores.

- ⇒ El fichero SVG no es binario: se trata de un fichero de texto normal y corriente, por ello sus contenidos se pueden indexar y buscar.
- ⇒ Admite efectos como sonido, efectos visuales al hacer clic o mover el ratón, etiquetas informativas.
- ⇒ Al ser XML, es un formato extensible: los fabricantes podrán adoptarlo como formato nativo de sus aplicaciones, añadiendo las características específicas que deseen, pero siempre mantendrá la compatibilidad básica y universal con toda aplicación que reconozca el formato.
- ⇒ Puede generarse dinámicamente en un servidor web como respuesta a instrucciones de Java, JavaScript, Perl o XML.

### **2.6.1.2. Herramientas para crear y ver SVG**

Algunos de los programas que tienen capacidad de crear SVG son:

- ⇒ Corel Draw (versiones 9-10): Dispone de un plugin de exportación a SVG.
- ⇒ Adobe Illustrator 9: Enfocada a la creación de gráficos web, incluyendo SVG.
- ⇒ Mayura Draw: Aplicación de dibujo vectorial básica
- ⇒ Jasc Trajectory Pro: Una solución pensada desde un principio para la creación de gráficos SVG.

#### La aplicación exploradora de SVG

La aplicación exploradora de SVG permite la visualización y el análisis de datos espaciales, codificados por GML. El programa respectivo es llevado a cabo en el lenguaje ECMAScript y usa el plugin SVG de Adobe para la visualización de los datos geográficos en atributos básicos, esta aproximación permite el desempeño de las operaciones respectivas al lado del cliente únicamente, cuando descarga los datos requeridos en GML.

A continuación se presenta un lista de las principales funcionalidades:

- Cargar archivos GML.
- Manejo de visualización (paneo, zoom, símbolos, colores, estilos, labels).
- Organización de capas jerarquizada y manipulación (selección, arriba/abajo, ocultar/mostrar, copiar, borrar, etc.).
- Descripción de elementos.
- Clasificación temática en atributos básicos.
- Visualización textual del código fuente GML.
- Visualización de mapas (compuestos de un conjunto de mapas).
- Descripción de atributos de capas.

- Lista de valores de atributos de capas y manipulación (ordenamiento).

### Arquitectura de la Aplicación

La aplicación exploradora SVG ha sido desarrollada usando una arquitectura modular, compuesta por un conjunto de bloques funcionales. Cada bloque contiene las funciones del ECMAScript llevadas a cabo en funcionalidades específicas de la aplicación.

Todas las funciones son implementadas al lado del cliente; ellos integran varios componentes y son activados por la interacción en la interfaz del usuario cargadas cuando la aplicación arranca. La arquitectura se presenta en la Figura 10.

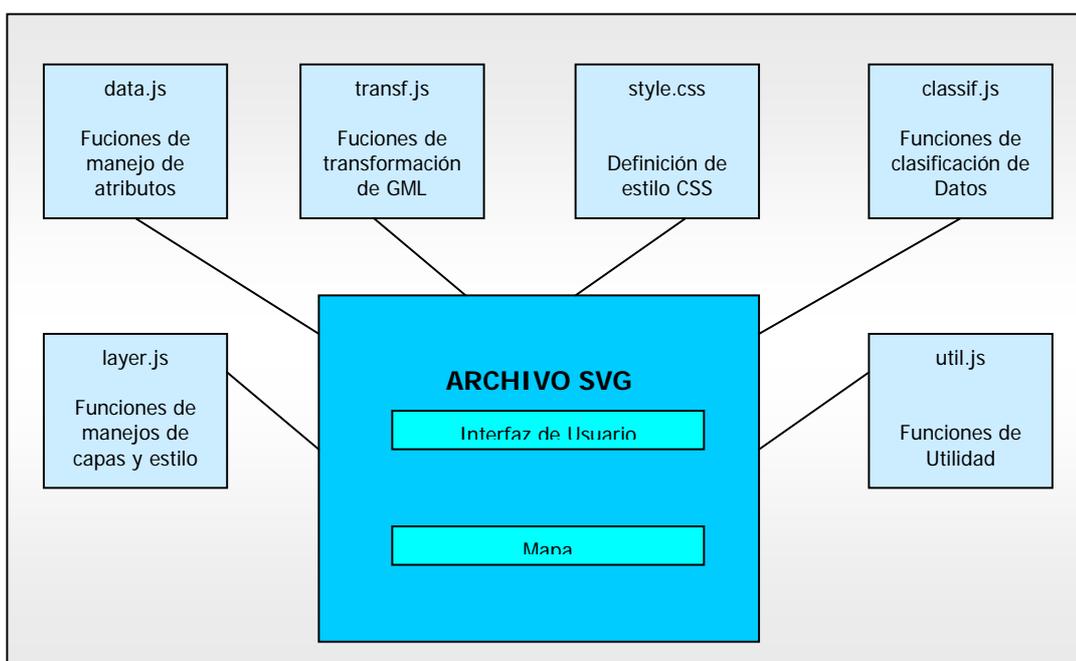


Figura 10. Arquitectura aplicación SVG<sup>17</sup>

Los bloques principales de la aplicación son:

- **layer.js**: Este contiene el script de funciones para el manejo de la organización jerárquica de capas y el estilo de visualización de los elementos de capa.

<sup>17</sup> <http://www.svgopen.org/2003/papers/SvgExplorerOfGmlData/> Fecha de Consulta: Marzo de 2004.

- *data.js*: este se refiere al manejo de atributos de las capas activas. Este contiene un parser para extraer del archivo GML los atributos de datos de los elementos geográficos e insertarlos en la estructura de datos "Attribute Table". Esta tabla es usada por todas las otras las operaciones de funciones en los atributos de datos.
- *tasf.js*: este módulo contiene el script de funciones que transforma los datos GML a datos SVG. Estas funciones son activadas cuando el archivo GML se carga y se parsea la estructura de documentos GML para crear la representación de elementos SVG en forma gráfica los datos geográficos. La transformación es hecha un estilo gráfico por defecto a los datos GML; el estilo puede ser modificado por el usuario a través de la interfaz de usuario gráfica de la aplicación.
- *cassif.js*: esta comprende las funciones para el manejo y clasificación de operaciones. Estas operaciones obtienen los request de los usuarios, analizan los atributos (en la tabla de atributos) de los elementos geográficos, desempeñan la clasificación de atributos y muestran los resultados en el mapa desplegado.
- *uil.js*: éste módulo está compuesto por un conjunto de funciones de soporte que tratan aspectos secundarios de las aplicaciones.
- *syle.css*: este es un archivo CSS que contiene la definición de los estilos aplicados a los elementos gráficos de la aplicación de interfaz de usuario.

### 2.6.2. Shape

El formato shapefile es usado para datos en formato vector en un entorno SIG. Este formato consta de un mínimo de 3 archivos para su correcto funcionamiento:

- ⇒ Archivo principal (.shp)
- ⇒ Archivo de índice (.shx)
- ⇒ Archivo de base de datos (.dbf)

Todos estos tres archivos o más deben ser presentados para poder utilizar adecuadamente el formato shape.

## **3. GUÍA METODOLÓGICA**

### **3.1. Introducción**

El objetivo principal de plantear una guía metodológica para la generación de servicios en línea geográficos, es guiar a un usuario con conocimientos y experiencia en la construcción de web services e información geográfica, a seguir una serie de pasos, que le permitirán construir una aplicación basada en estándares que previamente han sido definidos por entidades interesadas en la interoperabilidad entre sistemas como es el OGC (Open GIS Consortium) y que han definido la mejor forma de solicitar e intercambiar datos.

Para el logro de ello se diseñará una arquitectura multicapa que proveerá una infraestructura sólida sobre la cual se construirán los servicios basados en los estándares WMS y WFS, para el intercambio de información se empleará GML. Esta arquitectura también proveerá niveles de seguridad en varias capas, que se explicarán en el desarrollo de la guía. Por otra parte se planteará el uso de software libre para la implementación, teniendo en cuenta los altos costos del software especializado.

Una particularidad de la guía metodológica es que ésta se presentará organizada por fases, como una adaptación al modelo en cascada para el desarrollo de sistemas de información. Las fases planteadas son: Planeación, Análisis, Diseño, Implementación y Soporte con el fin de facilitar el manejo de actividades y artefactos de cada fase, y a su vez para dividir los pasos en fases lógicas que permitan comprender el ciclo de vida que abarca el desarrollo de los servicios en línea, para quien decida adoptar la guía metodológica.

### **3.2. Fase de Planeación**

#### **3.2.1. Identificación de la necesidad de generar servicios en línea**

Teniendo en cuenta que los servicios en línea han constituido el siguiente paso en la evolución de la tecnología orientada a objetos, y representan una revolución al alejarse de las arquitecturas tradicionales tipo *cliente-servidor* a

nuevas arquitecturas distribuidas tipo (*peer-to-peer*), se resalta la importancia de generar dichos servicios porque están formados a través de un conjunto de estándares que permiten a los desarrolladores implementar aplicaciones distribuidas, utilizando herramientas muy distintas para crear aplicaciones que utilizan una combinación de módulos de software, que son llamados desde diversos sistemas distribuidos en regiones geográficas distintas.

Lo anterior hace referencia a servicios en línea en general, es decir los que podría prestar una entidad financiera, de gobierno o como en el caso particular que quiere tratar esta guía, para entidades que generan información geográfica, lo cual no implica que su negocio sea explícitamente producir este tipo de información, sino que el poseerla trae consigo beneficios que podrían ser aprovechados de mejor forma al implementarlos en servicios en línea.

De allí que una entidad que genere información geográfica detectará la necesidad de generar servicios en línea cuando perciba, que toda esta información tendrá un mejor uso, en la medida en que diversos usuarios incluyendo los mismos funcionarios de la empresa quienes deben consultar, actualizar y modificar la información en el instante que lo necesiten, solo necesitarán de una conexión a Internet, pero sobretodo una empresa deberá contemplar la necesidad de generar servicios en línea basados en los estándares que ha escrito el OGC (Open GIS Consortium) para poder compartir información con las diversas entidades que posean este tipo de información y lograr la interoperabilidad entre los sistemas por medio de los Web Services.

Además de esto, los servicios en línea permitirán optimizar procesos, por medio de la integración de diversos sistemas de información que posea la empresa y eliminando actividades manuales necesarias para coordinar el flujo de información al interior de la empresa, por ende también optimizará los procesos que involucren interacción con otras empresas.

Para detectar la necesidad de generar servicios en línea geográficos, se propone tener en cuenta los beneficios traen consigo los servicios en línea:

- ✓ A grandes empresas y organizaciones, optimizar su operación al reorganizar de manera integral y modular sus actividades/servicios internos, así como ampliar la oferta al ofrecer nuevos servicios que incluyan operaciones de otras empresas.
- ✓ A medianas empresas/organizaciones, ampliar su oferta de servicios especializados/modulares a clientes y empresas.

- ✓ Mejores conexiones con clientes y proveedores permite optimizar las actividades de la cadena de valor y, al mismo tiempo, colaborar en mejorar los procesos internos, como por ejemplo, el desarrollo de productos.
- ✓ Actualmente el mejor uso de los Servicios Web se obtiene en la frontera de los sistemas de información, donde los problemas de conectividad son más complejos y las ganancias en eficiencia son mayores en el corto plazo.
- ✓ En el largo plazo, la cooperación entre diferentes negocios funciona de manera similar a una red, en la cual cada integrante enfoca sus esfuerzos en las actividades que generan valor, lo cual hace que la red sea más eficiente y flexible en aquello que ésta ofrece.

### **3.2.2. Factores a tener en cuenta cuando se planea construir un servicio en línea**

En esta fase de planeación es importante entrar a analizar si la persona o entidad que desean hacer uso de esta guía metodológica tienen ciertos elementos mínimos, para llevar a cabo el seguimiento de los pasos planteados por la misma y si son suficientes los recursos que posee para llevar a cabo la implementación de una manera adecuada.

A continuación se plantean los requisitos que se deben tener cuando se planea crear servicios en línea:

- Tener información geográfica:

Se considera un aspecto fundamental que la persona o entidad posean información geográfica, pues ésta será la base sobre la cual los servicios serán construidos; dicha información será consultada y en el caso que se desee implementar un WFS transaccional será la que se modificará. La información aquí es considerada como la materia prima para generar servicios en línea, que permita a la organización realizar una gestión adecuada de la información generada por los servicios.

El usuario puede tener la información en diversos formatos (archivos shape, SVG, tablas, etc) y sin un modelo relacional creado, en la fase de diseño, se explicará como modelar una base de datos relacional de capas de datos geográficos.

De esta forma si la información que se tiene es diferente del tipo geográfico, será tema que no compete a la guía, pues esta solo se enfoca a la creación de servicios geográficos y no de otro tipo.

- Tener experiencia en UML:

Un aspecto indispensable para el modelamiento del negocio, es tener alguna experiencia con UML (Unified Modeling Language). Este se define como “una notación estándar para modelado, pero con la característica importante de que es abierta, no es propietaria. UML no es un método de análisis y diseño orientado a objetos, sino una notación general y simple que sirve para utilizarse en métodos de desarrollo de software”<sup>18</sup>.

Se considera importante para esta guía metodológica, que quien la adopte, tenga alguna experiencia con UML, porque éste es de gran utilidad para el modelado de negocios y de software en cada una de las fases de desarrollo, para los tipos de sistemas existentes. La gran ventaja es que sirve para el modelado general de cualquier desarrollo que tenga un comportamiento estático y dinámico. UML en ésta guía metodológica se utilizará para el modelamiento de casos de uso.

- Tener infraestructura física y humana:

Un aspecto indispensable para el modelamiento del negocio, es tener alguna infraestructura tanto física como humana. Física se refiere a tener el entorno adecuado, los equipos suficientes, el hardware y software mínimo para soportar la arquitectura que permite generar servicios en línea basados en información geográfica.

Humana se refiere a recurso humano capacitado para dar soporte, mantenimiento y uso a los servicios de manera adecuada para sacarle el provecho que se merece los resultados obtenidos por los servicios.

### **3.3. Fase de Análisis**

#### **3.3.1. Análisis de la situación actual**

El análisis de la situación actual permite determinar los factores relevantes en los procesos actuales que lleva la organización con respecto al manejo de la información geográfica que ésta posee, y la manera como estos factores influyen en la implantación de la nueva arquitectura, para lo cual se establecen ciertos parámetros que reglamentan el proceso de migración de

---

<sup>18</sup> JOYANES AGUILAR Luis. Programación Orientada a Objetos. 1998.

un sistema a otro si éste existe, o se implanta el nuevo sistema partiendo de ceros.

Analizar la situación actual permite comprender el entorno y determinar si las circunstancias están dadas para implantar la arquitectura planteada, por lo cual éste análisis es quizás el factor más relevante para un correcto desarrollo de los servicios en línea basados en la aplicación de ésta guía metodológica, porque el ámbito del proyecto debe ser claramente definido y comprendido en su totalidad, con el objetivo de lograr eficiencia en el desarrollo de los servicios geográficos a través de los estándares WMS y WFS.

Los factores que se recomiendan analizar en ésta fase son los siguientes y se explican a continuación:

### **Infraestructura de Datos Geográficos**

Teniendo en cuenta que la información es el factor sobre el cual la organización tiene valor competitivo en el mercado, se plantea un análisis de la infraestructura de datos que tiene la empresa por medio de las siguientes actividades:

- ⇒ Determinar el tipo de información que la empresa posee, es decir si tiene información de mapas y de metadatos asociados a los mismos.
- ⇒ Determinar la forma como se encuentra estructurada la información, en cuyo caso se tienen 2 opciones: que no se tenga un modelo de datos estructurado o que se tenga.
- ⇒ Determinar la forma de almacenamiento de la información, es decir si se encuentra en bases de datos, los formatos en los que se encuentra, y los dispositivos de almacenamiento que se tienen.
- ⇒ Determinar la cantidad de información geográfica que posee la organización y los niveles de acceso que tiene la misma, es decir acceso interno para la organización, acceso externo para otras organizaciones o acceso público.
- ⇒ Determinar los procedimientos que se llevan para la manipulación de la información, es decir los procedimientos de captura, entrada, análisis y salida de información, teniendo en cuenta los factores de seguridad en cuanto a nivel de acceso se refiere.

Con base en las actividades mencionadas, los indicadores que se recomiendan tener en cuenta para el levantamiento de la infraestructura de datos geográficos, se determinan por medio de los diagramas de los modelos de datos que se tengan, teniendo en cuenta que son ellos quienes van a reflejar la estructura actual de la información, y de allí

conocer si se tienen los datos requeridos y si éstos se encuentran almacenados en una forma adecuada como lo es una base de datos relacional.

### **Infraestructura Física y Técnica**

La infraestructura física y técnica se refiere al entorno físico que se tiene para el manejo de la información geográfica que posee la organización. Para realizar éste análisis se plantea levantar la siguiente información:

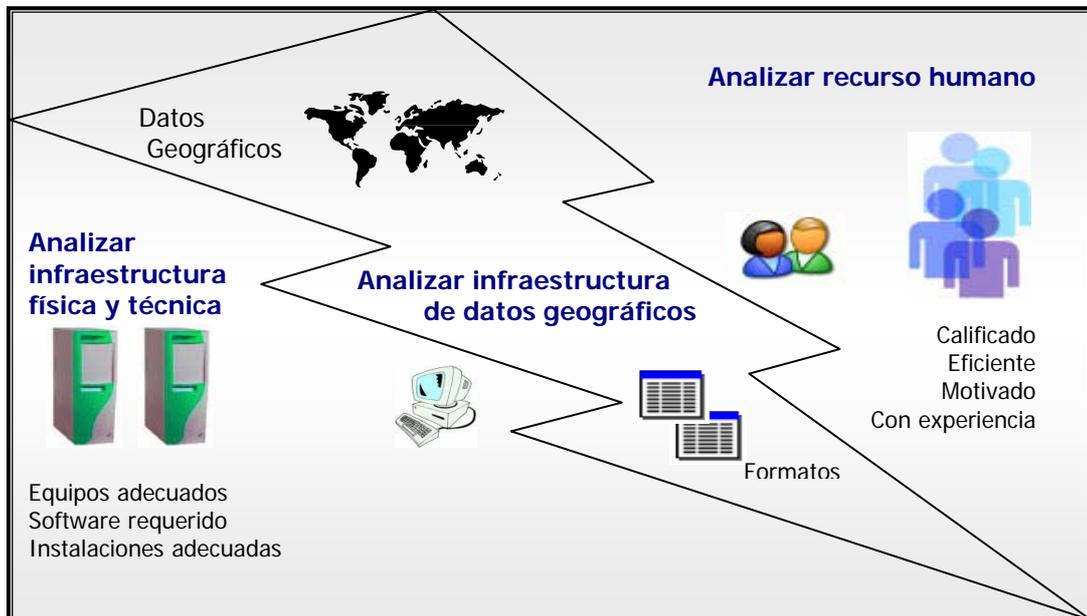
- ⇒ Número de servidores y de PC al interior de la organización, que se encuentren asociados al manejo y flujo de la información geográfica.
- ⇒ Características físicas de dichos equipos, lo que incluye hardware y software relacionado con el manejo de éste tipo de información, teniendo en cuenta que los programas SIG son aquellos que proveen las funciones y herramientas para almacenar, analizar y desplegar la información geográfica, que tiene componentes como: Herramientas para la entrada y manipulación de datos, Sistema manejador de base de datos, y herramientas de búsquedas, análisis y visualización de información geográfica.
- ⇒ Distribución geográfica de dichos equipos, para saber la posibilidad de implantar el sistema de manera distribuida, lo cual permite un manejo mas eficiente de grandes volúmenes de información, y factores como alta disponibilidad y replicación.
- ⇒ Análisis de conectividad, es decir la red que se maneja a nivel interno y a nivel externo, si se tiene acceso a Internet.
- ⇒ Condiciones físicas adecuadas, lo cual se refiere a características de las instalaciones aptas para el recurso humano, con el fin de controlar riesgos profesionales.

### **Recurso Humano**

La tecnología de los Sistemas de Información está limitada si no se cuenta con el personal que opera, desarrolla y administra el sistema, por lo cual es indispensable antes de analizar el desarrollo de un sistema de información, determinar si se cuenta con el personal adecuado, para llevar a cabo dicho fin se plantean las siguientes actividades:

- ⇒ Analizar el personal de la organización que tiene contacto de manera directa o indirecta con la información geográfica, éste análisis incluye factores como: nivel de experiencia en el área, motivación, conocimientos, habilidades, eficiencia y disponibilidad de tiempo.
- ⇒ Determinar si se cuenta con el recurso humano capacitado y apto para el desarrollo de los servicios en línea.

Los factores anteriores se presentan en la Figura 11:



**Figura 11. Levantamiento de la situación actual**

### 3.3.2. Levantamiento de requerimientos

El análisis de requerimientos es una de las etapas más importantes en el desarrollo de cualquier proyecto de software, en esta etapa se tratan de definir las condiciones o capacidades necesarias para uno o varios usuarios con el fin de solucionar un problema o conseguir un objetivo.

En este caso, para la creación global del sistema se necesita comprender todos los objetivos y necesidades del usuario. En primer lugar, se ha de especificar el comportamiento del sistema desde el punto de vista del usuario. Una vez hecho esto, se puede pensar en la arquitectura general del sistema, en términos de componentes físicos: hardware, software, usuarios y la comunicación entre ellos.

La determinación de los requerimientos para la guía metodológica se hallan con base en los estándares definidos por el OGC (Open GIS Consortium), además de hablar con los usuarios finales sobre sus necesidades y los servicios que desearían implementar en su organización. Se pueden modelar los requerimientos de usuario mediante lenguajes como UML, que

disponen de modelos llamados casos de uso pensados para describir las funcionalidades necesarias para los usuarios.

En este sentido, es indispensable comprender los niveles de requerimiento que existen, los cuales se describen en la Tabla 14, y cuyos formatos para cada nivel fueron desarrollados en este proyecto.

<b>Tipo</b>	<b>Descripción</b>	<b>Captura</b>
<i>Requerimientos de Negocio</i>	Representan el nivel alto de los objetivos de la organización.	Documento (Formato Anexo H)
<i>Requerimientos de Usuario</i>	Describen las tareas que los usuarios deben ser capaces de completar con el producto.	Casos de Uso (Formato Anexo I)
<i>Requerimientos Funcionales</i>	Definen la funcionalidad del software, que habilite a los usuarios a cumplir sus tareas y satisfacer los requerimientos del negocio.	Documento (Formato Anexo J)
<i>Requerimientos No Funcionales</i>	Incluye los estándares, regulaciones y contratos a los cuales el producto debe responder, así como los atributos de calidad y desempeño.	Documento (Formato Anexo K)

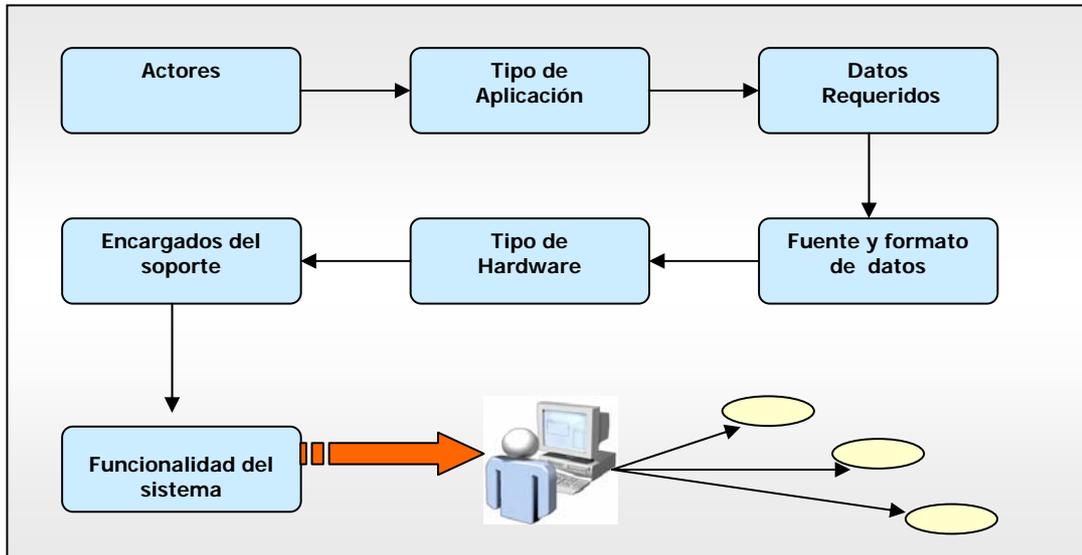
**Tabla 14. Niveles de Requerimiento**

Para el modelamiento de los requerimientos de usuario se plantea utilizar Casos de Uso, para lo cual se recomienda tener en cuenta una especificación en detalle de los siguientes factores:

1. La persona o entidad que será el actor de la aplicación, es decir quien la usará.
2. Tipo de aplicación: De tipo Internet o Intranet.
3. Los datos que son requeridos para los servicios en línea.
4. La fuente de datos y el formato de los mismos.
5. El tipo de hardware de red que los usuarios tienen.
6. Quien dará soporte al sistema.
7. La funcionalidad que el sistema debe incluir.

Cuando se establecen estos factores con claridad, se puede comenzar a construir la aplicación Internet, lo cual incluye la creación de servicios y el diseño del sitio Web.

El esquema de la Figura 12 muestra el flujo de factores que se deben tener en cuenta en el modelamiento de los casos de uso mencionados anteriormente.



**Figura 12. Factores de modelamiento de casos de uso**

### 3.3.2.1. Requerimientos Funcionales

Los requerimientos funcionales para la guía metodológica se establecerán con base en los estándares descritos por el OGC para los servicios:

- ⇒ WMS (Web Mapping Service)
- ⇒ WFS (Web Feature Service)

Teniendo en cuenta que estos son los servidores que harán parte de la arquitectura que se planteará en la fase de diseño.

Los servidores WMS y WFS se implementarán y prestarán los servicios que se describirán a continuación con la notación UML para Casos de Uso, utilizando el formato que se encuentra en el Anexo I.

#### 3.3.2.1.1. Web Map Service (WMS)

El usuario de los Servicios Web puede realizar las siguientes funciones con un Web Map Service:

⇒ Obtener Capacidades:

Solicitar información sobre los servicios y parámetros aceptables para las solicitudes. Ver profundización en el numeral 2.5.1.1.1.

<b>FACTORES</b>	<b>DESCRIPCIÓN</b>
<i>Identificación</i>	CU_WMS_GETCAPABILITIES
<i>Actor</i>	Usuario del sistema
<i>Nombre del Caso de Uso</i>	Obtener Capacidades
<i>Meta(Objetivo)</i>	Solicitar al servidor WMS información acerca de los servicios, capas y parámetros aceptables para las solicitudes.
<i>Pre Condiciones</i>	Los servicios que se describen en el documento XML devuelto por el servidor deben existir, así como las capas descritas en el mismo.
<i>Flujo Normal de Eventos</i>	<ol style="list-style-type: none"> <li>1) El usuario del sistema escoge la opción GetCapabilities.</li> <li>2) El Web Service (Adaptador) decide a quien va dirigida la petición en este caso WMS.</li> <li>3) El servidor WMS devuelve un documento XML con las capacidades (servicios) disponibles al igual que las capas.</li> </ol>
<i>Puntos de Extensión</i>	No tiene
<i>Caminos de Excepción</i>	No tiene
<i>Post Condiciones</i>	El resultado de la operación debe ser un XML con el detalle de los servicios prestados por el servidor.

⇒ Obtener un Mapa:

Solicitar una imagen de un mapa cuyos parámetros geoespaciales y dimensionales fueron definidos. Ver profundización en el numeral 2.5.1.1.2.

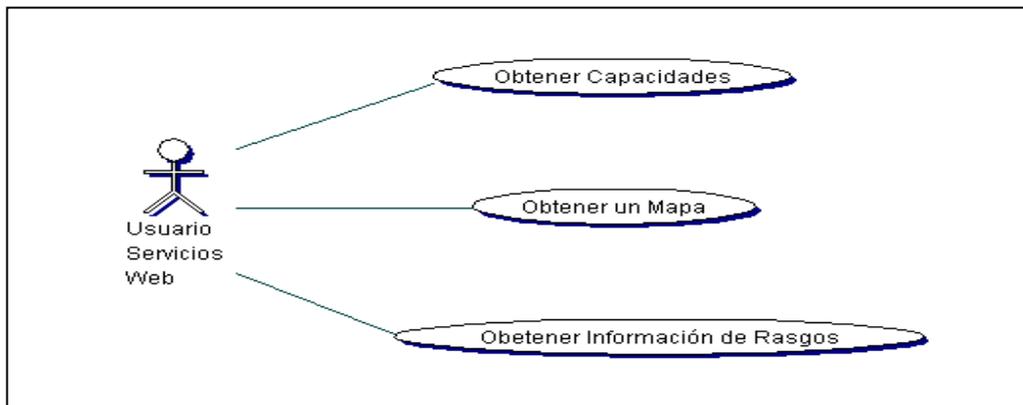
<b>FACTORES</b>	<b>DESCRIPCIÓN</b>
<i>Identificación</i>	CU_WMS_GETMAP
<i>Actor</i>	Usuario del sistema
<i>Nombre del Caso de Uso</i>	Obtener un Mapa
<i>Meta(Objetivo)</i>	Obtener un mapa con las capas que hayan sido seleccionadas.
<i>Pre Condiciones</i>	- Las capas listadas deben existir en el servidor WMS. - El servidor WMS debe prestar el servicio de GetMap.
<i>Flujo Normal de Eventos</i>	1) El usuario del sistema selecciona una capa. 2) Agrega la capa al mapa. 3) El Web Service solicita la capa al servidor WMS. 4) El servidor WMS devuelve un GML con la capa seleccionada. 5) El Web Service transforma el GML por medio de una hoja de estilo en SVG y lo envía al cliente.
<i>Puntos de Extensión</i>	No tiene
<i>Caminos de Excepción</i>	En caso de que en el paso 1 el usuario no seleccione una capa se mostrará un mensaje que indique que se debe escoger una capa para hacer la solicitud.
<i>Post Condiciones</i>	El resultado debe ser una imagen con formato SVG que contenga la capa seleccionada.

⇒ Obtener información de Rasgos:

Solicitar la información acerca de rasgos particulares que se muestran en el mapa. Ver profundización en el numeral 2.5.1.1.3.

<b>FACTORES</b>	<b>DESCRIPCIÓN</b>
<i>Identificación</i>	CU_WMS_GETFEATUREINFO
<i>Actor</i>	Usuario del sistema
<i>Nombre del Caso de Uso</i>	Obtener información de Rasgos
<i>Meta(Objetivo)</i>	Obtener la información de los rasgos de los mapas o capas que existan, esta es de carácter espacial.
<i>Pre Condiciones</i>	- Debe existir por lo menos un mapa en el servidor. - El mapa debe contener información de este tipo para que pueda ser consultada.
<i>Flujo Normal de Eventos</i>	1) El usuario del sistema selecciona el mapa del cual desea obtener la información de rasgos. 2) El Web Service solicita la información al servidor WMS. 3) El WMS devuelve la información en un documento. 4) El Web Service devuelve finalmente la información solicitada.
<i>Puntos de Extensión</i>	No tiene
<i>Caminos de Excepción</i>	No tiene
<i>Post Condiciones</i>	Al final de la operación el usuario habrá obtenido el documento de información de rasgos del mapa que seleccionó.

En la Figura 13 se presenta el esquema de los casos de uso para el actor: Usuario de Servicios Web de WMS.



**Figura 13. Esquema Casos de Uso WMS**

#### **3.3.2.1.2. Web Feature Service (WFS)**

Para el caso de la guía metodológica las funciones que prestará el Web Feature Service serán la de uno básico el cual se describe a continuación

##### Basic WFS

Un WFS básico implementa las operaciones:

- GetCapabilities
- DescribeFeatureType
- GetFeature

Estas funciones se consideran como servicios web feature de lectura únicamente.

El usuario de los Servicios Web puede realizar las siguientes funciones con un Web Feature Service:

⇒ Solicitar Capacidades:

Solicitar la definición de las capacidades del WFS. Específicamente, solicitar cuales tipos de rasgos puede servir y que operaciones son soportadas en cada tipo de rasgo. Ver profundización en el numeral 2.5.2.2.1.

<b>FACTORES</b>	<b>DESCRIPCIÓN</b>
<i>Identificación</i>	CU_WFS_GETCAPABILITIES
<i>Actor</i>	Usuario del sistema
<i>Nombre del Caso de Uso</i>	Solicitar Capacidades
<i>Meta(Objetivo)</i>	Obtener la definición de las capacidades del WFS. Específicamente, solicitar cuales tipos de rasgos puede servir y que operaciones son soportadas en cada tipo de rasgo.
<i>Pre Condiciones</i>	Los servicios que se describen en el documento XML devuelto por el servidor deben existir, así como los rasgos descritos en el mismo.
<i>Flujo Normal de Eventos</i>	<ol style="list-style-type: none"> <li>1) El usuario del sistema selecciona el mapa del cual desea obtener la información de rasgos.</li> <li>2) El Web Service solicita la información al servidor WFS.</li> <li>3) El WFS devuelve la información en un documento.</li> <li>4) El Web Service devuelve finalmente la información solicitada.</li> </ol>
<i>Puntos de Extensión</i>	No tiene
<i>Caminos de Excepción</i>	No tiene
<i>Post Condiciones</i>	El resultado de la operación debe ser un XML con el detalle de los servicios prestados por el servidor.

⇒ Solicitar la Descripción del Tipo de Rasgos:

Solicitar la descripción de la estructura de cualquier tipo de rasgo que el servidor pueda servir. Ver profundización en el numeral 2.5.2.2.2.

<b>FACTORES</b>	<b>DESCRIPCIÓN</b>
<i>Identificación</i>	CU_WFS_DESCRIBE_FEATURE_TYPE
<i>Actor</i>	Usuario del sistema
<i>Nombre del Caso de Uso</i>	Solicitar la Descripción del Tipo de Rasgos
<i>Meta(Objetivo)</i>	Solicitar la descripción de la estructura de cualquier tipo de rasgo que el servidor pueda servir.
<i>Pre Condiciones</i>	Debe existir información acerca de los rasgos en la base de datos del servidor WFS.
<i>Flujo Normal de Eventos</i>	<ol style="list-style-type: none"> <li>1) El usuario del sistema selecciona la opción describe feature type.</li> <li>2) El Web Service solicita la información al servidor WFS.</li> <li>3) El WFS devuelve la información en un documento XML.</li> <li>4) El Web Service devuelve finalmente la información solicitada.</li> </ol>
<i>Puntos de Extensión</i>	No tiene
<i>Caminos de Excepción</i>	En caso de que en el paso 2 el servidor WFS no tenga la información del rasgo solicitado, se devuelve una excepción.
<i>Post Condiciones</i>	El resultado de la operación debe ser un XML con el detalle de los rasgos solicitados, de acuerdo al esquema establecido por el OGC (Open GIS Consortium).

⇒ Solicitar Rasgos:

Especificar cuales propiedades del rasgo buscará y podrá mostrar de acuerdo a las restricciones de la consulta espacial y no espacial. Ver profundización en el numeral 2.5.2.2.3.

<b>FACTORES</b>	<b>DESCRIPCIÓN</b>
<i>Identificación</i>	CU_WFS_GETFEATURE
<i>Actor</i>	Usuario del sistema
<i>Nombre del Caso de Uso</i>	Solicitar Rasgo
<i>Meta(Objetivo)</i>	Obtener las propiedades del rasgo que se busca y poder mostrarlo de acuerdo a los constraint de la consulta espacial y no espacial
<i>Pre Condiciones</i>	Debe existir información acerca de los rasgos en la base de datos del servidor WFS.
<i>Flujo Normal de Eventos</i>	1) El usuario del sistema selecciona la opción GetFeature. 2) El Web Service solicita la información al servidor WFS. 3) El WFS devuelve la información en un documento XML. 4) El Web Service devuelve finalmente la información solicitada.
<i>Puntos de Extensión</i>	No tiene
<i>Caminos de Excepción</i>	En caso de que en el paso 2 el servidor WFS no tenga la información del rasgo solicitado, se devuelve una excepción.
<i>Post Condiciones</i>	El resultado de la operación debe ser un XML con el detalle de los rasgos solicitados, de acuerdo al esquema establecido por el OGC (Open GIS Consortium).

Para el caso de un WFS Transaccional se adicionan dos funciones, que se mencionarán pero no serán implementadas en el prototipo.

⇒ Hacer Transacción:

Crear, actualizar o borrar operaciones en un rasgo geográfico. Ver profundización en el numeral 2.5.2.2.4.

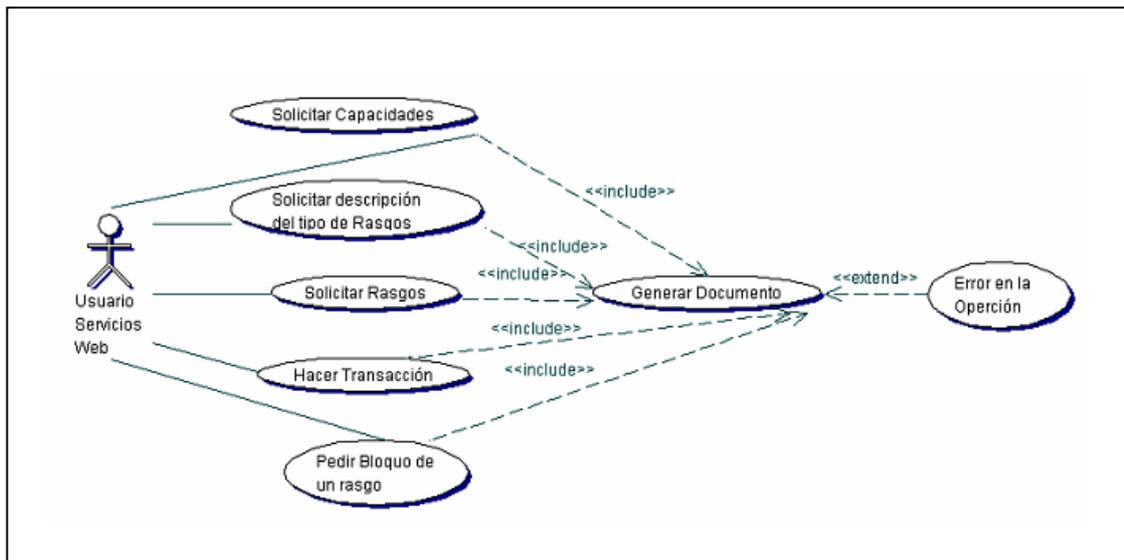
<b>FACTORES</b>	<b>DESCRIPCIÓN</b>
<i>Identificación</i>	CU_WFS_TRANSACTION
<i>Actor</i>	Usuario del sistema
<i>Nombre del Caso de Uso</i>	Hacer Transacción
<i>Meta(Objetivo)</i>	Crear, actualizar o borrar operaciones en un rasgo geográfico.
<i>Pre Condiciones</i>	Debe existir información acerca de los rasgos en la base de datos del servidor WFS.
<i>Flujo Normal de Eventos</i>	<ol style="list-style-type: none"> <li>1) El usuario del sistema selecciona la opción crear, actualizar o borrar.</li> <li>2) En caso de ser crear o actualizar deberá llenar los campos como bounding box, geometría, y otros datos adicionales.</li> <li>3) El Web Service hará el llamado correspondiente al servidor WFS.</li> <li>4) Se dirá al usuario si la operación fue exitosa o no.</li> </ol>
<i>Puntos de Extensión</i>	No tiene
<i>Caminos de Excepción</i>	Si en el paso 3 no se ingreso la información correcta del paso 2 aparecerá un mensaje de error indicando que falta información o que es incorrecta.
<i>Post Condiciones</i>	Se deberá realizar toda la transacción o de lo contrario se hará rollback, para que los datos queden consistentes.

⇒ Pedir bloqueo de un Rasgo:

Solicitar un candado para una o más instancias de un rasgo, por el período que dure la transacción, esto con el fin de asegurar transacciones serializables. Ver profundización en el numeral 2.5.2.2.5.

<b>FACTORES</b>	<b>DESCRIPCIÓN</b>
<i>Identificación</i>	CU_WFS_LOCK_FEATURE
<i>Actor</i>	Usuario del sistema
<i>Nombre del Caso de Uso</i>	Pedir bloqueo de un Rasgo
<i>Meta(Objetivo)</i>	Obtener un candado para una o más instancias de un rasgo, por el período que dure la transacción, esto con el fin de asegurar transacciones serializables.
<i>Pre Condiciones</i>	Debe existir información acerca de los rasgos en la base de datos del servidor WFS.
<i>Flujo Normal de Eventos</i>	<ol style="list-style-type: none"> <li>1) El usuario del sistema selecciona la GetLockFeature sobre un rasgo.</li> <li>2) El Web Service hará el llamado correspondiente al servidor WFS.</li> <li>3) El servidor WFS bloqueará el recurso o rasgo solicitado y entregara el ID al Web Service.</li> <li>4) Por ultimo el Web Service entregara el identificador del candado.</li> </ol>
<i>Puntos de Extensión</i>	No tiene
<i>Caminos de Excepción</i>	No tiene
<i>Post Condiciones</i>	Al final de la operación se debe haber obtenido el candado sobre el recurso para lograr una transacción serializable.

En la Figura 14 se presenta el esquema de los casos de uso para el actor: Usuario de Servicios Web de WFS.



**Figura 14. Esquema Casos de Uso WFS**

### 3.3.2.2. Requerimientos no Funcionales

Para el desarrollo de la aplicación se deben tener en cuenta factores tanto funcionales como no funcionales, entre estos últimos es importante tener en cuenta los siguientes aspectos:

#### 3.3.2.2.1. Alta disponibilidad

Deben tenerse en cuenta los posibles fallos en cada componente de la arquitectura. Esto tendrá un impacto en las decisiones que se tomen, desde la fase del diseño de la arquitectura, hasta factores como, la tecnología, la topología y los productos que se utilizarán para crear la solución completa.

Una vez definido el conjunto de la solución para satisfacer los requisitos de alta disponibilidad, se debe identificar las dependencias en las que puede influir a corto plazo y en las que no. Los conocimientos del personal y la disponibilidad del soporte de los proveedores son activos vitales para dar soporte a los requisitos de alta disponibilidad de la configuración de la aplicación generadora de servicios en línea geográficos.

Al planificar la alta disponibilidad del entorno en tiempo de ejecución y comprobar las configuraciones de alta disponibilidad existentes, deben aplicarse tres reglas básicas:

1. Todas las entidades deben ser redundantes.
2. Todas las entidades deben supervisarse por si sufren alguna anomalía.
3. Una entidad que sufre una anomalía no debe recibir trabajo.

Los términos de alta disponibilidad que están relacionados con las reglas descritas anteriormente se describen en la Tabla 15.

<b>Término</b>	<b>Definición</b>
<i>Detectar</i>	Supervisión continua de la configuración por si tiene lugar una anomalía o algún otro problema en los componentes.
<i>Recuperar</i>	Utilización de entidades redundantes para seguir procesando en caso de anomalía en los componentes.
<i>Decidir</i>	Tomar la decisión de no enviar trabajo a una entidad que sufre una anomalía y, a continuación, decidir el envío de trabajo a la entidad cuando esté reparada.
<i>Reparar</i>	Solventar el problema del componente y volverlo a conectar.

**Tabla 15. Términos de Alta Disponibilidad**

Para que las entidades contribuyan a la alta disponibilidad, deben agruparse de una de las formas mencionadas a continuación:

⇒ Par de alta disponibilidad

Algunos productos tienen la capacidad de actuar de dos en dos, cumpliendo las tres reglas de alta disponibilidad. Existen dos grupos de pares. Par Principal/Reserva y par Iguales. Con el tipo Principal/Reserva, uno hace el trabajo (función Principal) mientras que el otro lo supervisa (función de Reserva o en espera). El de Reserva toma el control si se detecta una anomalía. En una configuración de alta disponibilidad de igualdad, ambos miembros están en funcionamiento y se supervisan mutuamente. En caso de detección de una anomalía, toman el control del trabajo de su igual, lo cual es aplicable para cada uno de los servidores WMS y WFS.

⇒ Clúster

La mayoría de los productos que se agrupan en un clúster no tienen la capacidad de supervisar a sus iguales. Normalmente, no son conscientes de que están en un grupo. Necesitan otro producto que distribuya los

requerimientos entre ellos y que cumpla las reglas de alta disponibilidad 2 y 3, listadas anteriormente. Todos los miembros de un clúster deben poder enviar los mismos requerimientos. Normalmente, pueden enviar varios requerimientos de forma simultánea, lo que permite asignarles distintos niveles de carga, por ejemplo cuando se solicitan varias capas de un mapa, que son capas pesadas, se puede distribuir la carga entre los servidores del cluster que provean las capas solicitadas.

### Distribución de requerimientos

Para distribuir requerimientos a los miembros de un clúster, debe haber una entidad que esté al tanto de todos los miembros. Esta entidad debe cumplir las reglas 2 y 3 de alta disponibilidad. Si una entidad de distribución se amplía con la capacidad de calcular la carga de los miembros del clúster y distribuirla equitativamente entre ellos, recibe el nombre de balanceador de carga.

Hay dos configuraciones distintas que utilizan balanceadores de carga: la configuración en clúster y la configuración de alta disponibilidad.

#### ⇒ *Configuración en clúster*

En una configuración en clúster, varias entidades que no se reconocen entre ellas realizan funciones de redirección o de equilibrado básico de carga. Por lo tanto, están agrupadas en clústeres. Para cumplir las reglas 2 y 3 de alta disponibilidad, estas entidades necesitan otro producto. Cada entidad tiene las siguientes funciones:

- Supervisar a los miembros de un clúster secuencialmente y detectar una anomalía.
- Equilibrar los requerimientos entre los servidores, para el caso de esta guía, WMS y WFS secuencialmente de un clúster determinado.

#### ⇒ *Configuración de alta disponibilidad*

En una configuración de alta disponibilidad, los balanceadores de carga actúan en pares que pueden cumplir las tres reglas de alta disponibilidad. Cada balanceador de carga tiene las siguientes funciones:

- Capacidad de supervisar a su balanceador de carga homólogo y de detectar una anomalía.
- Capacidad de tomar el mando del equilibrado de su balanceador homólogo.

- Capacidad de supervisar a los miembros de un clúster de balanceadores de carga secuencialmente y de detectar una anomalía.
- Capacidad de equilibrar los requerimientos entre los servidores secuencialmente de un clúster de balanceadores de carga determinado.

### 3.3.2.2.2. Alto desempeño

En la actualidad, el desempeño no es sólo deseable, sino que es una obligación para sobrevivir en Internet y para superar la competencia, sobre todo cuando se trata del manejo de información que es de utilidad para varios entes tanto internos como externos a una organización, como es el caso de la información geográfica.

La necesidad de disponer de un buen desempeño es evidente: el tiempo de respuesta debe ser lo más bajo posible para los clientes que acceden al sitio Web, y que esperan eficiencia en la generación de los servicios WMS y WFS.

#### Factores que inciden en el desempeño

El desempeño se basa en dos factores bien diferenciados. El primero es la cantidad mínima de tiempo necesaria para procesar una tarea determinada, que se basa en el tipo de tarea y en el diseño de la aplicación. El segundo aspecto se centra en mantener siempre al mismo nivel la cantidad de tiempo necesaria para realizar cada tarea aunque el número de requerimientos para dicha tarea aumente.

En este sentido, el desempeño se puede ver como el resultado de la interacción de los factores mencionados a continuación:

- Tiempo
- Recursos
- Carga de trabajo

Conceptualmente, estos tres factores tienen una relación triangular de interdependencia. Cada uno de ellos es el resultado de los otros dos. En un sitio Web, el factor de tiempo está representado por el tiempo de respuesta objetivo:

$$\text{Tiempo} = \text{Carga de trabajo} / \text{Recursos}$$

Para manipular el factor de tiempo de esta ecuación se utilizan tres enfoques:

⇒ *Aumentar los recursos*

Éste enfoque permite aumentar la producción. La condición que debe cumplirse es la escalabilidad, tanto de la aplicación como del entorno de tiempo de ejecución. La escalabilidad del entorno de tiempo de ejecución incluye la alta disponibilidad, que se consigue mediante una arquitectura que admite el equilibrio de carga y la agrupación en clúster de los recursos.

La escalabilidad de la aplicación depende del diseño.

⇒ *Reducir la carga de trabajo*

Puesto que el tiempo necesario para realizar una tarea determinada viene definido por su naturaleza y por el diseño de la aplicación, desde aquí no se puede tener ninguna influencia sobre él. Sin embargo, se puede intentar reutilizar el resultado de dicha tarea. Para hacerlo, se utiliza la técnica de colocación en el caché. Actualmente, no sólo se puede colocar en el caché el contenido estático, sino también los resultados dinámicos del proceso de tareas, por ejemplo para el caso de las capas de un mapa que el cliente ya ha solicitado.

⇒ *Aumentar el tiempo disponible*

Se puede dividir la carga de trabajo en subtareas cuya realización requiere distintas funciones y distribuirlas entre instancias especializadas. De este modo, se pueden procesar simultáneamente y el tiempo disponible total aumenta. Este proceso estará limitado o potenciado por un gran número de factores, como el diseño de la aplicación, el empaquetado de la aplicación, la topología del tiempo de ejecución y los conocimientos disponibles entre otros.

Por otra parte, para lograr un alto desempeño se recomienda el uso de servidores proxy, los cuales tienen la capacidad de actuar como estación de retransmisión para los clientes, es decir que es la ubicación ideal para guardar los resultados de los requerimientos en el caché y reutilizarlos, porque reenvía los requerimientos y recibe las respuestas. Este tipo de servidor proxy recibe el nombre de servidor proxy caché, y ofrece las siguientes ventajas:

- ⇒ Se reduce el tiempo de respuesta para los requerimientos de los clientes.
- ⇒ Se reduce la carga en los nodos de servidor de contenido.
- ⇒ Se utiliza un ancho de banda de red inferior entre los nodos de servidor de contenido.
- ⇒ Al liberar recursos, se consigue una mayor escalabilidad.

Para optimizar el tiempo de respuesta se plantea también el uso de memoria caché sobre los resultados de las operaciones WMS y WFS, teniendo en cuenta que la memoria es el medio más rápido para grabar y leer datos. Sin embargo, existen limitaciones en el tamaño de la memoria que un servidor puede gestionar, y es por ello que se recomienda el uso de memoria caché para la modificación frecuente de datos de pequeño o mediano tamaño. Para el caso de los servicios WMS y WFS, en caché se sugiere que sean almacenados los resultados de cada una de las operaciones que el cliente solicite más de una vez. Para llevar a cabo dicho fin, es indispensable diferenciar las dos técnicas para colocación de información en caché que se describen a continuación:

#### Colocación estática en caché

Cuando un cliente solicita un mapa desde un servidor Web, éste debe encontrarlo, leerlo desde el disco y enviarlo a la red. Las operaciones de E/S de disco están entre las operaciones en cuya realización el servidor Web invierte más tiempo. Guardar un archivo de este tipo en la memoria y enviarlo desde ésta reduce drásticamente el tiempo de respuesta para todos los requerimientos subsiguientes. La implementación de una capa de colocación en caché frente a los clústeres de servidores Web optimiza la utilización del caché y su tasa de aciertos.

#### Colocación dinámica en caché

La colocación en el caché de los resultados de operaciones para presentación dinámica de datos, recibe el nombre de colocación dinámica en caché. Reduce el tiempo de respuesta para el cliente y también la carga, tanto en los nodos de servidor Web como en los nodos de servidor de aplicaciones. Asimismo, también reduce el ancho de banda entre los nodos de servidor de contenido.

### **3.4. Fase de Diseño**

La fase de diseño, para el caso de la generación de los servicios WMS y WFS, está basada en el análisis previo de los requerimientos funcionales y no funcionales obtenidos por medio de la aplicación de los formatos adecuados.

Adicionalmente al análisis de requerimientos, se tienen en cuenta las herramientas y los factores internos y externos que se requieren para el desarrollo de una arquitectura que soporte la eficiencia y la seguridad en la generación de los servicios WMS y WFS.

En esta fase se establece el comportamiento dinámico del sistema, es decir, la manera como debe reaccionar ante cada una de las solicitudes del cliente, teniendo en cuenta que se regirá por los estándares para solicitud y respuesta en cada una de las operaciones WMS y WFS.

El resultado obtenido de la etapa de diseño facilita la implementación posterior del sistema generador de servicios en línea geográficos, en el sentido que proporciona la estructura básica del sistema de modo que se pueda comprender claramente la forma como los diferentes componentes actúan y se relacionan entre sí.

Uno de los factores que se recomienda tener en cuenta en tiempo de diseño, es el uso de patrones, teniendo en cuenta que éstos, ofrecen soluciones concretas a los posibles problemas que se puedan encontrar en la construcción del sistema, indicando la mejor forma de modelar los objetos.

La elección de la arquitectura interna es otro de los factores elementales en la etapa de diseño, porque de allí se desprende la eficiencia y la funcionalidad. La arquitectura que se sugiere se rige por un modelo genérico por capas: Presentación, Dominio y Acceso a datos. No obstante, al escoger la arquitectura dependerá también de la tecnología que se utilizará.

#### **3.4.1. Recomendaciones de diseño de servicios en línea**

Las siguientes son las recomendaciones que se plantean cuando se desea diseñar una aplicación que provea servicios en línea a los usuarios:

- ⇒ Identificar los tipos de componentes que se requieren utilizar en la aplicación, es decir los componentes que desempeñan tareas específicas, siguiendo un modelo de capas para desacoplar funcionalidad. En este punto es importante tener en cuenta el tamaño de la aplicación y la cantidad de información que se manejará.
- ⇒ El diseño de componentes de un tipo debe ser coherente, es decir que todos los componentes de un tipo sean agrupados con base en el mismo objetivo, y sean diseñando bajo el mismo modelo, lo que facilita el mantenimiento del diseño y la implementación.
- ⇒ Realizar un análisis de la forma de conectar los componentes o módulos, de tal forma que sean altamente cohesivos y levemente acoplados, lo que trae consigo la facilidad para el mantenimiento de los módulos. En este punto es relevante el uso de patrones de software en el momento de diseño, que permitan modelar de tal manera que se logre la eficiencia en los procesos, ya que los patrones ayudan a solucionar problemas de una

índole específica, a la vez que activan un diseño modelado a alto nivel para aplicaciones que así lo requieren.

- ⇒ Conservar claras las relaciones de comunicación, en cuanto a formatos de intercambio de datos, tanto para solicitudes como para respuestas, y en cuanto a los protocolos de comunicación que se van a utilizar para intercambiar información entre las diferentes capas de la aplicación. Para el caso de servicios en línea geográficos, la estandarización de la comunicación está basada en los formatos que plantean los estándares WMS, WFS y GML, los cuales describen cómo se deben estructurar los Request y los Response en cada una de sus operaciones. Dicha estandarización provee al sistema una mayor extensibilidad y robustez, si por ejemplo se quiere agregar otro servidor que provea otros servicios, el cual estará regido por los mismos estándares.
- ⇒ Teniendo en cuenta que se trata de servicios sobre Internet, son muchas las funciones que se deben suplir, tales como seguridad, administración, o procedimientos especializados, y es por ello que se debe abstraer el código de cada una de esta funcionalidad, porque esto facilita desde el diseño hasta el mantenimiento. Esto se realiza por medio de atributos, interfaces de programación de aplicaciones (API) de plataforma o componentes de utilidades que proporcionen acceso a la funcionalidad relacionada con las políticas, como la publicación de excepciones y la autorización de usuarios, entre otras.
- ⇒ Uno de los factores esenciales para la eficiencia de una aplicación Web es el modelo multicapa, por lo cual se recomienda en la fase inicial del proceso, determinar el tipo de capas que se desea aplicar, teniendo en cuenta el nivel de flexibilidad que debe tener un sistema de capas de alto nivel, en el cual no deben existir llamados y dependencias ascendentes, que impliquen problemas en cascada.

### **3.4.2. Fuentes de Datos**

Los datos geográficos, debido a su naturaleza y a su gran tamaño, requieren de cierta organización de acuerdo a varios factores, para facilitar su acceso, y de éste modo garantizar la eficiencia de las operaciones que sobre éstos se realicen.

Las formas que se sugieren para organizar las capas de datos en grupos lógicos, se describen en la Tabla 16.

Forma	Descripción
<i>Reusable</i>	Los mapas con frecuencia contienen capas que son principalmente usadas para referencia. Estas capas de datos referenciales son usadas repetidas veces sobre diferentes aplicaciones, es por ello que se debería crear un servicio que incluya sólo capas de datos referenciales.
<i>Dominio específico</i>	Cuando los datos son especializados, se debería considerar la asignación de éstos a su propio servicio. En esta categoría encajan: Datos colectivos, datos que requieren análisis extenso y datos para los cuales se quiere restringir el acceso.
<i>Aplicación específica</i>	Algunas capas de datos son siempre usadas juntas para soportar una aplicación especializada, por ejemplo líneas de agua, válvulas y metros, y por ello deberían estar guardadas en el mismo servicio.
<i>Frecuente cambio</i>	Las capas de datos que son actualizadas frecuentemente, deberían ser guardadas separadas de las capas que no son cambiadas con frecuencia. Esto previene la actualización frecuente de capas de datos afectadas que solo necesitan cambios infrecuentes.
<i>Detalle</i>	Las capas que contienen información detallada podían ser agrupadas juntas en el mismo servicio.

**Tabla 16. Ventajas Grupos lógicos en Capa de Datos**

### 3.4.3. Tipos de Servicios

Para determinar el tipo de servicio a crear, se deben tener en cuenta los requerimientos funcionales descritos en la fase de análisis, que como es sabido fueron tomados a partir de los estándares WMS y WFS y de allí se sabe que:

- Un WMS (Web Map Service): será el servidor de mapas.
- Un WFS (Web Feature Service): será el servidor de rasgos.

Con base en lo anterior se puede optar por implantar los siguientes servicios:

#### Para un WMS

1. Servicio de imagen, el cual serviría en los siguientes casos:  
La funcionalidad requerida se limita a una vista y consulta de un mapa.
2. Servicio de superposición de capas:

- Se requiere la visualización de una o más capas.
3. Servicio de georeferenciación:  
Se quiera obtener y visualizar una ubicación específica dentro de un mapa.

#### Para un WFS

1. Feature Service:
  - ⇒ La funcionalidad incluye interacción y análisis extenso por parte del usuario.
  - ⇒ Las capas de datos son usadas para un análisis especializado.
  - ⇒ Se desea conocer características de determinados rasgos sobre el mapa que se está consultando.

#### **3.4.4. Diseño de Base de Datos**

Los servicios en línea geográficos se generan con base en una información, que por lo regular se encuentra almacenada en una base de datos geográfica relacional. Un correcto diseño de base de datos, entre otros factores de diseño, permite garantizar un acceso a dato óptimo de modo que se garantice la integridad y la consistencia de los datos que allí se almacenan.

El modelo Entidad Relación se considera la primera aproximación a un modelo de datos, para lo cual se plantea seguir los siguientes pasos:

- Definición de las entidades relevantes.
- Ilustración de dichas entidades.
- Determinación de identificadores y atributos que pertenecen a cada entidad.
- Definición de las relaciones asociadas.

El enfoque que se propone para el modelo de datos se divide en 3 fases, determinadas a partir de la funcionalidad para cada una de las operaciones de los estándares WMS y WFS, éstas fases son: Modelo de capacidades de los servidores, Modelo de datos geográficos y Modelo de seguridad, los cuales se explican a continuación:

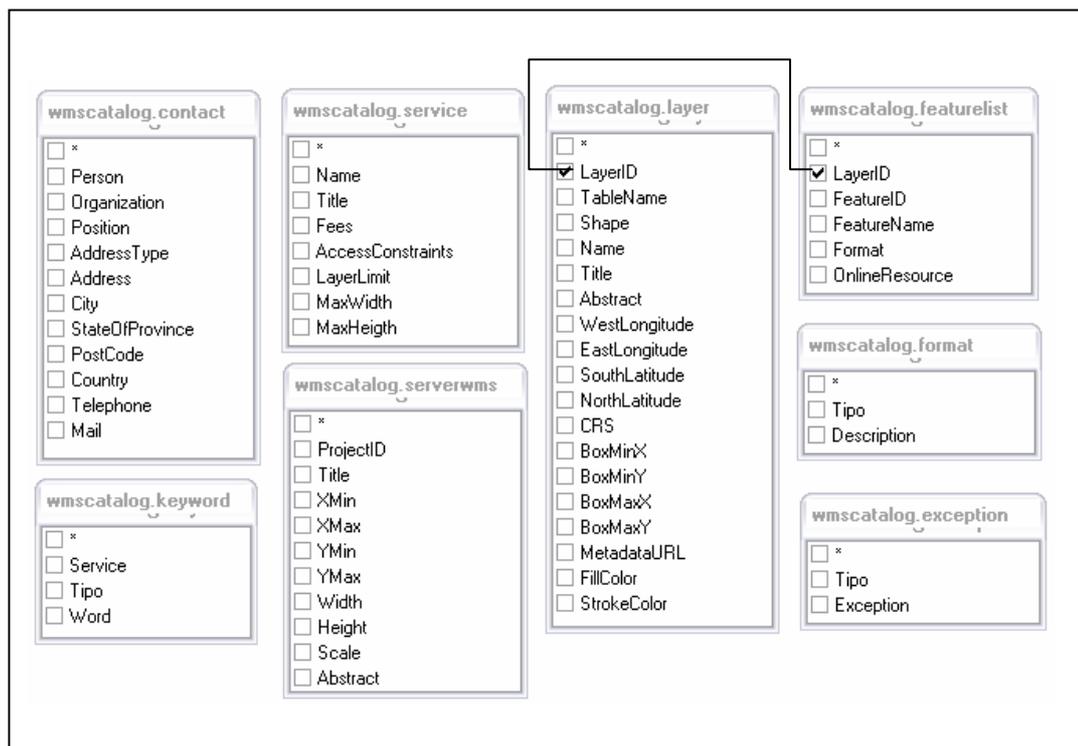
⇒ Modelo de datos para las Capacidades de los Servidores

Los servidores WMS y WFS soportan la operación básica llamada GetCapabilities, que despliega la información en formato XML de las capacidades, es decir de las operaciones que soporta cada servidor, especificando cierta información de contactos, capas, y formatos entre otros,

con base en un esquema(diagramado en el Anexo N), para cada uno de los servidores. El modelo Entidad Relación que se presenta en la Figura 15, representa el modelo de datos para especificar las capacidades de los servidores WMS y WFS.

Las entidades identificadas son:

- ☉ Contact: Información asociada al contacto del servidor.
- ☉ Service: Descripción del servicio.
- ☉ Layer: Información de cada capa del mapa.
- ☉ FeatureList: Descripción de los rasgos asociados a una capa.
- ☉ Format: Formato de entrada y salida de datos a cada operación
- ☉ ServerWMS: Características generales del servidor
- ☉ Keyword: Palabras clave de servidor y de capa.
- ☉ Exception: Posibles excepciones que se pueden lanzar.



**Figura 15. Modelo Entidad Relación para capacidades de los servidores**

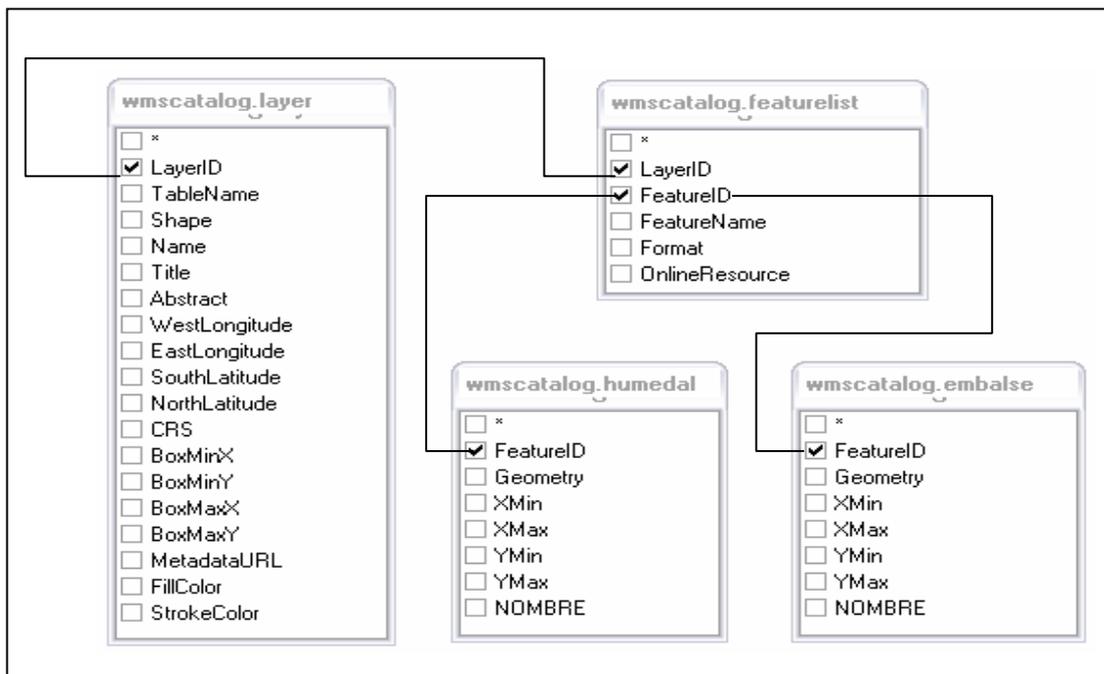
⇒ Modelo de datos para Datos Geográficos:

Los datos geográficos que se almacenan en una base de datos geográfica pueden ser de 2 tipos como se menciona en el numeral 2.1.4, datos espaciales y datos no espaciales. Para el caso de las capas, se plantea considerar a cada capa como una entidad, teniendo en cuenta que cada capa de un mapa puede tener diferentes atributos.

Por otro lado, se plantea establecer una entidad que represente las características generales de todas las capas y otra entidad que describa cada uno de los rasgos de las capas, y dicha entidad relacionarla con la entidad de cada una de las capas. La Figura 16 presenta el modelo Entidad Relación planteado para el manejo de éste tipo de información.

Las entidades identificadas son:

- 🌐 Layer: Información asociada a una capa.
- 🌐 FeatureList: Descripción de cada uno de los rasgos de las capas.
- 🌐 CapaX: Información de cada capa con datos de visualización que es la geometría y metadatos como nombre de cada rasgo.



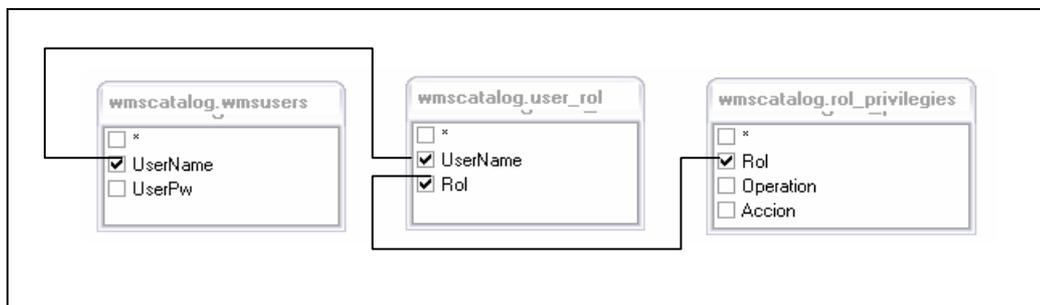
**Figura 16. Modelo Entidad Relación para Datos Geográficos**

⇒ Modelo de datos para Seguridad:

Las bases de datos geográficas deben ser particularmente seguras debido a la importancia y el grado de confidencialidad que tiene este tipo de información. Se plantea un modelo de seguridad en el cual al cliente se le asocia uno o varios roles, los cuales tienen privilegios sobre las operaciones que pueda desempeñar y las acciones sobre las mismas, es decir acceso a operaciones y acciones tales como CRUD (crear, consultar, actualizar o eliminar) datos. El modelo planteado se puede ampliar como se requiera con base en las necesidades de la organización para los servicios en línea, puede llegar a niveles de complejidad como restringir el acceso a ciertas capas de datos, rasgos de las capas de datos o incluso a metadatos asociados al mapa. La Figura 17 presenta el modelo Entidad Relación que se plantea para el manejo de este tipo de información, teniendo en cuenta que se puede volver tan complejo como se quiera, en la medida que se quiera manejar el acceso a la información en niveles más detallados como se mencionaba anteriormente.

Las entidades identificadas son:

- 🌐 Users: Información de usuarios con sus contraseñas.
- 🌐 Rol: Descripción de los roles asociados a los usuarios.
- 🌐 Rol\_Privileges: Descripción de los privilegios asociados a cada rol.



**Figura 17. Modelo Entidad Relación para Seguridad**

### 3.4.5. Diseño de la Arquitectura

Para el diseño de la arquitectura se plantea el uso de un modelo multicapa que permita la alta cohesión, el bajo acoplamiento, y por lo tanto eficiencia y seguridad, propias de una aplicación generadora de servicios en línea.

#### 3.4.5.1. Factores que debe proveer la arquitectura

Se plantea un modelo multicapa, que provea factores descritos en Tabla 17.

FACTORES	DESCRIPCIÓN
<i>Mantenibilidad</i>	Se representa a través de la capacidad que tiene el modelo para ser analizado en cuanto a: diagnósticos de problemas, causas de fallos en el software o identificación de partes que se deben modificar.
<i>Modularidad</i>	“Es la posibilidad de subdividir una aplicación en piezas más pequeñas denominadas módulos, cada una de las cuales debe ser tan independiente como sea posible, considerando la aplicación como un todo” <sup>19</sup> Se representa a través de la utilización de módulos y patrones de software que permiten que el software sea más fácilmente integrado, así como comprendido en su dimensión general y especializada.
<i>Reutilización (reusabilidad)</i>	“La reutilización es un medio para ahorrar esfuerzo en tiempo y espacio en el diseño de programas. Se basa en el concepto de componente, considerado un fragmento de programa que puede ensamblarse a otros componentes para producir software de un modo más fiable y eficiente” <sup>20</sup> . Los componentes de software están representados por las clases y las variantes de las mismas, y forman parte de módulos que a su vez forman parte de cada una de las capas de la arquitectura multicapa.

**Tabla 17. Factores que provee la arquitectura**

#### 3.4.5.2. Sistema Distribuido

En la actualidad los volúmenes de información que maneja la industria de los sistemas de información geográfica, tiene un crecimiento exponencial, de modo que obliga a una arquitectura que soporta el manejo de esta información, sea lo suficientemente eficiente para proveer las funciones que surgen a partir de la información.

Para ello se plantea un sistema distribuido manejado por capas en el cual se pueden tener servidores WMS y WFS ubicados en diferentes puntos geográficos, que proveen funciones especializadas según la información que se provea, por medio del uso de objetos remotos para cada uno de los servidores.

El servicio Web, como adaptador invoca remotamente a los métodos que necesita, y de ésta manera no obliga a tener una infraestructura localizada en un solo sitio, lo cual puede ser un factor de riesgo y de costo para la organización cuando se trata de información con cierto grado de confidencialidad.

---

<sup>19</sup> Ibidem 13

<sup>20</sup> Ibidem 13

Teniendo en cuenta que un sistema distribuido requiere flujo de información por medio de la comunicación entre objetos situados geográficamente en diferentes puntos, pero unidos bajo una misma red como lo es Internet, surgen factores como eficiencia y seguridad, que es uno de los objetivos de ésta guía metodológica, los cuales se explicarán más adelante.

### 3.4.5.3. Patrones de Software

Teniendo en cuenta que un patrón de software es considerado como una descripción detallada de una solución adecuada a un problema concreto, el objetivo básico que se persigue con la idea de patrón es no tener que reinventar respuestas cada vez que se surge la necesidad de utilizar dichas soluciones. Los patrones de software se encargan de capturar soluciones a situaciones de especial interés, por ser tanto complejas como comunes.

Un patrón está compuesto por el enunciado del problema y una o varias propuestas de solución, y para formalizar estas soluciones, se requiere una notación expresiva y rica en semántica que permita crear los atributos de las mismas. La notación más extendida es UML (Unified Modeling Language) el cual permite capturar con precisión las soluciones a problemas de todo tipo. La eficiencia es un objetivo fundamental en ésta guía metodológica, para lo cual se recomienda hacer uso adecuado de los siguientes patrones de software, que aportan a la arquitectura ciertos niveles de eficiencia y de extensibilidad según su función.

⇒ Data Access Object (DAO):

Patrón que abstrae y encapsula todos los accesos a la fuente de datos. Maneja la conexión con la fuente de datos para obtener y almacenar datos, también implementa el mecanismo de acceso necesario para trabajar con la fuente de datos, la cual puede ser un almacenamiento persistente como una RDMBS. El DAO es aplicable para encapsular el acceso a los datos de la base de datos geográfica que da soporte a la generación de los servicios en línea para WMS y WFS. Se recomienda utilizar el patrón DAO para el acceso a los datos de la fuente de información de cada uno de los servidores.

⇒ Abstarct Factory:

Provee una interface para crear objetos de alguna “fábrica”, sin especificar la clase en concreto. En la implementación de los servicios en línea, se utiliza para crear objetos de acceso a datos como conexiones y comandos de objetos de un RDBMS específico, es decir que permite la creación de fábricas de un proveedor específico que implementen esta interface, lo cual

induce a un desacoplamiento del proveedor de datos. Abstract Factory se puede considerar como una fábrica de objetos que retorna una de las fábricas disponibles.

⇒ Factory:

Retorna una instancia de una de varias clases posibles dependiendo de los datos proveídos por ésta. Usualmente todas las clases que éste retorna tienen una clase padre común y métodos comunes, pero la diferencia radica en que cada una de dichas clases desempeña una tarea diferente y es optimizada para diferentes clases de datos.

La abstracción que se recomienda analizar para el uso de éste patrón es la creación de una fábrica para los objetos de un RDBMS específico y otra fábrica para clases de acceso a datos especializadas, es decir para crear objetos como capas de cada servidor WMS y WFS, objetos para obtener las capacidades de cada servidor y objetos para la lógica asociada a la seguridad para cada uno de los servidores.

⇒ Adapter:

Provee una interface que conoce a sus clientes y proporciona acceso a una instancia de una clase que no conoce a sus clientes, es decir que convierte la interface de una clase en una interface que el cliente espera. Un objeto Adapter proporciona la funcionalidad prometida por una interface sin necesidad de conocer la clase que es utilizada para implementar esa interface. Permite trabajar conjuntamente a dos clases con interfaces incompatibles.

La implementación en la generación de los servicios en línea consiste en un Web Service que es con quien se comunica el cliente, el cual al solicitar los servicios al adaptador, especifica el tipo de servicio que desea, para lo cual el Web Service se adapta a dicha solicitud e invoca al servidor correspondiente, ya sea WMS o WFS para solicitar el servicio indicado, y de ésta manera poder devolver al cliente una respuesta adecuada.

⇒ Singleton:

Patrón que asegura que hay una y sólo una instancia de un objeto, y que es posible obtener un acceso global a tal instancia. Para el caso de la implementación, se recomienda usar éste patrón para el objeto DAO, para asegurar un solo punto de acceso al motor de base de datos para cada servidor WMS y WFS.

#### 3.4.5.4. Eficiencia y Seguridad

La eficiencia y la seguridad son 2 factores esenciales en ésta guía metodológica, para lo cual se explica a continuación la forma que se propone para conseguir cada uno de ellos:

##### **EFICIENCIA**

La eficiencia del sistema generador de servicios geográficos, se ve representada en 2 elementos: Tiempo y Calidad. La idea es conseguir una respuesta al cliente en el menor tiempo posible, y brindarle información de alta calidad con base en las solicitudes que el cliente realice. Para conseguir la eficiencia, se han planteado los siguientes elementos, cada uno de los cuales en cierta medida aporta para que el sistema sea eficiente.

➤ Modelo Multicapa:

El modelo que se plantea por capas, implica eficiencia en el sistema en el sentido que se abstrae funcionalidad por capas, lo cual implica mayor eficiencia en los procesos ya que cada capa se especializa en sus funciones.

Además, éste modelo provee facilidad en el mantenimiento del software, a pesar de que se tiene cierta complejidad en la comunicación entre las capas, para lo cual se establecen parámetros de comunicación, que permitan que el flujo de información entre capas sea liviano, y con ello reducir el tráfico en la red.

➤ Patrones de Software:

El uso de patrones de software en el desarrollo de cualquier aplicación busca solucionar problemas con cierta complejidad de la mejor forma. En el caso de los servicios en línea, se determinó el uso de patrones de software con el objetivo de hacer el mejor uso de los recursos de software que se tienen, como ofrecer mantenibilidad y reutilización del código. De igual forma, los patrones que se recomiendan usar como se mencionó anteriormente, cada uno cumple su objetivo específico, al mismo tiempo que provee reducción de la complejidad que se presenta en cada uno de los procesos del desarrollo del software.

➤ Memoria Caché:

El manejo adecuado de la memoria caché puede proveer mucha eficiencia, en el sentido que ahorra trabajo y tiempo en el flujo de la información, que ya una vez ha sido generada. Cuando un cliente solicita una capa de un mapa

más de una vez, la idea es que la primera vez, se realiza todo el proceso de comunicación entre capas y funcionalidad requerida para obtener ésta información, tal como consulta y transformación de datos, pero una vez obtenida, dicha información se almacena en memoria caché bajo una clave que identifica a ese elemento, y cuando el cliente solicite de nuevo dicha capa, el sistema simplemente la trae de la memoria y éste proceso es bastante rápido porque se tiene la información como si estuviera localmente y lista para ser desplegada al cliente, tal y como lo solicitó la primera vez.

➤ Estándares Open GIS:

Con base en las operaciones básicas de los estándares WMS y WFS, se tiene que tanto la solicitud de una operación, como la respuesta, son codificaciones de información en lenguaje de marcas, lo cual permite que el transporte de información sea rápido. La mayoría de los sistemas de información geográfica, devuelven al cliente los mapas en formato de imagen raster tal como JPG o GIF, los cuales son archivos bastante grandes, sobre todo cuando se trata de mapas geográficos, y debido al gran tamaño de éstos archivos, el tiempo de transmisión es mayor.

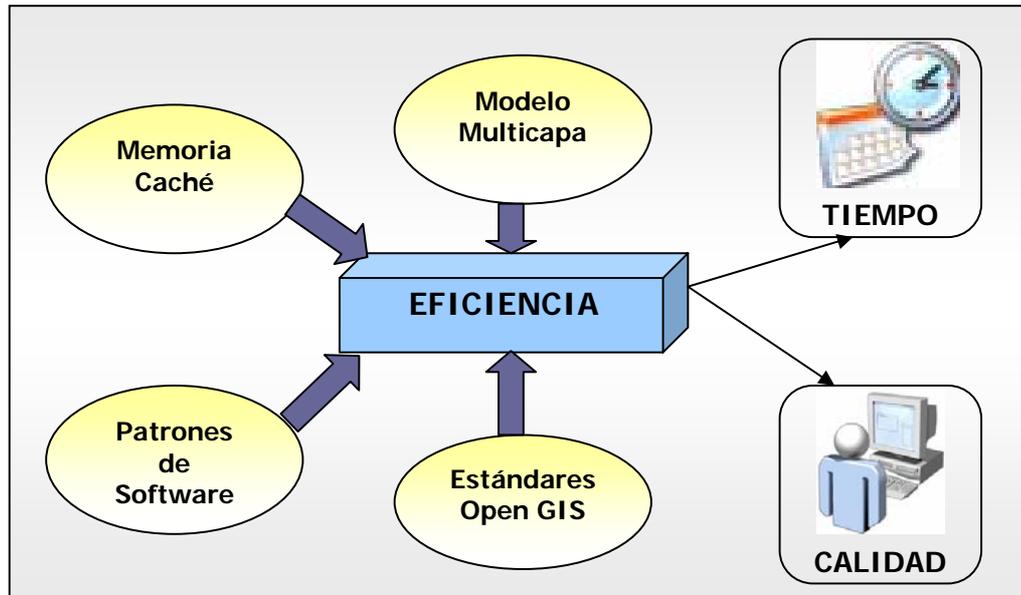
Lo que se propone en ésta guía metodológica es utilizar tanto para solicitudes y respuestas, como para procesamiento de información, estándares geográficos, especializados para el manejo de ese tipo de información, teniendo en cuenta que el Open GIS ha realizado muchas investigaciones en el tema y ha publicado especificaciones que facilitan el uso de la información geográfica en un sistema.

Para la manipulación de la información geográfica almacenada en una base de datos relacional, se recomienda el uso de GML como lenguaje de codificación de datos geográficos, el cual precisa específicamente la posición de los puntos, líneas y polígonos usados para representar los rasgos de interés en un mapa.

Es indispensable tener en cuenta que ésta codificación no es un formato que el cliente pueda visualizar, entonces se realiza una transformación de dicha información con hojas de estilos y se genera un formato de imagen que también es un lenguaje de marcas llamado SVG, el cual trae muchas ventajas mencionadas en el numeral 2.6.1.1.

Como resultado del uso de los estándares WMS, WFS y GML así como XML y SVG, se obtiene la información codificada al lado del cliente, el cual por medio de un servicio web, la interpreta de forma que la pueda visualizar. Por lo tanto, éste conjunto de procesos de solicitud, consulta, procesamiento, codificación, transformación y respuesta se torna más rápido teniendo en cuenta que se maneja de forma estandarizada.

La Figura 18 presenta un esquema simple que muestra los factores anteriormente mencionados, los cuales aportan para la eficiencia del sistema, representada en tiempo y calidad.



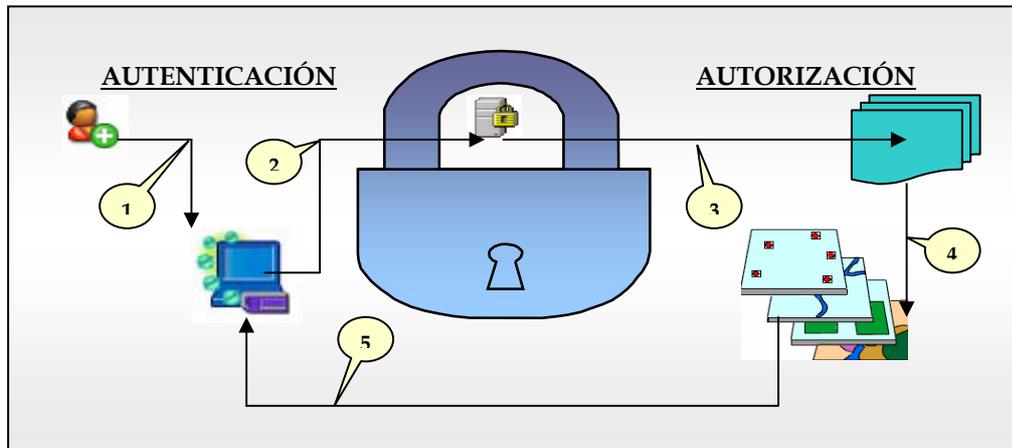
**Figura 18. Factores para conseguir eficiencia**

## **SEGURIDAD**

La seguridad de un sistema de información geográfica que genera servicios en línea, debe tener especial cuidado, teniendo en cuenta que la información que se publica puede ser accedida por toda la población Web si ésta no se encuentra adecuadamente protegida, y puede ser utilizada con fines malignos como por ejemplo terrorismo, robos, atentados entre otros.

Por dicha razón es indispensable en ésta etapa de diseño, establecer los roles adecuados para cada uno de los niveles de información que se tienen, para que los usuarios que accedan a la información le saquen el mayor provecho con fines por ejemplo de investigación, de construcción, o de análisis entre otros.

Los elementos que se establecen para el manejo de seguridad son autenticación y autorización, los cuales se diagraman en la Figura 19, y se explican a continuación.



**Figura 19. Modelo de Seguridad Servicios en Línea**

➤ Autenticación:

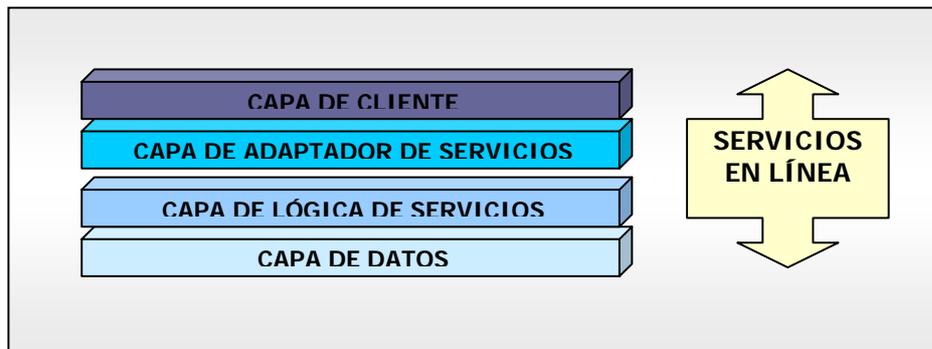
- (1) El usuario se autentica ante el sistema ingresando un nombre de usuario y una contraseña asociada.
- (2) El sistema envía dicha información a un servidor de autenticación, el cual valida la información ingresada por el usuario.

➤ Autorización:

- (3) El servidor de Autenticación analiza los permisos que tiene el usuario ingresado, para lo cual establece los privilegios sobre la información con base en los roles que tenga asociados ese usuario.
- (4) Se establecen los niveles de información para los cuales el usuario tiene acceso y las operaciones WMS y WFS que éste puede desempeñar.
- (5) Se informa al cliente las operaciones que puede desempeñar y la información que puede acceder.

**3.4.5.5. Modelo Multicapa:**

Se plantea un modelo de 4 capas el cual cumple con los requerimientos y factores esenciales como eficiencia y seguridad. La Figura 20 muestra un esquema general de las capas que se sugieren para el diseño de la arquitectura.



**Figura 20. Modelo de Capas**

El modelo multicapa que se plantea se ha denominado por la sigla **DLAC** (**Data, Service Logic, Service Adapter, Client**).

A continuación se describen cada una de dichas capas:

#### **3.4.5.5.1. Capa de Datos**

Esta capa busca encapsular la lógica asociada al acceso a los datos, los cuales deben estar soportados por un modelo de datos adecuado, teniendo en cuenta el modelo relacional de bases de datos.

Esta capa se encarga básicamente de proveer las funciones básicas de acceso a la base de datos: Consultar, eliminar, actualizar y agregar, así como toda la parte de conexiones y permisos de acceso. Para el caso del prototipo, se tienen solamente operaciones de consulta a la base de datos.

Los patrones de software aplicados en ésta capa son Abstract Factory, Singleton, Factory y DAO para proveer una independencia propia de los sistemas distribuidos.

Esta capa es la base de los servicios que se desean generar, ya que al definir correctamente el almacenamiento de los datos, se podrá obtener una estandarización de las fuentes para conseguir la interoperabilidad del sistema.

Se plantean las siguientes aproximaciones para conseguir lo anteriormente descrito:

- ⇒ *Integración de base de datos:* Se refiere a proveer a los usuarios un catálogo global de fuentes de información accesible, donde cada fuente

es descrita por metadatos asociados, incluyendo modo de representación, escala, fecha de última actualización, y nivel de calidad de los datos.

⇒ *Estandarización*: Se refiere a la definición de estándares para el modelamiento y la manipulación de los datos, para facilitar el intercambio de datos entre sistemas heterogéneos.

A continuación se presentan las precondiciones, los procedimientos generales y las postcondiciones que se deben cumplir en esta capa de datos:

*Precondiciones:*

- Modelo de datos relacional.
- Datos estandarizados
- Unidades de medida establecidas
- Datos suficientes para generar servicios WMS y WFS: Datos de las capacidades que ofrece cada uno de los servidores, y datos asociados a cada una de las capas de los mapas que se ofrezcan.

*Procedimientos:*

- Creación de objetos de acceso a datos de un RDBMS específico.
- Conexión a la base de datos.
- Manejo de permisos y roles a nivel de base de datos.
- Operaciones de consulta a la base de datos CRUD (Create – Recovery – Update – Delete).

*Postcondiciones:*

- Entrega de la información de manera estandarizada para la otra capa, en arreglos o cadenas de caracteres a partir de la información solicitada.

#### **3.4.5.5.2. Capa de Lógica de Servicios**

Esta capa se encarga del manejo de 2 servidores básicos: WFS Server y WMS Server, los cuales implementan las funciones básicas de cada uno.

Estos encapsulan la lógica necesaria para recibir solicitudes, realizar las operaciones requeridas y generar las salidas entendibles y funcionales para la otra capa. La lógica de los servicios consiste básicamente en la lógica del negocio, es decir en la implementación de los servidores remotos WMS y

WFS, los cuales hacen uso de la capa de datos para proveer la información solicitada.

Particularmente en la operación GetMap de WMS, éste servidor se encarga de generar la codificación del mapa solicitado a GML y enviarla a la otra capa de adaptador de servicios quien se encargaría de transformarla a formato entendible por el cliente.

A continuación se presentan las precondiciones, los procedimientos generales y las postcondiciones que se deben cumplir en esta capa de lógica de servicio para cada servidor WMS y WFS:

## **WMS**

### Precondiciones:

- Request WMS

### Procedimientos:

- Extracción de los datos del request.
- Llamado a procedimientos para obtener información de la Base de Datos.
- Implementación de los métodos: GetCapabilities, GetMap y GetFeatureInfo.

### Postcondiciones:

- Response WMS

## **WFS**

### Precondiciones:

- Request WFS

### Procedimientos:

- Extracción de los datos del request.
- Llamado a procedimientos para obtener información de la Base de Datos.
- Implementación de los métodos: GetCapabilities, GetFeature, DescribeFeatureType, Transaction y LockFeature.

Postcondiciones:

- Response WFS

### **3.4.5.5.3. Capa de Adaptador de Servicios**

Esta capa se encarga de encapsular la lógica que integra la información de los servicios proveídos por la capa anterior en un solo servicio. El propósito es proveer a la capa del cliente, la información codificada en lenguaje GML en caso de que sea una petición diferente a la operación GetMap y SVG en caso de que sea una petición de la operación anteriormente mencionada, para poder visualizarla.

Para implementar esta capa se recomienda el uso de Web Services, el cual será quien exponga toda la funcionalidad que prestan ambos servidores, tanto WMS (Web Map Service) como el WFS (Web Feature Service), para ello se tendrán que definir las funciones que cada servidor presta, en caso de ser un WMS y un WFS básico, se tendrían que definir las siguientes funciones:

Para WMS (Web Map Service):

- ✓ GetCapabilities
- ✓ GetMap

Para WFS (Web Feature Service):

- ✓ GetCapabilities
- ✓ DescribeFeatureType
- ✓ GetFeature

Estas funciones han sido descritas en los requerimientos de cada uno de los servidores con sus correspondientes parámetros de entrada y salida. Ver numerales 3.3.2.1.1 y 3.3.2.1.2 respectivamente para WMS y WFS.

El adaptador de servicios juega un papel elemental en el sistema, porque ésta capa es quien tiene comunicación directa con el cliente, el cual solicita las operaciones a un Web Service, el cual determina a cuál servidor remoto debe invocar para poder enviar la respuesta al cliente en el formato adecuado.

Particularmente para el caso de la operación GetMap de WMS, se tiene que el servidor remoto devuelve una codificación GML al adaptador, quien se encarga de transformar dicha codificación a un formato SVG entendible por

el cliente Web, por medio de hojas de estilos adecuadas para mostrar el mapa, de tal modo que el cliente pueda visualizar la información que éste trae.

#### **3.4.5.5.4. Capa de Cliente**

Esta capa representa la capa del usuario que va a interactuar directamente con el sistema generador de servicios en línea WMS y WFS. La labor de esta capa es proveer al usuario la interfaz adecuada para hacer uso de las funciones que provee cada uno de los servidores.

El cliente, como se mencionaba en los casos de uso, es quien se comunica con el sistema por medio de solicitudes que hace al mismo, las cuales deben llevar el formato adecuado para que puedan ser correctamente interpretadas por el servidor, y pueda recibir la respuesta adecuada en el formato adecuado, que sea entendible por el cliente.

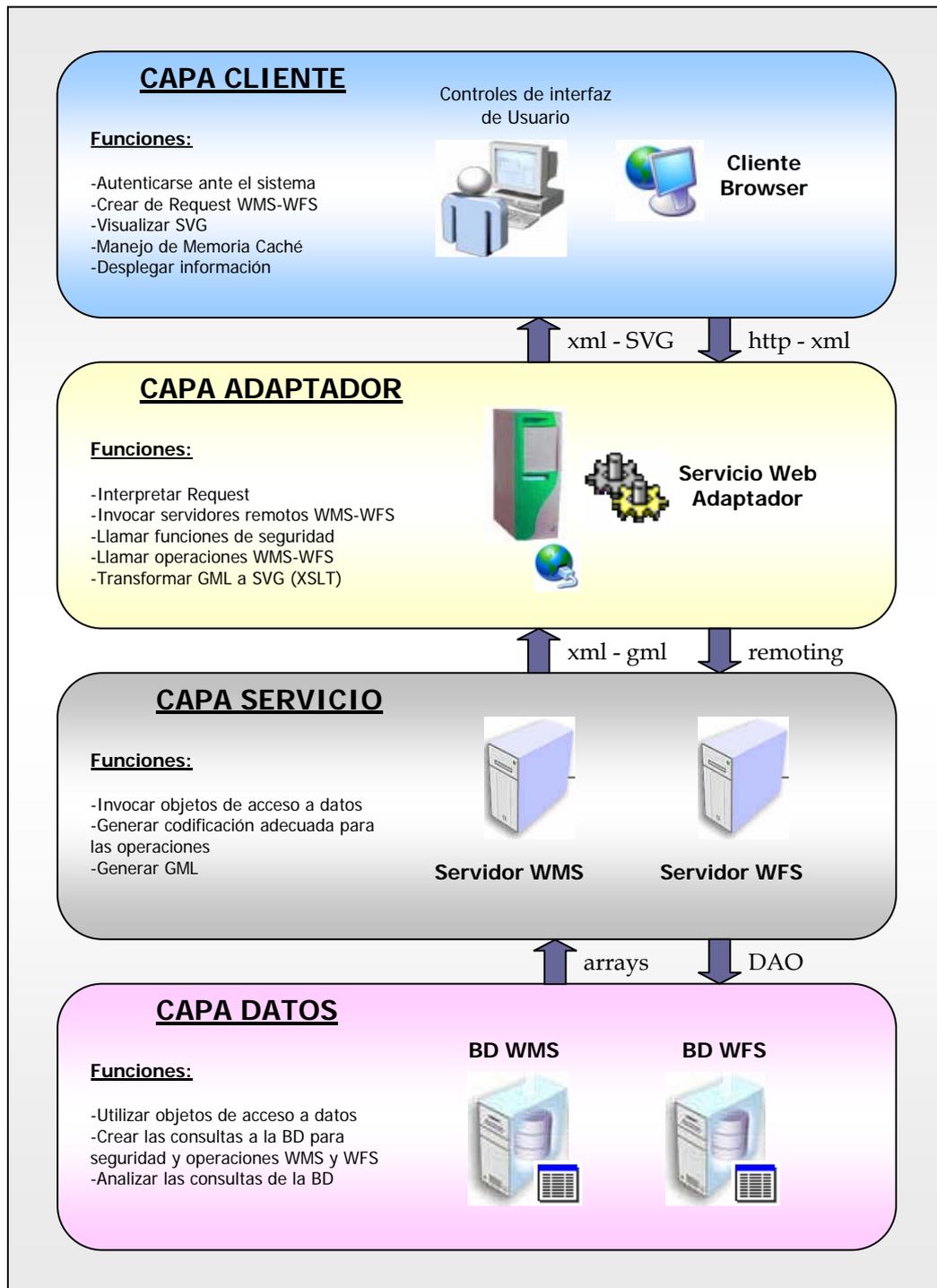
En este punto es indispensable tener en cuenta la negociación de formatos, es decir que el cliente debe solicitar los formatos que el servidor provee, y también el cliente debe poder soportar los formatos que el servidor le responda.

Por otro lado, la interfaz del usuario debe facilitar al usuario el uso correcto del software, y debe advertirle cuando éste cometa algún error ya sea de navegación o de digitación. También debe proveer la forma adecuada para el ingreso de los parámetros de las consultas de manera clara para el cliente.

En cuanto a la eficiencia en la transmisión del mapa, para el caso de la operación GetMap de WMS, es indispensable que el cliente tenga las herramientas adecuadas para poder visualizarlo e interpretarlo. Por otra parte, el usuario final debe poder descargar las consultas que éste pueda realizar con base en los permisos.

#### **3.4.5.6. Arquitectura para la generación de servicios WMS y WFS**

La arquitectura que se plantea para la implementación de los servicios en línea que soportan las operaciones básicas de los estándares WMS y WFS, se establece con base en las capas mencionadas anteriormente, y se presenta en la Figura 21.



**Figura 21. Arquitectura para la generación de Servicios en Línea WMS y WFS**

### **3.5. Fase de Implementación**

#### **3.5.1. Patrones de Implementación**

El patrón de implementación que adopta una aplicación está determinado por los encargados de la arquitectura, por lo cual éste es un factor relativo porque las organizaciones pueden abordar el problema desde diversas perspectivas, teniendo en cuenta que no existe un único enfoque cuando se va a determinar una infraestructura a una solución. Cada patrón que se aplique, tiene ciertos requisitos para su correcto funcionamiento, ya que el ámbito en el cual se da cierto patrón debe tener las condiciones dadas para que éste opere correctamente y se aproveche para lo cual fue creado.

También se debe tener en cuenta que se puede realizar la combinación de patrones de implementación. Se recomienda que se implemente cada componente de la solución únicamente en un nivel físico, aunque por razones de seguridad puede que se desee considerar la implementación del mismo componente en distintas ubicaciones por razones de facilidad de uso.

En la parte de integración de servicios, en este caso en el adaptador de servicios WMS y WFS, se utiliza un patrón que consiste en la separación de componentes empresariales de agentes e interfaces de servicios. Este modelo de infraestructura es utilizado para separar los niveles que tienen contacto con Internet (mediante la recepción o la realización de llamadas a otros servidores) desde los servidores que alojan lógica empresarial, es decir la lógica del negocio en este caso los servidores que proveen las operaciones de describen las especificaciones WMS y WFS.

Los patrones de software que se recomienda utilizar son los planteados en la fase de diseño, y son implementados como se explica en el capítulo del Prototipo.

Para ésta fase de implementación, se establece un contexto en el cual se desarrolla la funcionalidad por partes, es decir por funciones de las capas más simples, hasta las de mayor nivel, con el objetivo de lograr un producto de mayor calidad en menor tiempo, y para que los errores que se presenten en el proceso de implementación, se puedan detectar y corregir de manera sencilla y rápida.

Por otro lado, es indispensable hacer uso de información real para realizar las pruebas de funcionalidad, porque con datos ficticios es posible que no se detecten los errores debido a factores como formatos, tamaño de datos y dimensión de los datos entre otros.

### 3.5.2. Modelo de clases e interfaces para la arquitectura por capas

A continuación se describen cada una de las capas del modelo multicapa propuesto, con sus respectivos paquetes o proyectos asociados y a su vez con las clases que forman parte de cada uno de dichos paquetes.

#### 3.5.2.1. Capa de Datos

La estructura de la capa de datos, se recomienda realizarla en un sólo paquete (aunque dependiendo de la complejidad de los datos en cuanto a cantidad y análisis, se podría pensar en utilizar más paquetes), que en este caso sería llamado DAO con base en el patrón Data Access Object utilizado, en el cual se tienen las siguientes clases y se diagraman en la Figura 22.

##### ☞ AbstarctFactory:

Publica métodos para obtener objetos de acceso a datos independientes del proveedor de datos tales como conexiones, lectores de datos y comandos.

##### ☞ XXFactory:

Implementa la clase abstracta anterior como una fábrica para el proveedor de datos particular XX = OR (Oracle), SS (SqlServer), MS (MySql).

##### ☞ DAOFactory:

Clase abstracta que implementa un método que obtiene un objeto DAO.

##### ☞ IDAO:

Interface que publica métodos para obtener las siguientes interfaces: ILayer, ICapabilitiesWMS, ICapabilitiesWFS, IWMSSeguridad y IWFSSeguridad

##### ☞ DAO:

Clase que implementa la interface IDAO.

##### ☞ ICapabilitiesWMS:

Interface que publica métodos para acceder a la funcionalidad para obtener las capacidades del servidor WMS.

☞ CapabilitiesWMSImpl:

Clase que implementa la interface ICapabilitiesWMS.

☞ ICapabilitiesWFS:

Interface que publica métodos para acceder a la funcionalidad para obtener las capacidades del servidor WFS.

☞ CapabilitiesWFSImpl:

Clase que implementa la interface ICapabilitiesWFS.

☞ ILayer:

Interface que publica métodos para acceder a información detallada de cada una de las capas de los mapas.

☞ LayerImpl:

Clase que implementa la interface ILayer.

☞ IWMSSeguridad:

Interface que publica métodos para acceder a información asociada con seguridad de datos, para llevar a cabo los procesos de autenticación y autorización del servidor WMS.

☞ WMSSeguridadImpl:

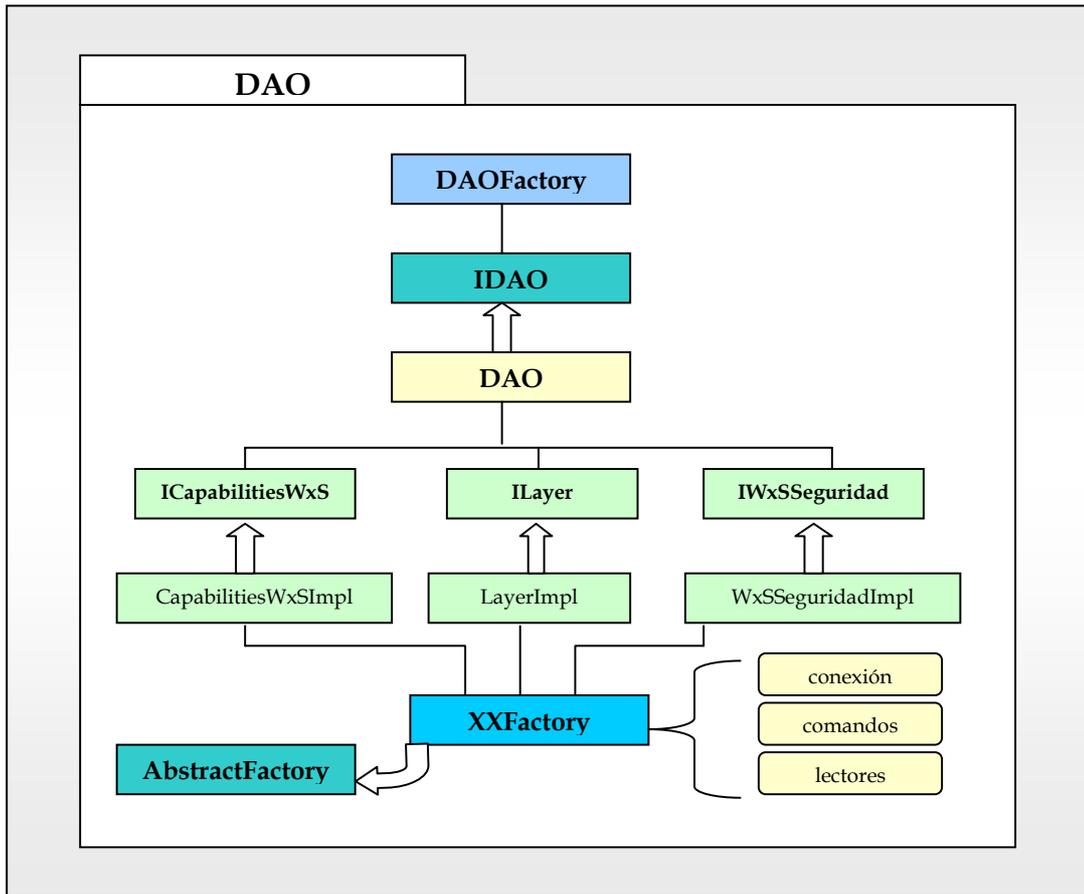
Clase que implementa la interface IWMSSeguridad.

☞ IWFSSeguridad:

Interface que publica métodos para acceder a información asociada con seguridad de datos, para llevar a cabo los procesos de autenticación y autorización del servidor WFS.

☞ WFSSeguridadImpl:

Clase que implementa la interface IWFSSeguridad.



**Figura 22. Esquema de paquetes y clases de la Capa de Datos**

### 3.5.2.2. Capa de Lógica de Servicio

La estructura de la capa de la lógica de los servicios en línea, se planea de la siguiente manera:

Un paquete para almacenar las clases que desempeñan las operaciones de cada uno de los estándares (WMS y WFS) la cual se ha denominado GML, y dos paquetes para cada servidor: uno que contiene la clase llamada WindowsServiceWxS cuya función es registrar el objeto remoto del servidor WxS, otro paquete con el objeto remoto como tal que se ha denominado ServerWxS.

Dichos paquetes se describen en seguida y se diagraman en la Figura 23.

## PAQUETE GML

### ☞ WMSOperations:

Esta clase se encarga de generar los documentos resultado de las operaciones básicas propias de WMS. Utiliza la estructura del paquete DAO, para obtener la información de la base de datos, y posteriormente genera un documento XML para la operación GetCapabilities y un documento GML para la operación GetMap.

### ☞ WFSOperations:

Esta clase es la encargada de generar los documentos resultado de las operaciones básicas propias de un WFS, para lo cual utiliza la estructura del paquete DAO, para obtener la información de la base de datos, y posteriormente genera un documento XML para la operación GetCapabilities, GetFeature y DescribeFeatureType.

## PAQUETE SERVER\_WxS

### ☞ WMS:

Objeto remoto el cual se encarga de realizar las operaciones de autenticación por medio de la estructura DAO, y de realizar las operaciones WMS por medio del objeto WMSOperations.

### ☞ WFS:

Objeto remoto encargado de realizar las operaciones de autenticación por medio de la estructura DAO, y de realizar las operaciones WFS por medio del objeto WFSOperations.

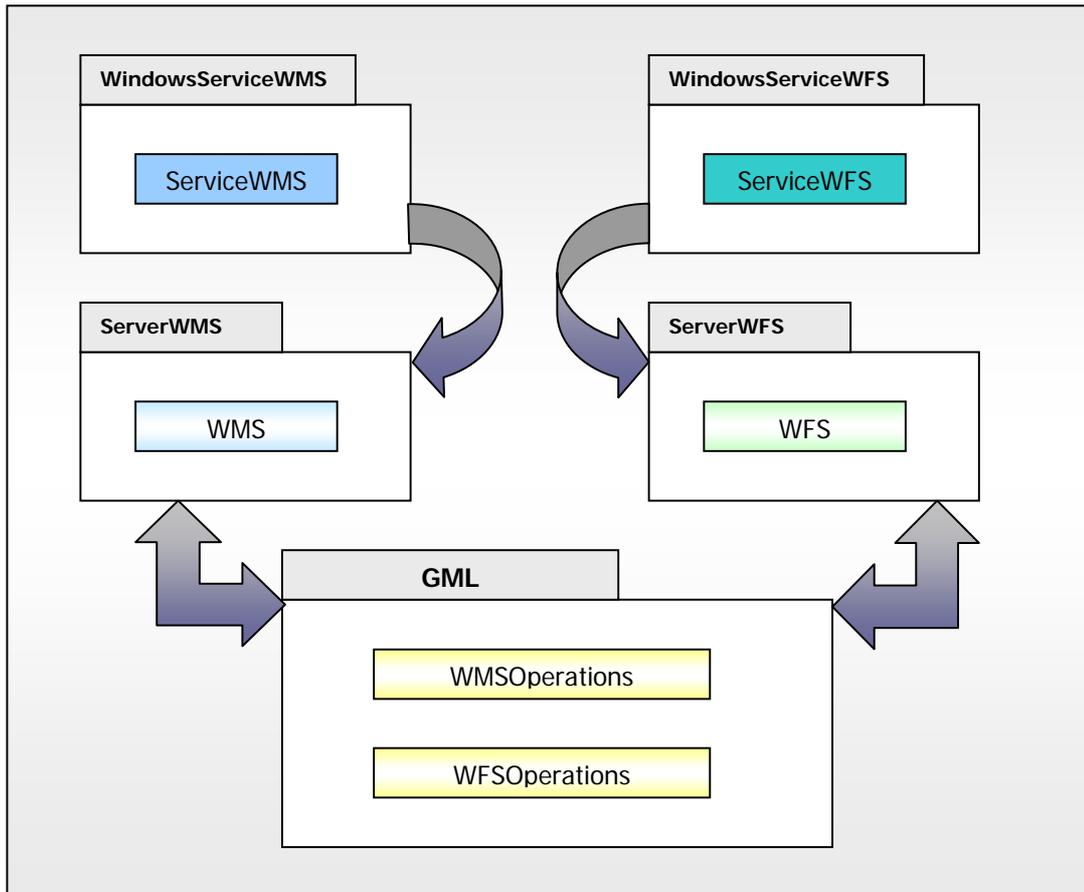
## PAQUETE WindowsServiceWxS

### ☞ ServiceWMS:

Clase que contiene una función principal que es ejecutable, la cual registra el objeto WMS remoto por medio de un canal y en un puerto asociado, para poder ser invocado después por el servicio Web.

### ☞ ServiceWFS:

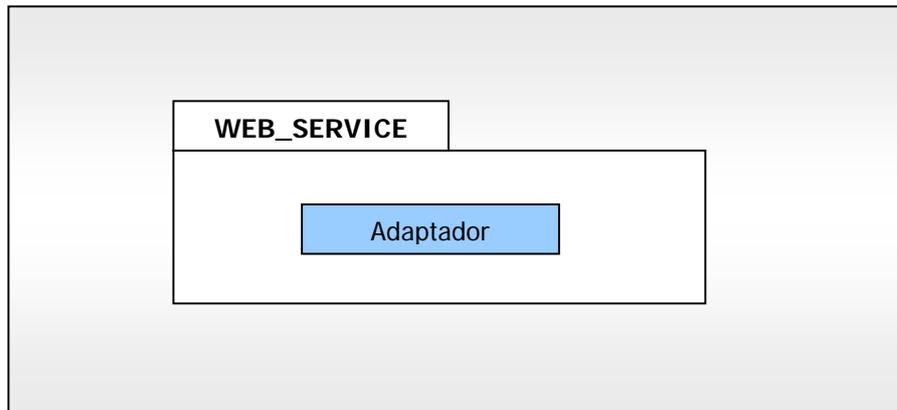
Clase que contiene una función principal que es ejecutable, la cual se encarga de registrar el objeto WFS remoto por medio de un canal y en un puerto asociado, que le permita ser invocado después por el servicio Web.



**Figura 23. Esquema de paquetes y clases de la Capa de Lógica de Servicio**

### 3.5.2.3. Capa de Adaptador de Servicio

La estructura de la capa del adaptador de servicios consiste en un paquete llamado **WebService**, que contiene un servicio Web denominado **Adaptador**, el cual publica métodos que el cliente puede acceder directamente cuando solicita las operaciones básicas de WMS y WFS. Dicho paquete se diagrama en la Figura 24.



**Figura 24. Esquema de paquetes y clases de la Capa de Adaptador de Servicios**

#### **3.5.2.4. Capa de Cliente**

La estructura de la capa de cliente, depende básicamente de la funcionalidad que se determine para cada uno de los servidores WMS y WFS. Sin embargo, se plantea almacenar toda la parte asociada a la interface del usuario, en un paquete que se ha denominado Cliente, el cual contiene clases que especifican el curso del comportamiento de un cliente que use los servicios en línea ofrecidos por los servidores WMS y WFS.

Las clases forman parte de éste paquete con clases de tipo interface de usuario ya sea Web o usuario de escritorio, para el caso se plantea el uso de interface de usuario Web, ya que los servicios que se ofrecen serán sobre Web.

Las clases que contiene el paquete Cliente son:

##### **☞ Ingreso:**

Esta es una clase que especifica el ingreso del usuario al sistema, en la cual se le da al usuario la opción de ingresar su nombre de usuario y contraseña para que el adaptador de servicios lleve a cabo el proceso de autenticación y autorización, y así el cliente se pueda adaptar a dicha respuesta del adaptador.

☞ Cliente WMS:

Esta clase representa la interface requerida para soportar los servicios de un WMS, contiene una instancia de una clase que permite visualizar el formato SVG que se devuelve al cliente como respuesta a la solicitud de algún mapa (Visor SVG). De igual forma provee al cliente la forma para visualizar y guardar en cualquier momento la información proveída por los servicios representada en documentos.

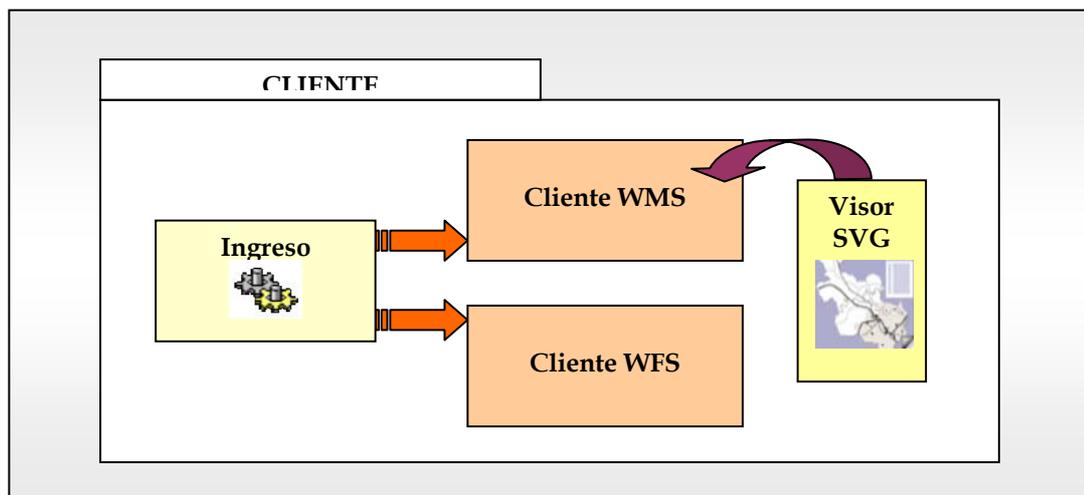
☞ Cliente WFS:

Esta clase representa la interface requerida para soportar los servicios de un WFS. Provee al cliente la forma de ingresar la información requerida para solicitar un servicio WFS con las características seleccionadas por el cliente, así como controles y componentes gráficos para visualizar y guardar en cualquier momento, la información proveída por los servicios representada en documentos.

☞ Visor SVG:

Esta es una clase que se debe diseñar para poder visualizar el formato SVG, y con ello permitir la superposición de capas de manera clara para que el cliente pueda visualizar la información solicitada. La idea es que una instancia de ésta clase se incluya o se embeba en la interface del cliente WFS quien es el que visualiza los mapas.

El paquete anteriormente descrito se diagrama en la Figura 25.



**Figura 25. Esquema de paquetes y clases de la Capa de Cliente**

### 3.5.3. Implementación de las operaciones básicas de WMS y WFS

A continuación se describe la forma en la cual se realiza cada una de las operaciones básicas de los estándares WMS y WFS. Se destacan 3 tipos de operaciones en general: Las operaciones que consultan las capacidades de los servidores, la operación que solicita un mapa y las operaciones de consulta con parámetros de entrada para filtrar la respuesta en el documento que se genera, teniendo en cuenta que se van a implementar las operaciones básicas de los servidores WMS y WFS. A continuación se describen los tipos de operaciones planteados anteriormente:

#### ☞ Obtener las capacidades de un servidor

El cliente envía al servicio Web representado por el Adaptador, un request(solicitud) adecuado y entendible por el servidor correspondiente. Cuando el adaptador recibe el request, invoca al servidor remoto e invoca al método GetCapabilities, para lo cual el servidor remoto se encarga de utilizar las clases planteadas en la capa de lógica de servicios para acceder a la base de datos, y extraer la información requerida para armar el documento XML resultado de dicha operación. El documento XML es devuelto al cliente, el cual lo visualiza como texto y lo puede guardar. El funcionamiento de ésta operación se diagrama en la Figura 26.

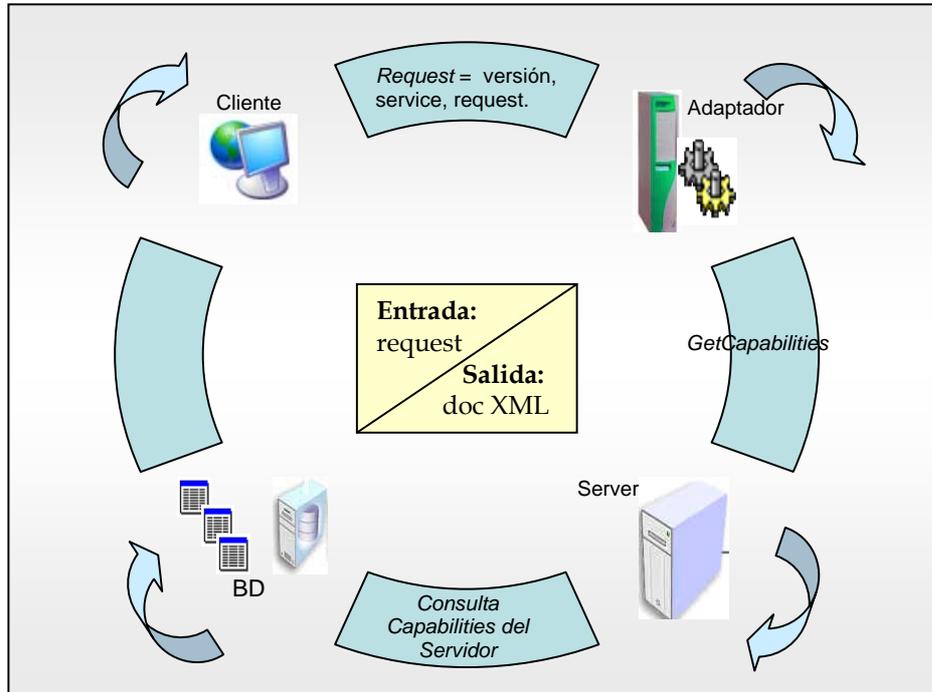
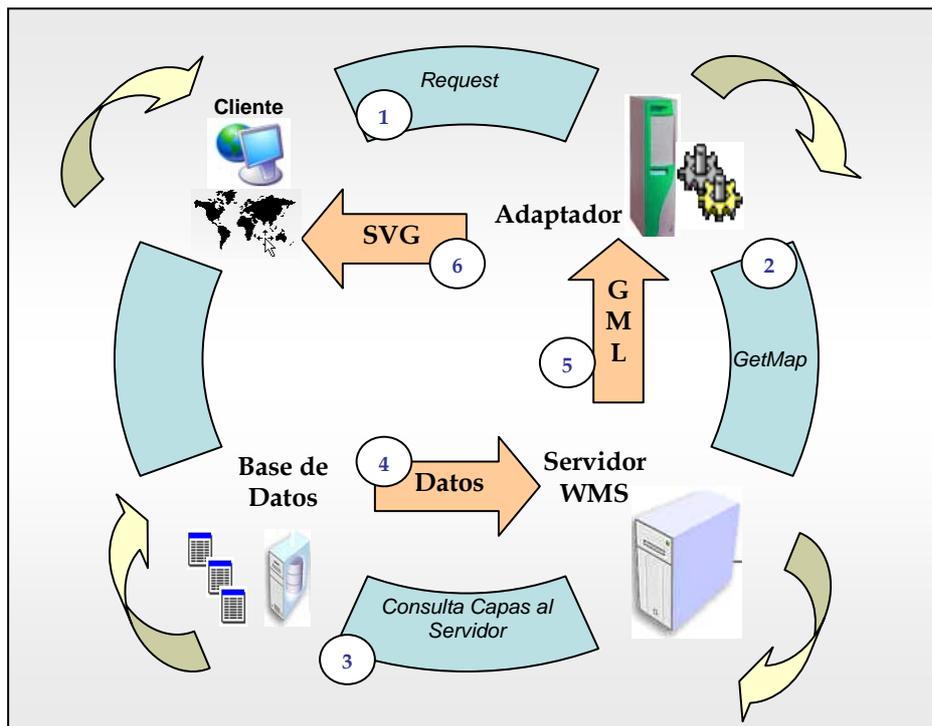


Figura 26. Esquema de funcionamiento de la operación GetCapabilities

### ☞ Obtener un mapa

El cliente envía al adaptador la solicitud de las capas que desea ver en el mapa, el adaptador entonces invoca al servidor remoto WMS, el cual utiliza las funciones de la capa del negocio para acceder a la información asociada a las capas solicitadas y devuelve dicha información al adaptador como un documento GML, y es entonces cuando el adaptador hace uso de hojas de estilos adecuadas para transformar la codificación geográfica GML en formato gráfico visible para el cliente como lo es SVG. Se determinó que el formato más adecuado para devolver mapas al cliente es SVG debido a las características que se mencionan en la Tabla 18.

Además el servidor tiene la función de guardar en memoria caché, las capas que el cliente ya haya solicitado anteriormente, con el propósito de ahorrar tiempo en el procesamiento de los request siguientes, y con ello conseguir eficiencia, porque no se tiene que seguir el mismo proceso de recorrido por todas las capas, para la obtención del mapa que es devuelto al cliente. El esquema de funcionamiento de la operación se presenta en la Figura 27.

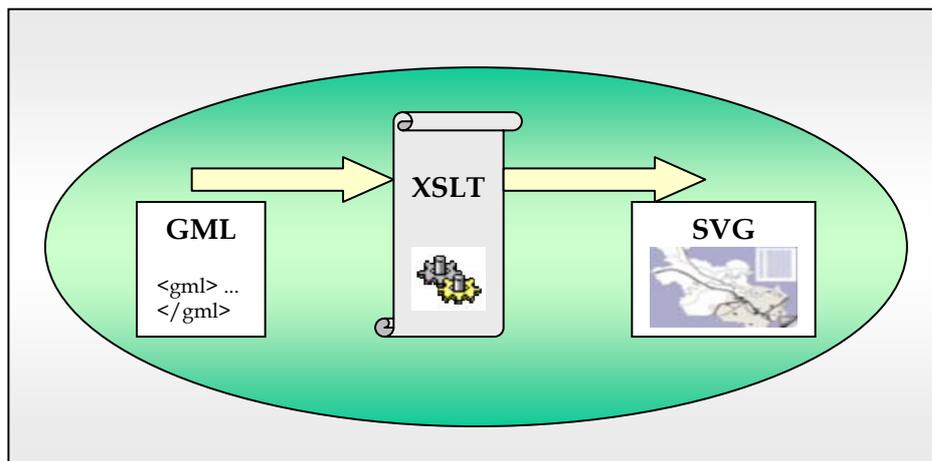


**Figura 27. Esquema de funcionamiento de la operación GetMap**

Uno de los elementos esenciales de la operación GetMap, es ofrecer al cliente un formato de imagen que sea liviano y al mismo tiempo con características que se adapten a sus necesidades tales como modificaciones del mapa, capacidades de zoom sin distorsión y otras como las mencionadas en la Tabla 18.

Para la obtención del formato SVG se sigue el proceso esquematizado en la Figura 27, en la cual se genera el GML a partir de datos de una base de datos geográfica, y posteriormente se transforma a formato SVG entendible por el cliente.

En la Figura 28 se ve representada la forma como una hoja de estilos toma la codificación GML y la transforma en SVG por medio de equivalencias entre los tags de GML con los tags de SVG, agregándole propiedades de dibujo para poder pintar cada uno de los polígonos que representa un rasgo de una capa determinada.



**Figura 28. Esquema de Transformación de GML a SVG**

☞ Obtener un documento filtrando la salida con parámetros de entrada.

Este tipo de operaciones se comportan de igual forma que las operaciones de obtener capacidades de los servidores, con la diferencia de que los parámetros de entrada para las consultas, son utilizados para filtrar los resultados de los documentos XML dados como resultado al cliente.

En éste tipo de operaciones se incluyen las operaciones básicas de WFS DescribeFeatureType y GetFeature.

### 3.5.4. Diseño de Interfaces de usuario

Para el diseño de las interfaces del usuario Web, se plantean las siguientes recomendaciones:

- ⇒ Implementar una página de error personalizada y un controlador de excepciones global, para tener una función completa de detección de excepciones que evitará que el usuario vea páginas no descriptivas en caso de que ocurra algún problema.
- ⇒ Implementar o usar controles de validación de entradas del usuario, para evitar las excepciones o las respuestas erróneas por parte del sistema. El uso de controles de validación permite garantizar que los datos escritos por el usuario se ajusten a determinados criterios, lo que también ahorra esfuerzo en el lado del servidor.
- ⇒ Cuando se crean controles de usuario Web, se deben exponer únicamente las propiedades y los métodos públicos que se necesiten realmente, lo que facilita las labores de mantenimiento.
- ⇒ Las funciones de control deben invocar a las acciones del componente de proceso de usuario para guiar al usuario a través de la tarea actual, en lugar de redireccionarlo a la página directamente.

### 3.5.5. Despliegue de Información al Cliente

Con una aplicación Web que genere servicios en línea geográficos, se pueden desplegar dos tipos de datos, basados en las respuestas a los estándares WMS y WFS:

- ⇒ Datos o atributos gráficos.
- ⇒ Datos o atributos no gráficos.

En el despliegue de los datos se deben proveer las siguientes funciones:

- ⇒ Localizar e identificar elementos geográficos

Con un mapa se puede determinar lo que existe en un sitio particular, para lo cual se deben especificar las condiciones. Esto se hace especificando la localización de un objeto o región para la cual se desea información.

Los métodos comúnmente usados son:

- Señalar con el apuntador gráfico o mouse el objeto o región.
- Escribir en el teclado la dirección.
- Escribir en el teclado las coordenadas.

Luego de dadas las condiciones para localizar un objeto o región se obtienen ciertas salidas, las cuales pueden presentar algunas de las características del objeto seleccionado.

⇒ Especificar condiciones

Esta función determina el lugar donde se satisfacen ciertas condiciones. La especificación de dichas condiciones se puede hacer por medio de:

- La selección desde unas opciones predefinidas.
- La escritura de expresiones lógicas.
- El diligenciamiento interactivo en la pantalla.

Luego de dadas las condiciones que como usuario requiere, se obtiene la respuesta esperada. En cada respuesta se puede presentar:

- Un listado de todos los objetos que reúnen la condición.
- Los elementos que cumplen la condición resaltada gráficamente.

⇒ Hacer análisis espaciales

En esta función los datos se pueden analizar para obtener:

- Respuestas a preguntas particulares.
- Soluciones a problemas particulares.

Los análisis geográficos se pueden hacer por medio de superposición de las características de los elementos de una misma categoría.

El formato seleccionado para devolver mapas al cliente fue SVG debido a los siguientes factores:

- GML puede entregar datos vector sobre Internet por nombramiento de los datos en formato Scalable Vector Graphics (SVG).
- La mayoría de los programas SIG en Internet entregan datos espaciales a través de la transmisión de imágenes raster tales como formatos GIF y JPEG sobre World Wide Web, lo cual implica mas tiempo en la transmisión de la imagen, y es por ello que el formato SVG tiene las siguientes ventajas sobre formatos raster:

Las ventajas que tiene usar el formato SVG se describen en la Tabla 18.

<b>VENTAJA</b>	<b>DESCRIPCION</b>
<i>COMPATIBILIDAD</i>	SVG usa texto basado en formato XML, el cual es compatible con otros formatos. Este puede ser integrado con tecnologías Web actuales tales como HTML, JavaScript, ASP, JPEG, GIF, JSP, , etc.
<i>CALIDAD DEL GRÁFICO</i>	<p>El formato gráfico SVG es escalable e independiente de la resolución, lo cual permite que esta clase de datos pueda ser escalada sin dejar la calidad a través de las diferentes plataformas y recursos. Por su parte las imágenes raster tienen una menor calidad cuando tienen una menor resolución, en particular cuando los usuarios hacen zoom muchas veces, las imágenes se vuelven poco claras y pixeladas. Una imagen raster con alta resolución, usualmente tiene un archivo de gran tamaño ya que necesita almacenar información de píxeles más finos.</p> <p>La velocidad en la entrega de archivos pesados sobre la Web comienza a bajar, y esto no es práctico cuando se usan imágenes de alta resolución como son los mapas en un SIG. Al respecto, Bertolotto y Egenhofer en el año 2001 plantean que: “La necesidad de entregar mapas gráficos vector de alta calidad sobre Internet surge por la urgencia de disponer de los datos y el incremento de la repartición global de los mismos”.</p>
<i>FORMAS DE ACCESO</i>	Los datos vector SVG pueden ser accedidos de una manera mas interactiva y dinámica. En documentos SVG pueden ser integradas algunas funciones dinámicas tales como animaciones en la Web, por ejemplo un gráfico SVG puede interactuar con los usuarios por pasar el mouse sobre el gráfico añadiendo una función mouserover() en el documento SVG. Por combinación de SVG con otras tecnologías tales como HTML, JavaScript, JSP o ASP, base de datos basada en GML, se puede proveer a los usuarios una interface rica en gráficos interactivos.

**Tabla 18. Ventajas del formato SVG**

### **3.6. Fase de Soporte y Mantenimiento**

La primera fase del mantenimiento de un sistema software consiste en la implantación de la aplicación en el entorno donde se va a desarrollar. Posteriormente, es necesario realizar una serie de pruebas para comprobar que no han surgido problemas, en tal caso será recomendable realizar una depuración. Finalmente, se debe mantener un control del rendimiento.

#### **3.6.1. Pruebas al sistema**

En la fase de validación y prueba del sistema con los requisitos especificados por los usuarios, se diseña un conjunto de pruebas que hagan uso de datos reales, que se deben llevar a cabo de manera metódica y

rigurosa. Es importante darse cuenta de que la fase de prueba no sirve para demostrar que no hay fallos, sirve para encontrarlos. Si la fase de prueba se lleva a cabo correctamente, descubrirá los errores en los programas de aplicación y en la estructura de la base de datos, y por otro lado demostrará que los programas parecen trabajar tal y como se especificaba en los requisitos y que las prestaciones deseadas parecen obtenerse. Por último, en las pruebas se podrá hacer una medida de la fiabilidad y la calidad del software desarrollado, con base en la especificación de los requerimientos del usuario.

En el proceso de prueba del software es significativo contar con documentos o plantillas que ayuden a conocer el funcionamiento real de la aplicación y por medio de ellos se puedan detectar y corregir los fallos que ocurran en la aplicación.

La guía propone la siguiente plantilla de pruebas y es la que se seguirá cuando se evalúe el prototipo, cabe notar que esta plantilla será diligenciada con cada script de pruebas que se desee evaluar.

<b>Identificación de la prueba</b>		CP01 - 1	
<b>Proyecto:</b>	Servicios en línea a partir de los estándares WFS y WMS basados en visualización con tráfico liviano y manejo de seguridad		
<b>Caso de Uso:</b>			
<b>Definido por:</b>		<b>Fecha:</b>	dd/mm/aaaa
<b>Realizado por:</b>		<b>Fecha:</b>	dd/mm/aaaa
<b>Estado:</b>			
<b>Descripción breve de la prueba</b>			
Esta prueba se hace con el objetivo de .....			
<b>Secuencia del Script</b>	<b>Resultados Esperados</b>	<b>Errores</b>	<b>Gravedad</b>
1.			
2.			
3.			
<b>Observaciones:</b>			

### **3.6.2. Despliegue e instalación del sistema**

Cuando el sistema ya ha sido probado con datos reales y corregido, es indispensable tener las condiciones adecuadas para su correcta instalación en los equipos correspondientes, teniendo en cuenta elementos de configuración que se requieren para la correcta comunicación entre el sistema, porque se trata de un sistema distribuido.

Los equipos para instalar el sistema deben estar adecuados con el hardware requerido en los requerimientos no funcionales, y de igual forma debe estar disponible el personal humano para el proceso de instalación del software y para la realización de las pruebas del sistema ya instalado en su totalidad. Los archivos de configuración deben estar adecuadamente diligenciados, dependiendo de la infraestructura de red que maneje la organización que ha seguido ésta guía metodológica.

### **3.6.3. Mantenimiento del Sistema**

Una vez que el sistema está completamente implementado y probado, se pone en marcha. El sistema está ahora en la fase de mantenimiento en la que se llevan a cabo las siguientes tareas:

- Monitorización de la funcionalidad del sistema:

Si la funcionalidad cae por debajo de un determinado nivel, puede ser necesario reorganizar la base de datos.

- Mantenimiento y actualización del sistema.

Cuando sea necesario, los nuevos requisitos que vayan surgiendo se incorporarán al sistema, siguiendo de nuevo las etapas del ciclo de vida que se acaban de presentar.

### **3.7. Resultados esperados**

Con el seguimiento adecuado de los pasos planteados en la guía metodológica en cada una de las fases propuestas: Planeación, Análisis, Diseño, Implementación y Soporte, se espera como resultado, un sistema de información que permita la generación de servicios en línea geográficos de manera eficiente y segura, con base en los estándares del OpenGIS Consortium: WMS, WFS y GML.

## 4. PROTOTIPO

### 4.1. Introducción

El prototipo es la forma en la cual se comprueba la efectividad de la guía metodológica planteada, y es por ello que se decidió realizarlo utilizando datos reales de una entidad de Bogotá generadora de información geográfica. Teniendo en cuenta que son muchas las entidades en Bogotá que generan Información geográfica, se llevó a cabo un proceso de selección de la entidad; posteriormente se realizó un análisis de los datos para la implementación del prototipo, y al final se realizó una comparación en cuanto a eficiencia de los servicios en línea que genera la empresa, contra los servicios básicos generados a través de la guía metodológica.

### 4.2. Caso de Estudio

Pasos que se tomaron en cuenta para llevar a cabo el análisis de entidades de Bogotá que producen información geográfica

- ✓ Selección de las principales entidades de Bogotá que manejan información geográfica, para aplicarles el cuestionario diseñado para ello (Ver Anexo F). En este punto se determinó una muestra que fuera representativa, para lo cual se seleccionaron 7 entidades cuya misión es proveer los principales servicios públicos y geográficos actualmente a la ciudad de Bogotá.

#### **ENTIDADES SELECCIONADAS**

- *Instituto Colombiano Agropecuario (ICA)*
- *Empresa de Acueducto y Alcantarillado de Bogotá (EEAB)*
- *Instituto Geográfico Agustín Codazzi (IGAC)*
- *Empresa de Teléfonos de Bogotá (ETB)*
- *Departamento Administrativo Catastro (DACD)*
- *Gas Natural*
- *Codensa*

- ✓ Búsqueda de contactos adecuados dentro de cada entidad.
- ✓ Realización del cuestionario (Fecha de realización: Febrero-Marzo de 2004).
- ✓ Desarrollo de un cuadro de análisis de las entidades (Ver Anexo G), con base en los siguientes criterios:
  1. Si genera servicios en línea.
  2. Servicios en línea que genera.
  3. Manejo de interoperabilidad.
  4. Fuentes de información para la generación de servicios en línea.
  5. Encargados de generar los servicios en línea.
  6. Proceso de generación de los servicios en línea.
  7. Herramientas y estándares que utilizan para generar los servicios en línea.
  8. Proyecto(s) actual(es) asociado(s) a la generación de servicios en línea.
  9. Probabilidad de generar servicios en línea.
  10. Entidades con las cuales le gustaría compartir información.
  11. Apoyo a éste proyecto por parte de la entidad en cuanto a información que se requiera.

#### **4.2.1. Análisis de entidades generadoras de Información Geográfica de Bogotá**

Con base en el cuadro de análisis desarrollado (Ver Anexo G), se obtuvieron las siguientes conclusiones:

- ✓ Es conveniente seleccionar una entidad que genere servicios en línea basados en información geográfica, así sean mínimos, porque se cuenta con los recursos físicos y de información, para llevar a cabo el proceso de generación de los nuevos servicios para probar la guía metodológica que se generará con éste proyecto de investigación.
- ✓ La entidad a seleccionar debe estar dispuesta a apoyar el proyecto de investigación, en el sentido que debe adquirir un compromiso con el equipo de investigación, para proveer la información que éste mismo requiera y de ésta manera llevar a cabo las pruebas de la guía metodológica.
- ✓ La entidad a seleccionar debe tener claridad en los nuevos servicios en línea que podría generar con el apoyo del equipo investigador.
- ✓ La entidad seleccionada debe tener las herramientas necesarias para la generación de los servicios en línea.
- ✓ La entidad debe tener los recursos físicos que soporten una arquitectura que se plantearía con la guía metodológica.

#### 4.2.2. Selección de Caso de Estudio

La entidad seleccionada con base en las conclusiones planteadas en el análisis de entidades generadoras de información geográfica en la ciudad de Bogotá, fue la Empresa de Acueducto y Alcantarillado de Bogotá (EAAB).

Esta entidad maneja información geográfica de la ciudad de Bogotá con el objetivo de mantener un control de la red de acueducto y alcantarillado. Dicha información geográfica (distribución geográfica de la red de acueducto y alcantarillado de Bogotá) se encuentra sistematizada a través de un Sistema de Información Geográfica interno para la empresa. Para el manejo de la información a nivel externo, tienen una página Web en la cual los usuarios pueden acceder a mapas interactivos y realizar ciertas consultas, como solicitud de capas.

Para llevar a cabo el proceso de formalización con la entidad, y así mismo adquirir el compromiso de facilitar la información requerida, se realizaron las siguientes actividades:

- Entrega de una carta de presentación a la persona encargada del departamento de Sistemas de Información Geográfica.
- Entrevista inicial con las personas que colaborarán en el proyecto, el director del proyecto y el grupo investigador.
- Entrevistas y visitas para adquirir la información relevante para el trabajo. inicial con las personas que colaborarán en el proyecto, el director del proyecto y el grupo investigador.

#### 4.3. Desarrollo de la Guía Metodológica en la entidad seleccionada

A continuación se describe el seguimiento de la Guía Metodológica planteada anteriormente, en la entidad seleccionada para el caso de estudio: **La Empresa de Acueducto y Alcantarillado de Bogotá (EAAB)**.

Teniendo en cuenta que la guía plantea una serie de actividades por fases (Planeación, Análisis, Diseño, Implementación y Soporte), en ésta parte se describe la forma detallada como se siguió el procedimiento para la ejecución de cada una de las actividades, aplicando las sugerencias, modelos y demás factores que propone la guía metodológica.

### **4.3.1. Fase de Planeación**

#### **4.3.1.1. Identificación de la necesidad de generar servicios en línea**

La EAAB como entidad pública que busca compartir información con otras entidades, manifestó interés en los estándares internacionales propuestos por la guía metodológica (WMS, WFS y GML del OpenGIS Consortium), para el manejo de la información geográfica, por lo cual presenta la disposición para analizar los beneficios de ésta estandarización.

La necesidad para generar servicios en línea, ya había surgido anteriormente, por lo cual cuentan con un software en la Web que provee servicios de visualización y consulta de mapas de la ciudad de Bogotá.

Para el caso de la aplicación de ésta guía metodológica la necesidad que surge, es optimizar estos servicios por medio del uso de estándares internacionales para el manejo de información geográfica, con el objetivo de lograr la interoperabilidad entre los sistemas por medio de los Web Services.

Un beneficio claro que se percibe con la implantación de los estándares geográficos para la generación de los servicios en línea, es que la entidad puede ampliar su oferta de servicios especializados y modulares a clientes y empresas, de igual forma puede conseguir mejores conexiones con clientes y proveedores para optimizar las actividades de la cadena de valor y, al mismo tiempo, colaborar en mejorar los procesos internos, como el desarrollo de productos con información actualizada.

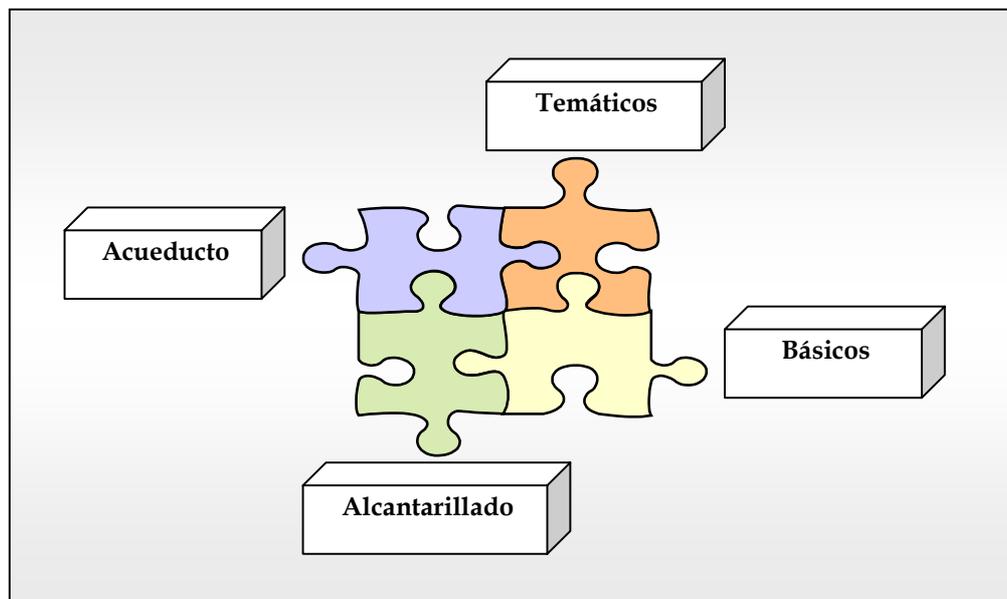
#### **4.3.1.2. Factores a tener en cuenta cuando se planea construir un servicio en línea**

A continuación se describen para el caso de estudio, los requisitos que se deben tener en cuenta cuando se planea crear servicios en línea:

- Tener información geográfica:

La entidad evidentemente tiene información geográfica de la ciudad de Bogotá, para lo cual tienen en funcionamiento una base de datos geográfica en la que manejan los datos de cada una de las capas de datos que ofrecen en los mapas interactivos en línea.

La EAAB maneja un modelo de datos dividido en cuatro áreas representadas en la Figura 29.



**Figura 29. Modelo de datos geográficos de la EAAB**

Como se observa en la Figura 29, la entidad maneja 4 grupos de datos, en los cuales estructura por áreas la información que posee. Para el caso de estudio se tomaron datos de prueba del grupo de datos de capas temáticas, ya que es la más general, y además por cuestiones de confidencialidad la entidad no proporcionó las otras capas especializadas.

Como resultado de éste análisis se determina que se cumple con éste requisito de poseer información geográfica.

- Tener experiencia en UML:

UML como lenguaje de modelado unificado, actualmente es aplicado en la EAAB, lo cual se observa en los modelos en UML de la aplicación SIGUE(Ver Anexo P) y por dicha razón se determina que se cumple con el requisito de tener alguna experiencia en UML, el cual es aplicado como una notación general y simple, que se emplea en métodos de desarrollo de software.

UML en la aplicación de la guía metodológica de utilizará para el modelamiento de casos de uso.

▪ Tener infraestructura física y humana:

La infraestructura física existe actualmente teniendo en cuenta que la entidad genera servicios en línea. En cuanto al recurso humano, se tiene que las áreas involucradas en la generación de los servicios tales como:

- ⇒ Dirección SIG
- ⇒ Topografía y Geodesia
- ⇒ Sistema de Información Documental

La entidad apoyaría la generación de los servicios en el sentido de proveer la información necesaria para el desarrollo, pero en este caso el recurso humano utilizado para el desarrollo de los servicios es el equipo investigador, pero si se pensara en implantar los servicios a nivel global de la entidad, se requeriría un equipo para el desarrollo de los servicios basado en la guía metodológica propuesta.

#### **4.3.2. Fase de Análisis**

##### **4.3.2.1. Análisis de la situación actual**

A continuación se describen los factores sugeridos por la guía metodológica para el análisis de la situación actual:

##### **Infraestructura de Datos Geográficos**

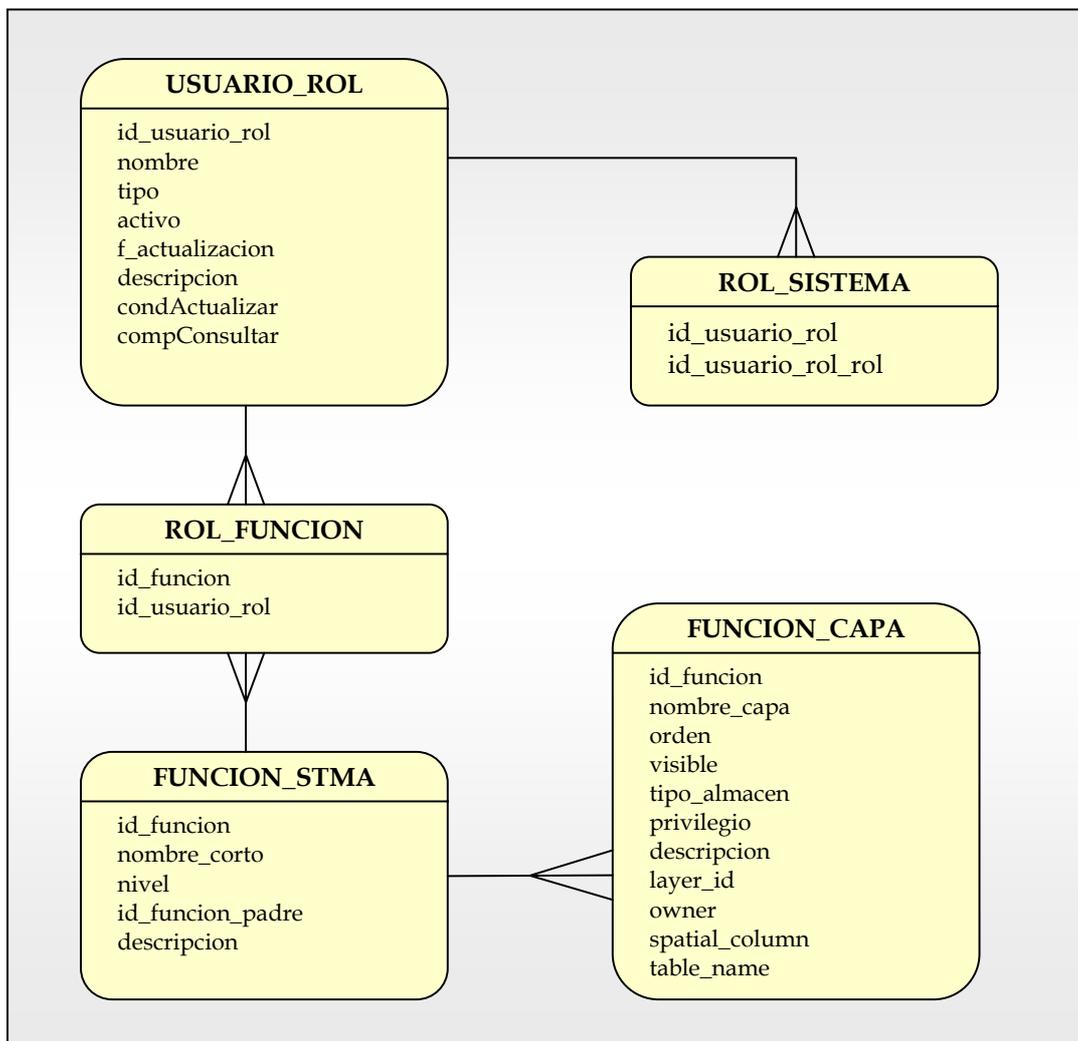
- ⇒ La entidad posee información de mapas de la ciudad de Bogotá por capas, con los metadatos asociados a cada capa.
- ⇒ La forma en la cual se encuentra estructurada la información es un modelo de datos relacional.
- ⇒ La forma de almacenamiento es en bases de datos, las cuales se encuentran en servidores ubicados en la entidad. Los formatos de la información son definidos por estándares creados por la misma entidad.
- ⇒ El volumen de información geográfica que se tiene es bastante en el sentido, que existen diferentes niveles de información para cada una de las capas, cuyo acceso es controlado por usuarios, roles y privilegios cuyo modelo de seguridad se presenta en la Figura 30.
- ⇒ El procedimiento que se lleva a cabo para la gestión de la información se divide en dos procesos básicos:
  - Gestión para obtención de datos
  - Gestión para uso de datos

Existen áreas prestadoras de servicio las cuales manejan un catálogo básico de datos.

Para la gestión de los datos se realizan las siguientes actividades:

- Seguimiento a los constructores.
- Monitoreo del terreno.
- Intercambio con entidades del distrito, para manejo de parte administrativa.
- Uso de ArcIMS el cual publica los datos cuando son ingresados a la Base de Datos, esto es manejado por un Servidor Web el cual es monitoreado por un Web Master. Cuando hay errores este último intenta resolverlo o si no recurren a la empresa que les da soporte.
- Dentro del área existe un DBA (Data Base Administrator) el cual se encarga de revisar el sistema, los TableSpaces, los permisos, los usuarios, roles y demás aspectos de la Base de Datos.

La infraestructura del sistema actual SIGUE inicialmente tuvo problemas en pruebas piloto, pero actualmente es una infraestructura estable.



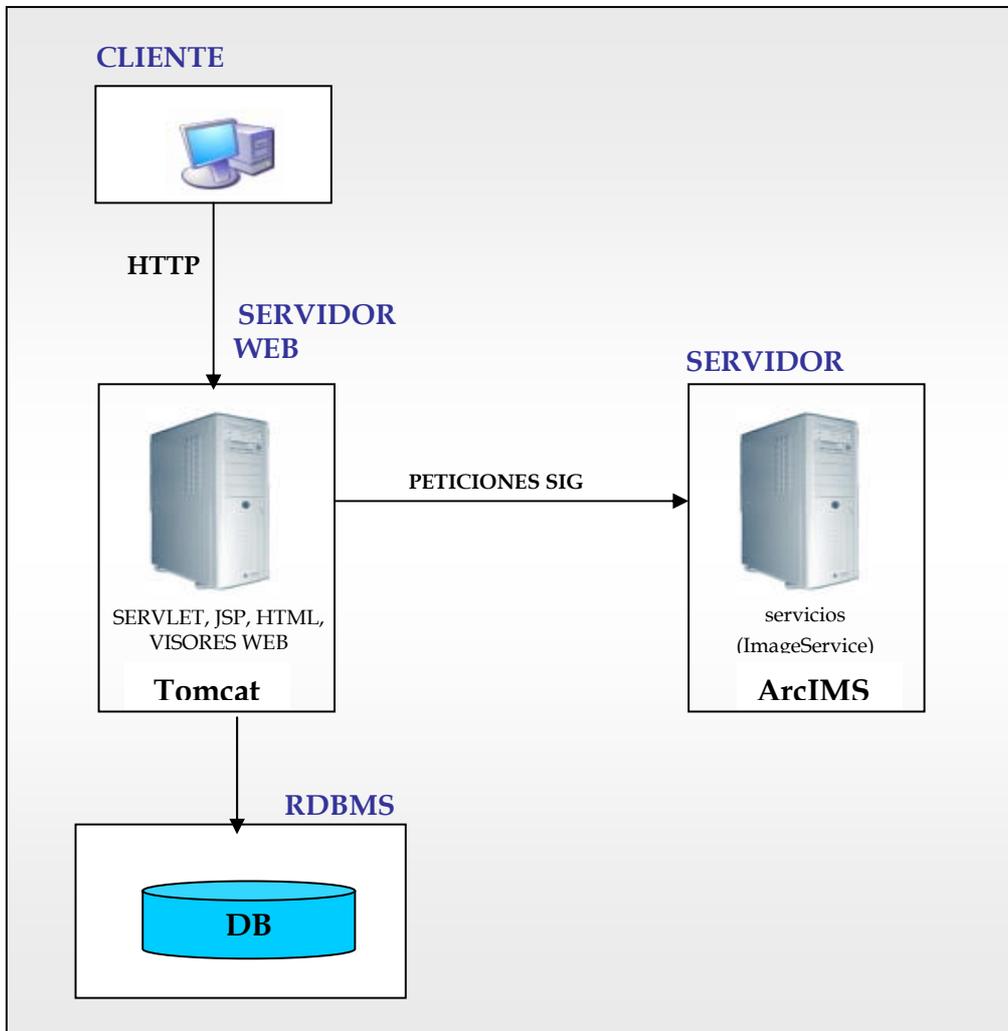
**Figura 30. Arquitectura de Seguridad de Servicios en línea EAAB**

**Infraestructura Física y Técnica**

⇒ La Empresa de Acueducto y Alcantarillado de Bogotá (EAAB), cuenta con una página en Internet que provee diversa información sobre la empresa, para el caso de estudio correspondiente a esta investigación, se tendrá en cuenta el “SISTEMA DE INFORMACION GEOGRAFICA UNIFICADO EMPRESARIAL “SIGUE” que actualmente cuenta con módulos para Internet, Intranet y Extranet.

La arquitectura de este sistema se muestra en la Figura 31.

- ⇒ La EAAB por medio de SIGUE, actualmente genera servicios en línea geográficos por medio de mapas interactivos. Teniendo en cuenta que la entidad es generadora de información geográfica, se muestra interesada en optimizar dichos servicios, y de ésta manera poder generar otros tipos de servicios que originen beneficios tanto para la entidad como para sus clientes.
- ⇒ El número de servidores y de PC's que se encuentran asociados al manejo y flujo de información geográfica es una información confidencial. Sin embargo, las áreas que manipulan este tipo de información tales como Dirección SIG, Topografía y Geodesia y el Sistema de Información Documental, son áreas que se verían involucradas si se decide implantar un sistema global, con el uso de esta guía metodológica, para lo cual se haría necesario un inventario de servidores y equipos disponibles para el desarrollo e implementación del sistema.
- ⇒ El análisis de la red interna es indispensable, porque el sistema es de tipo distribuido, para lo cual es importante tener la estructura de red de los equipos involucrados si se quisiera implantar el sistema.
- ⇒ Las instalaciones de la entidad por lo que se observó, son las óptimas para que el recurso humano pueda trabajar con un control adecuado de los riesgos profesionales.



**Figura 31. Arquitectura Servicios en línea EAAB**

### **Recurso Humano**

- ⇒ El recurso humano que se requiere debe ser capacitado en las herramientas en las cuales se decida implantar el sistema, para lo cual la entidad determinaría si se cuenta con personal propio de la entidad o si se contrataría a una empresa especializada.

- ⇒ Por otra parte, actualmente existe un DBA (Data Base Administrator) el cual se encarga de revisar el sistema, los TableSpaces, los permisos, los usuarios, roles y demás aspectos de la Base de Datos.

#### **4.3.2.2. Levantamiento de requerimientos**

En primer lugar, se ha de especificar el comportamiento del sistema desde el punto de vista del usuario, y luego del sistema global.

Los requerimientos fueron levantados con base en los niveles de requerimiento sugeridos por la guía metodológica tales como Requerimientos del negocio, del usuario, funcionales y no funcionales.

Para la captura de cada uno de los niveles de requerimiento (Ver numeral 3.3.2) se utilizaron los formatos propuestos en los anexos H, I, J y K respectivamente.

- ⇒ Los requerimientos del negocio se determinaron con base en el nivel alto de los objetivos de la organización, con respecto a los beneficios que se obtendrían, y se describen en el Anexo. L
- ⇒ Los requerimientos del usuario fueron determinados tal y como los plantea la guía metodológica basados en las operaciones básicas de los estándares WMS y WFS utilizando el formato descrito en el Anexo I.
- ⇒ Los requerimientos funcionales se determinaron con respecto a la funcionalidad de cada uno de los servidores WMS y WFS, y se describen en el Anexo M.
- ⇒ Los requerimientos no funcionales fueron analizados con respecto al alcance de la guía metodológica en la generación de servicios en línea geográficos, y se describen en el Anexo N.

#### **4.3.3. Fase de Diseño**

##### **4.3.3.1. Recomendaciones de diseño de servicios en línea**

A continuación se describe la forma como se siguieron las recomendaciones de diseño planteadas por la guía metodológica:

- ⇒ Los tipos de componentes que se han identificado para utilizar en la aplicación, se determinan con respecto a funcionalidad específica que se debe encapsular para desacoplar funcionalidad. Dichos componentes son: Componentes de acceso a datos, componentes de acceso a la información de cada uno de los servidores WMS y WFS, componentes de servicio y componentes de lógica de los servicios.
- ⇒ Los componentes fueron agrupados con base en el mismo objetivo, para facilitar el mantenimiento del diseño y la implementación.
- ⇒ La forma de comunicación entre cada uno de los componentes se estableció desde el comienzo, utilizando estándares de Internet y de manejo de información geográfica tales como XML y GML. De igual forma, los patrones de software que se utilizaron son los sugeridos por la guía en cada una de las funciones asociadas con cada uno.
- ⇒ El formato de intercambio de datos fue claramente definido cuando se especificó la arquitectura multicapa, basados en los estándares WMS, WFS y GML, los cuales describen cómo se deben estructurar los Request y los Response en cada una de sus operaciones.

#### 4.3.3.2. Fuentes de Datos

La forma en la cual se han organizado las capas de datos en grupos lógicos se describe en la Tabla 19, teniendo en cuenta los niveles de información que soportan los modelos de Acueducto, Alcantarillado y Datos Básicos y Temáticos que se presentan en la Figura 32<sup>21</sup>.

Forma	Modelos EAAB considerados
<i>Reusable</i>	Red de Acueducto y Red de Alcantarillado.
<i>Dominio específico</i>	Red de Acueducto y Red de Alcantarillado.
<i>Aplicación específica</i>	Red de Acueducto y Red de Alcantarillado.
<i>Frecuente cambio</i>	Red de Acueducto, Red de Alcantarillado y Básicos Acueducto.
<i>Detalle</i>	Red de Acueducto, Red de Alcantarillado, Básicos Acueducto y Temáticos Acueducto.

**Tabla 19. Grupos lógicos de datos para la EAAB**

<sup>21</sup> <http://www.acueducto.com.co/siguweb/index.htm> Fecha de Consulta: Mayo de 2004.

RED ALCANTARILLADO	LINEA ALCANTARILLADO TUBO	TEMATICOS ACUEDUCTO	PROYECTOS ACUEDUCTO EAAB
	LINEA ALCANTARILLADO CULVERT		PROYECTOS ALCANTARILLADO EAAB
	LINEA ALCANTARILLADO OVOIDE		PROYECTOS ESPECIAL EAAB
	LINEA ALCANTARILLADO CANAL		CICLO FACTURACION
	LINEA ALCANTARILLADO HERRADURA		ZONA EAAB
	LINEA ALCANTARILLADO ALIVIO		ZONAS DE MANEJO AMBIENTAL
	LINEA ALCANTARILLADO SIFON INVERTIDO		RONDA DE RIO
	LINEA ALCANTARILLADO VALLADOS		CLIENTES EAAB
	LINEA ALCANTARILLADO TUBO SUMIDERO		PUNTOS SONDEO GEOTECNICO
	SUMIDEROS		RIOS
	MANHOLES		QUEBRADAS
	PUNTOS DE ENTREGAS		CANADAS
	ESTRUCTURAS DE RED		CANALES
RED ACUEDUCTO	LINEA ACUEDUCTO PRESION	PANTANOS	
	LINEA ACUEDUCTO GRAVEDAD	HUMEDALES	
	ACCESORIOS ACUEDUCTO	EMBALSE	
	MEDIDOR ACUEDUCTO	LAGOS Y LAGUNAS	
	ACOMETIDAS ACUEDUCTO	POZOS DE SUCCION	
	HIDRANTES ACUEDUCTO	CUENCA PLUVIAL	
	VALVULAS DE SISTEMA ACUEDUCTO	CUENCA SANITARIA	
	ESTRUCTURA DE CONTROL ACUEDUCTO	CUENCA COMBINADA	
	ESTRUCTURA RED ACUEDUCTO	ISOYETAS	
		ESTACIONES HIDROMETEOROLOGICAS	
BASICOS ACUEDUCTO	PUNTOS GEODESICOS	GEOMORFOLOGIA	PUNTES PEATONALES
	PUNTOS TOPOGRAFICOS	AMENAZAS	CICLO RUTA
	PUNTOS CONTROL DIRECTO	RIESGOS	CORREDORES
	GRILLA 5000	PERTECUBA VEGETAL	LINEA TRANSMILENIO
	GRILLA 2000	MICROZONIFICACION SISMICA	UPZ
	GRILLA 1000	ECOLOGIA	LOCALIDAD
	GRILLA 25000	RED ENERGIA	DISTRITO
	CURVA DE NIVEL INDICE	RED GAS	ZONA URBANA
	CURVA DE NIVEL INTERMEDIA	RED TELECOMUNICACIONES	ZONA RURAL
	CURVA DE NIVEL APROXIMADA	PUNTOS ENERGIA	MUNICIPIOS
	SITIOS DE INTERES	PUNTOS GAS	ZONA POT
	PUNTOS IDF	PUNTOS TELECOMUNICACIONES	ZONA POSTAL
	LOTEO	CRUCES CON SEMAFORO	ESTRATO URBANO
	LOTEO EAAB	CRUCES SIN SEMAFORO	USO SUELO
	BARRIOS	PUNTES VEHICULARES	SECTOR DANE
	MANZANO	TUNEL VEHICULAR	ZONA EXPANSION URBANA
	CONSTRUCCIONES	ROUND POINT	
	SERVIDUMBRES	VIA PRINCIPAL	
	PARQUES	VIA SECUNDARIA	
	DECLARATORIAS DE UTILIDAD PUBLICA	AEROPUERTO	
ALAMEDAS	TERMINAL FERREO		
SARDINEL	TERMINAL TERRESTRE		
GEOLOGIA	CAMINOS		

Figura 32. Niveles de Información para cada modelo de datos de la EAAB

#### 4.3.3.3. Tipos de Servicios

Para la determinación de los tipos de servicios que se crearán, se tuvieron en cuenta los requerimientos funcionales descritos en la fase de análisis, tomados a partir de los estándares WMS y WFS, de los cuales se estipula que:

- Un Web Map Service: será el servidor de mapas.
- Un Web Feature Service: será el servidor de rasgos.

Con base en los dos servidores establecidos, a continuación se describen los casos en los cuales se tendría acceso a cada uno de los servidores:

### WMS

- ④ Servicio de imagen, en el cual la funcionalidad se limita a una vista y consulta de un mapa.
- ④ Servicio de superposición de capas, en la cual se requiere la visualización de una o más capas.
- ④ Servicio de georeferenciación, cuando se quiera obtener y visualizar una ubicación específica dentro de un mapa (esta función no aplica para éste prototipo).

### WFS

- ④ Servicios de rasgos, tales como interacción y análisis extenso por parte del usuario.
- ④ Las capas de datos son usadas para un análisis especializado.
- ④ Obtención de características de determinados rasgos sobre el mapa que se está consultando.

#### **4.3.3.4. Diseño de Base de Datos**

Para el modelado de la base de datos, se aplica el modelo de datos en las 3 fases que propone la guía metodológica: Modelo de capacidades de los servidores (Figura 15), Modelo de datos geográficos (Figura 16) y Modelo de seguridad (Figura 17). Dichos modelos fueron diseñados tal como lo propone la guía metodológica de manera general, para las capas del modelo temático que fueron suministradas por la entidad:

- ④ Localidades
- ④ Zonas EAAB
- ④ Ríos
- ④ Quebradas
- ④ Lagos y lagunas
- ④ Embalses
- ④ Humedales

#### **4.3.3.5. Diseño de la Arquitectura**

Para el diseño de la arquitectura se aplicó el modelo multicapa DLAC (Data, Service Logic, Adapter Logic, Client) planteado, el cual aporta para las labores de eficiencia y seguridad, propias de una aplicación generadora de servicios en línea, y otros factores propios de un sistema distribuido tales como mantenibilidad, modularidad y reutilización.

El sistema que se plantea tiene servidores WMS y WFS ubicados en diferentes puntos geográficos, que proveen funciones especializadas por medio del uso de objetos remotos para cada uno de los servidores.

El servicio Web, como adaptador se encargaría de la invocación remota de los objetos WMS y WFS y posteriormente el llamado a los métodos que necesita, y de ésta manera no obligar a tener una infraestructura localizada en un solo sitio, ya que esto puede ser un factor de riesgo y de costo para la entidad que maneja información con cierto grado de confidencialidad.

Por otra parte, el uso de patrones de software planteados en la guía metodológica, se consideran un factor relevante en ésta fase de diseño, teniendo en cuenta que son paradigmas de programación que en tiempo de ejecución trae ventajas como eficiencia, calidad y en tiempo de mantenimiento ventajas como facilidad de cambios así como ajustes en la información que la EAAB publique por medio de los servicios en línea que provea.

Los patrones de software sugeridos y aplicados en el desarrollo de éste prototipo son:

- Data Access Object
- Abstract Factory
- Factory
- Adapter
- Singleton

La eficiencia y la seguridad son 2 factores esenciales que plantea la guía metodológica como alcance de la misma, por lo cual se siguieron los puntos para proveer correctamente tanto la eficiencia como la seguridad de la siguiente manera:

## **EFICIENCIA**

- Modelo Multicapa
- Patrones de Software
- Memoria Caché
- Estándares Open GIS

## **SEGURIDAD**

- Autenticación
- Autorización

El modelo aplicado fue el modelo multicapa denominado por la sigla **DLAC (Data, Service Logic, Service Adapter, Client)**, el cual se especifica en la arquitectura que se presenta en la Figura 21.

### **4.3.4. Fase de Implementación**

Una vez establecido el diseño, se especificaron las herramientas a utilizar para la implementación del prototipo, las cuales fueron:

- ⇒ Sistema Administrador de Base de Datos: MySQL  
(Ver Numeral 2.4.1.1).
- ⇒ Plataforma de Desarrollo: Microsoft Visual Studio .NET.  
(Ver Numeral 2.4.2.1).
- ⇒ Lenguaje de Programación: C#

Las razones por las cuales se seleccionaron dichas herramientas fueron debido a son software libre, y además de ello por conocimiento, experiencia, facilidad de adquisición de productos, y las características descritas en los numerales 2.4.1 y 2.4.2. del marco teórico como herramientas de desarrollo.

La arquitectura se modeló por clases, cuyo esquema se encuentra en el Anexo Q.

#### **4.3.4.1. Patrones de Implementación**

Se implementó cada componente de la solución únicamente en un nivel físico, en el cual se probó el funcionamiento del sistema, pero se tuvo el soporte de copias diarias por versiones.

Para la integración de servicios, en este caso en el adaptador de servicios WMS y WFS, se utilizó un patrón que consiste en la separación de componentes empresariales de agentes e interfaces de servicios, lo cual es proveído por las arquitecturas distribuidas. Los patrones de software que se utilizaron fueron los planteados en la fase de diseño.

Para ésta fase de implementación, se estableció un contexto en el cual se implementó la funcionalidad por partes, es decir por funciones de las capas más simples, hasta las de mayor nivel, para conseguir un producto de mayor calidad en menor tiempo, lo cual fue una buena alternativa porque se detectaron los errores con mayor facilidad.

El levantamiento de los datos de prueba fue realizado de la siguiente manera:

- ⇒ La entidad suministró los archivos shape con sus respectivos archivos asociados de las 7 capas (Localidades, Humedales, Quebradas, Embalses, Lagos y lagunas, Ríos, y Zonas de la EAAB).
- ⇒ Los archivos shape fueron transformados a scripts SQL por medio de un software libre llamado deggree.
- ⇒ Los scripts SQL fueron adaptados al modelo de datos propuesto por la guía metodológica y posteriormente fueron ejecutados para migrar la información a la base de datos diseñada para cada uno de los servidores.
- ⇒ La información no disponible como la relacionada a capacidades de cada uno de los servidores y datos de seguridad, fueron creados e insertados en el modelo relacional correspondiente, de tal modo que la información quedó completa para la generación de los servicios básicos WMS y WFS.

#### **4.3.4.2. Modelo de clases e interfaces para la arquitectura por capas**

Se creó una solución para almacenar los proyectos asociados al prototipo. A continuación se describen cada una de las capas del modelo multicapa propuesto, con sus respectivos paquetes o proyectos asociados y a su vez con las clases que forman parte de cada uno de dichos paquetes.

##### Capa de Datos

Se creó un proyecto para almacenar la capa de datos, la cual se compone de las siguientes clases e interfaces:

-  AbstractFactory.cs
-  CapabilitiesWFSImpl.cs
-  CapabilitiesWMSImpl.cs
-  DAO.cs
-  DAOFactory.cs
-  ICapabilitiesWFS.cs
-  ICapabilitiesWMS.cs
-  IDAO.cs
-  ILayer.cs
-  IWFSSeguridad.cs
-  IWMSSeguridad.cs
-  LayerImpl.cs
-  MSFactory.cs
-  WFSSeguridadImpl.cs
-  WMSSeguridadImpl.cs

### Capa de Lógica de Servicio

Se crearon 5 proyectos asociados a la lógica de los servicios WMS y WFS, los cuales se describen a continuación con sus clases correspondientes:

- *Proyecto/Paquete GML*

-  WFSOperations.cs
-  WMSOperations.cs

- *Proyecto/Paquete WindowsServiceWFS*

-  ProjectInstallerWFS.cs
-  ServiceWFS.cs

- *Proyecto/Paquete WindowsServiceWMS*

-  ProjectInstallerWMS.cs
-  ServiceWMS.cs

- *Proyecto/Paquete ServerWMS*

-  WMS.cs

- *Proyecto/Paquete Server WFS*

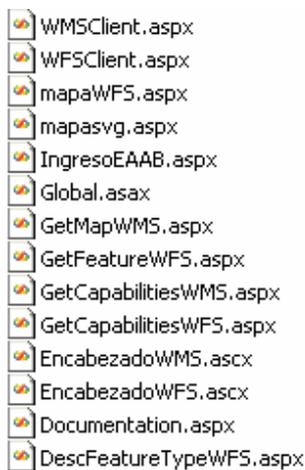
-  WFS.cs

### Capa de Adaptador de Servicio

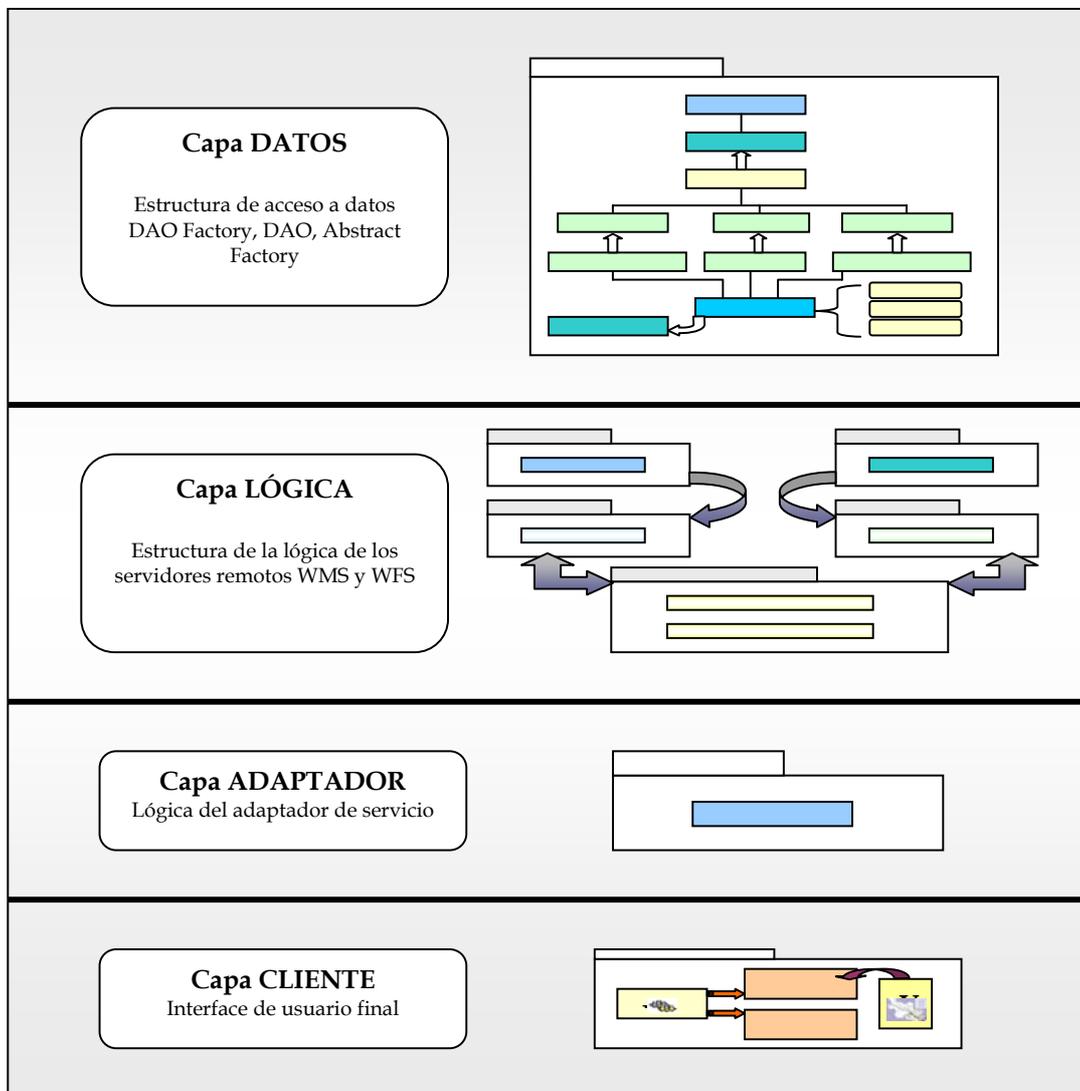
Se creó un proyecto el cual contiene un Web Service, que juega el papel de adaptador de los servicios WMS y WFS. Está compuesto por una única clase que es el servicio Web que publica los métodos que el cliente invocará.

### Capa de Cliente

Se creó un proyecto el cual contiene las clases asociadas a la interface de usuario final, compuesto por las siguientes clases:



La Figura 33 muestra la representación del modelo de clases por capa, descrito anteriormente.



**Figura 33. Esquema de paquetes por capas del modelo DLAC**

#### 4.3.4.3. Implementación de las operaciones básicas de WMS y WFS

Con base en los 3 tipos de operaciones que propone la guía metodológica para agrupar las operaciones básicas de WMS y WFS, se implementaron las operaciones con base en los procedimientos planteados en la guía; dichas operaciones fueron exitosamente implementadas debido a que fueron probadas cada una para corregir los errores de manera más sencilla.

#### 4.3.4.4. Diseño de Interfaces de usuario

El diseño de las interfaces del usuario Web, se realizó de la siguiente manera:

- ⇒ Se creó la página de ingreso en la cual se validan las entradas del usuario en los controles de interface asociados.
- ⇒ Se creó una página para los servicios WMS, en la cual se tiene un control de texto para presentar al usuario los documentos XML y GML resultados de las operaciones WMS, así como un visor SVG requerido para la visualización de los mapas.
- ⇒ Se creó una página para los servicios WFS en la cual se tienen tanto los controles asociados para las entradas del usuario, como los controles para la visualización de las salidas de las operaciones del servicio.
- ⇒ Para los controles de usuario Web, se expusieron únicamente las propiedades y los métodos públicos que se necesiten realmente, para facilitar labores de mantenimiento.
- ⇒ Las funciones de control invocan a las acciones del componente de proceso de usuario.

#### 4.3.4.5. Despliegue de Información al Cliente

Los tipos de datos que se despliegan al cliente son:

- ⇒ Datos o atributos gráficos para el caso de GetMap de WMS.
- ⇒ Datos o atributos no gráficos para las operaciones: GetCapabilities de WMS y WFS, DescribeFeatureType y GetFeature de WFS.

De las funciones mencionadas en la guía metodológica, se aplicó en el prototipo las siguientes:

##### Especificar condiciones

En el caso de las operaciones WFS: DescribeFeatureType y GetFeature, se tiene el ingreso de parámetros para la consulta por medio de filtros dados por la selección desde opciones predefinidas para capas y rasgos de los mapas. Luego de dadas las condiciones que el usuario requiere, se obtiene la respuesta esperada, en documentos GML y XML. En cada respuesta se presenta un listado de todos los rasgos o capas que reúnen la condición.

#### **4.3.5. Fase de Soporte y Mantenimiento**

La implantación de la aplicación se llevó a cabo en las instalaciones de la Universidad, adecuando la infraestructura distribuida que se diseñó.

Posteriormente, se realizaron una serie de pruebas para comprobar el correcto funcionamiento del sistema.

##### **4.3.5.1. Pruebas al sistema**

Para la validación y prueba del sistema propuesta en la guía metodológica, se construyeron unos scripts de prueba, basados en los formatos diseñados para dicho fin. Posteriormente, se ejecutaron con datos reales de la EAAB, y conforme a los resultados obtenidos se realizó una depuración de los errores.

Como se mencionó en la guía, la fase de prueba no sirve para demostrar que no hay fallos sino para encontrarlos, se comprobó que en el transcurso del desarrollo del prototipo es de vital importancia la fase de pruebas, para garantizar la calidad del software, por medio de la detección y corrección a tiempo de los problemas.

Por cada requerimiento del usuario, especificado en los casos de uso, se realizaron los scripts de pruebas correspondientes, de modo que se probó la funcionalidad por partes, y luego en su totalidad.

Los scripts de pruebas desarrollados se presentan en el CD Anexo.

##### **4.3.5.2. Despliegue e instalación del sistema**

Después de haber realizado pruebas al sistema por medio de datos reales de la entidad EAAB y de los scripts mencionados en el punto anterior, se prosiguió con la instalación de la aplicación con base en las condiciones adecuadas para dicho fin, de esta forma se tuvo en cuenta los siguientes aspectos:

Los requerimientos no funcionales que se tuvieron en cuenta para que se pudiese hacer la instalación correcta fueron los siguientes:

- ⇒ El sistema operativo de la máquina, fuese Windows, preferiblemente 2000 en adelante.
- ⇒ La máquina en donde se fuese a instalar el Web Service y la parte cliente tuviera instalado IIS (Internet Information Server), que viene con el sistema

operativo, ya que este serviría como servidor Web, además de ello se tuvo en cuenta la creación de un directorio virtual en el IIS que apuntase a la carpeta que contenía las DLL y los aspx.

⇒ La máquina en donde se fuese a instalar tanto el servidor WMS como el WFS, tuviesen instalado el motor de Base de Datos MySQL, ya que la ambas bases de datos fueron creadas en este motor.

Después de realizar la correcta instalación de la aplicación conforme a lo mencionado anteriormente, existe una etapa en la que hay dar entrenamiento al usuario de cómo manejar la herramienta debidamente, para ello, se realizó un Manual de Usuario y Manual de Instalación, que hace parte de documentos anexos de esta investigación que se encuentran en el CD.

#### **4.3.5.3. Mantenimiento del Sistema**

Después de haber cumplido con todos los pasos anteriores y haber puesto en marcha la aplicación, para el desarrollador viene la responsabilidad de mantener el sistema que se diseñó e implementó, en el caso del prototipo este paso se realizaría en caso de que ocurriese algún fallo en las operaciones que presta la aplicación, caso que no debiese ocurrir, porque en el paso anterior se han debido hacer las pruebas suficientes para asegurar el correcto funcionamiento del programa, o en caso de que de que surgiese un nuevo requerimiento, lo cual conllevaría a una nueva iteración en el desarrollo.

Lo anterior implicaría volver a estudiar los pasos anteriores con detenimiento, para determinar si los cambios y nuevas funcionalidades son posibles de adecuar en la aplicación, la ventaja que existe, es el diseño que se hizo, teniendo en cuenta que provee desacoplamiento y fácil adaptabilidad a los cambios, tanto de motor de Base de Datos, como de adición de nueva funcionalidad, dentro de las operaciones que se plantearon en la guía metodológica, cabe resaltar que se podrían prestar los siguientes servicios al usuario, después de la puesta en marcha de la aplicación:

- Monitorización de los servicios del sistema:

Prestar el servicio de mantenimiento de las bases de datos, para asegurar que se este haciendo la correcta administración de las mismas, ya que ello podría afectar el rendimiento de los servicios que presta la aplicación, debido al crecimiento o inadecuada gestión de los datos al interior de la Base de Datos Geográfica.

- Mantenimiento y actualización del sistema.

En caso de que surja un nuevo requerimiento por parte del usuario, o si se deseara implementar las otras funciones que especifican los estándares, estos se incorporarán al sistema, siguiendo de nuevo las etapas del ciclo de vida y determinando el nivel de impacto que causará sobre el sistema existente.

#### **4.4. Análisis de eficiencia en la generación de servicios de Mapas**

Para el análisis de eficiencia de la aplicación se tuvieron en cuenta los dos factores planteados por la guía Tiempo y Calidad. Lo que se pretende con esto es medir el tiempo de respuesta al cliente en una solicitud con la hipótesis de obtener una respuesta de alta calidad en el menor tiempo posible. Para lograr dicho fin se tuvieron en cuenta los factores de eficiencia sugeridos por la guía.

- Modelo Multicapa:

El prototipo se diseñó basado en la arquitectura DLAC, la cual provee varias capas, factor que implica mayor eficiencia en los procesos teniendo en cuenta que cada capa se especializa en sus funciones. Además, éste modelo provee facilidad en el mantenimiento del software, adaptabilidad al cambio y un factor importante de resaltar es que la comunicación se hace a través de documentos XML, los cuales inherentemente brindan interoperabilidad al sistema y permiten que el flujo de información entre capas sea liviano, y con ello se reduce el tráfico en la red.

- Patrones de Software:

Este es un factor que se tuvo en cuenta, tanto en el tema de eficiencia como de mejores prácticas de programación, en el caso del prototipo como se menciona en la guía metodológica se usaron para solucionar problemas con cierta complejidad como en el caso del patrón Factory que permite tener una fábrica de proveedores de Fuentes de Datos y el fácil cambio de Motor de Base de datos, sin implicar un cambio drástico en el diseño y lógica de la aplicación.

En el caso de los servicios en línea, se determinó que además de seguir los estándares del OGC (Open GIS Consortium) era importante el uso de patrones de software con la finalidad de hacer mejor uso de los recursos de software disponibles, y para ofrecer mantenibilidad, reutilización del código y reducción

de la complejidad que se presenta en cada uno de los procesos del desarrollo del software.

➤ Memoria Caché:

En el prototipo este fue uno de los factores que permitió mejor tiempo de respuesta a las solicitudes del cliente, teniendo en cuenta que el manejo adecuado de la memoria dio mayor eficiencia, en el sentido de ahorro de trabajo y tiempo en el flujo de la información, pues después de haber sido procesada en la primera ocasión se almacena para su posterior consulta.

➤ Estándares Open GIS:

Los estándares propuestos por el OGC (Open GIS Consortium) permiten la interoperabilidad entre sistemas, debido a que la comunicación es realizada a través de documentos XML, en el caso del prototipo, se buscó cambiar la forma tradicional de mostrar mapas al lado del cliente, es decir, no enviar la imagen en formato GIF, JPG, entre otras, sino enviar una codificación GML que a través de una hoja de estilo pudiese ser convertida a formato SVG e interpretada al lado del cliente, con tan solo tener instalado un plug-in para su visualización, este procedimiento es el propuesto por los estándares y es un tema que aporta eficiencia, mayor facilidad para manipular y adecuar el mapa según el cliente lo necesite.

#### **4.4.5. Uso de la herramienta ArcIMS**

En esta parte de la investigación se hizo uso de la herramienta ArcIMS, cuya arquitectura y funcionamiento fueron tratados en el capítulo del Marco Teórico de este trabajo.

Cabe aclarar que el ambiente de prueba para demostrar la eficiencia del prototipo que se construyó paralelo a esta investigación, frente a esta herramienta, no cumple con las mismas características de arquitectura, por ello se harán varias pruebas y se obtendrá un margen de error, al igual que se mencionarán los aspectos que marcan la diferencia entre las dos aplicaciones.

## Desarrollo

Para desarrollar las pruebas se construyó un servicio Web por medio de la herramienta ArcIMS, para ello fue necesario:

- ✓ Tener instalada la aplicación ArcIMS 4.0.1.
- ✓ Archivos shape de las capas a comparar.
- ✓ Construcción del servicio Web por medio de la herramienta.

### **4.4.6. Uso del modelo planteado en la Guía Metodológica**

DLAC(Data, Service Logic, Service Adapter, Client), es la arquitectura planteada por la guía metodológica la cual hace uso de una arquitectura multicapa, explicada en el capítulo de Guía Metodológica en el numeral 3.6.5.5, con base en esta se construyó el prototipo y para realizar las pruebas se hará uso de las mismas capas con las que se construyó el servicio Web en la herramienta ArcIMS.

### **4.4.7. Análisis Comparativo de eficiencia**

La eficiencia será analizada en tiempo y calidad, en el proceso de solicitud y respuesta de mapas geográficos.

Para llevar a cabo dicho fin se establecieron los siguientes criterios para realizar la comparación:

- ④ Capas de datos iguales en ambos casos.
- ④ Uso del mismo explorador Web, en este caso Internet Explorer 6.0.
- ④ Solicitud de capas de datos por separado.
- ④ Solicitud de capas de datos sobrepuestas.
- ④ Solicitud de capas de datos más de una vez.
- ④ La operación que se comparará será GetMap de WMS, una de las 5 que provee el prototipo.

La Tabla 20. presenta un análisis de tiempos de respuesta de ArcIMS y la Tabla 21 los de el prototipo.

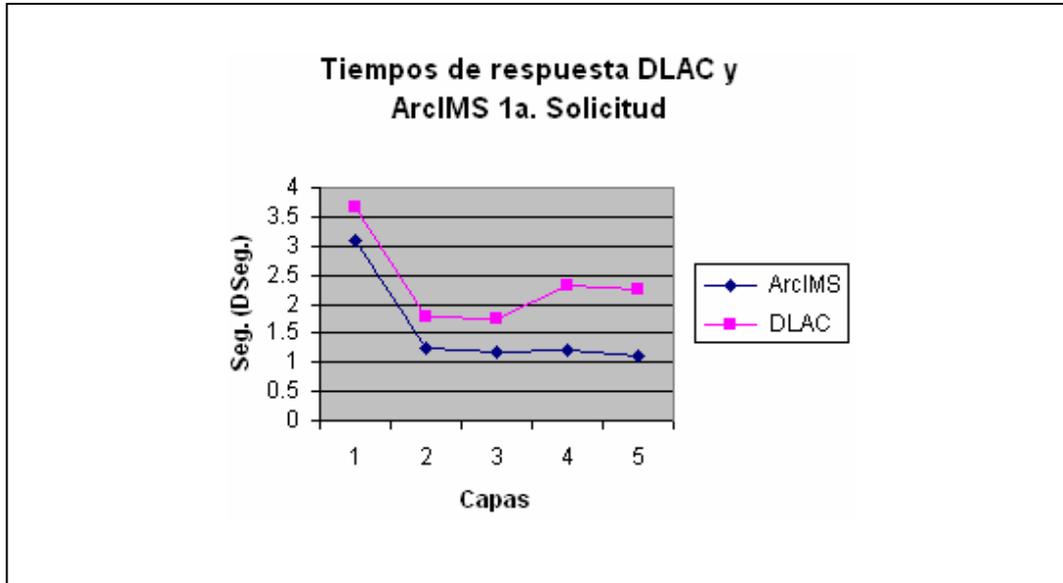
Criterio	ArcIMS			
	1ra Vez	2da vez	3ra Vez	4ª vez
<b>SOLICITUD DE UNA CAPA</b>				
🌐 Capa Localidades	3"08	2'85	2'84	2"87
🌐 Capa Embalse	1"26	1'12	1'09	1"11
🌐 Capa Humedal	1"16	1'	1'10	1"08
🌐 Capa Río	1"20	1"25	1"23	1"26
🌐 Capa Quebrada	1"11	1"09	1"00	1"05

**Tabla 20. Análisis de tiempos de respuesta ArcIMS y Prototipo**

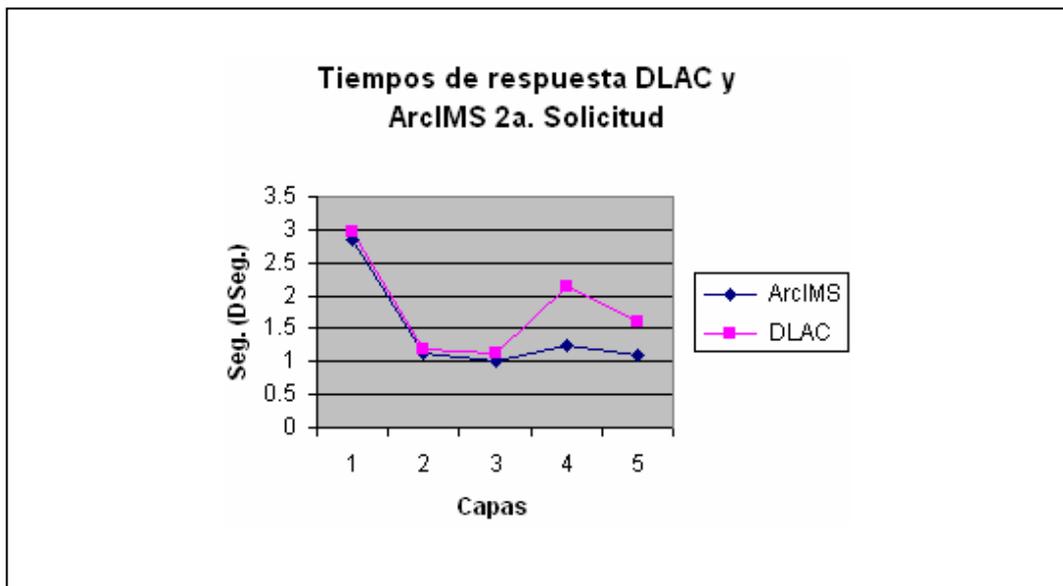
Criterio	Prototipo (DALC)			
	1ra Vez	2da vez	3ra Vez	4ª vez
<b>SOLICITUD DE UNA CAPA</b>				
🌐 Capa Localidades	3"66	2"98	2"90	2"95
🌐 Capa Embalse	1"78	1"19	1"13	1"15
🌐 Capa Humedal	1"75	1"12	1"11	1"13
🌐 Capa Río	2"32	2'14	2"10	2"12
🌐 Capa Quebrada	2"25	1'59	1"60	1"60

**Tabla 21. Análisis de tiempos de respuesta ArcIMS y Prototipo**

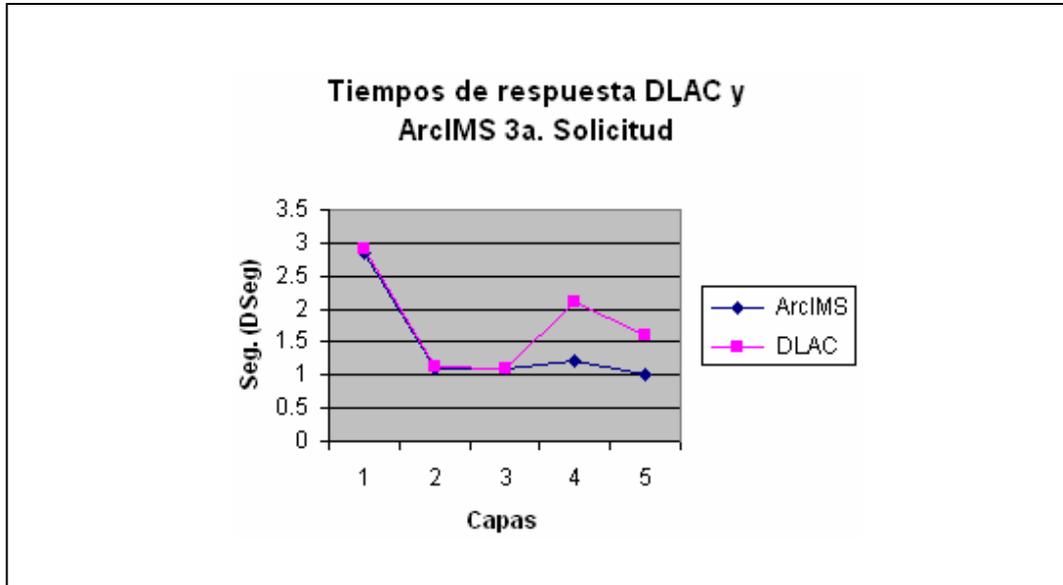
En la Figura 34. Se muestra un esquema en el cual se representa gráficamente el análisis comparativo de tiempos.



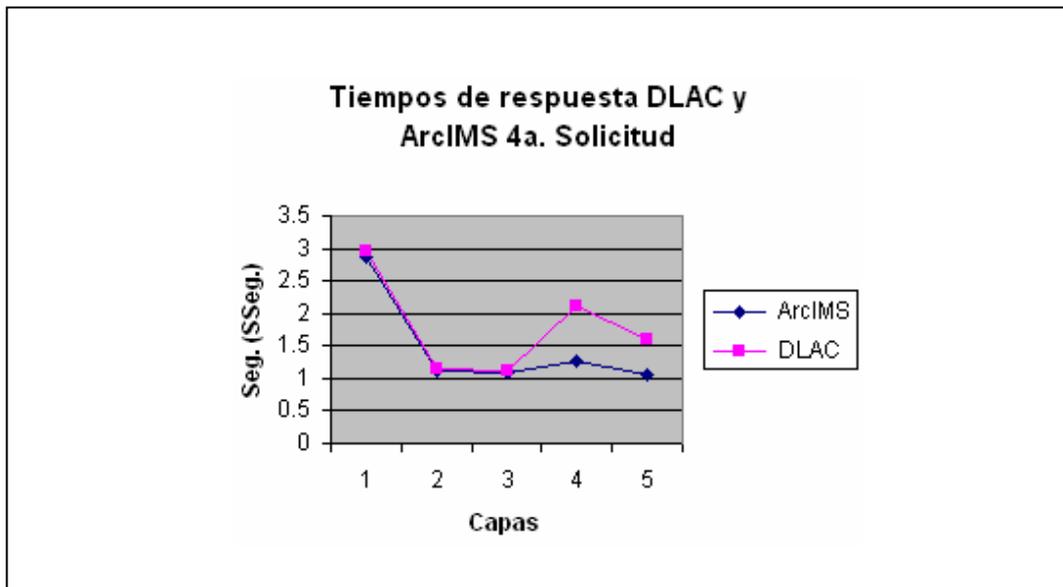
**Figura 34. Esquema comparativo DLAC y ArcIMS 1a Solicitud**



**Figura 35. Esquema comparativo DLAC y ArcIMS 2a Solicitud**



**Figura 36. Esquema comparativo DLAC y ArcIMS 3a Solicitud**



**Figura 37. Esquema comparativo DLAC y ArcIMS 4a Solicitud**

## **Análisis**

Después de haber realizado las pruebas de funcionalidad y tiempos con ambas herramientas, se encontraron las siguientes conclusiones:

- ✓ El prototipo en comparación con la herramienta ArcIMS provee bastante eficiencia en el cargue de capas, a pesar de que en la primera solicitud se tarde un poco más que esta herramienta comercial, con esto no se afirma que el prototipo sea mejor que esta herramienta comercial, debido a que las condiciones de cada sistema son diferentes.
- ✓ El prototipo ofrece un tipo de formato de imagen distinta a la ofrecida por ArcIMS, SVG, el cual es un formato con bastantes ventajas frente a los formatos tradicionales, entre otras la calidad de color es excelente, admite efectos como sonido, efectos visuales al hacer clic o mover el ratón, etiquetas informativas y esto puede ser realizado por el usuario con tan solo descargar el archivo y hacer las modificaciones que requiera.
- ✓ El formato SVG que utiliza el prototipo es el sugerido por el OGC para la visualización de mapas en el estándar WMS.
- ✓ La utilización de caché disminuye el tiempo de solicitud de una capa.
- ✓ El envío de codificación SVG a través de un string es un a manera eficiente de transmitir las solicitudes hechas por los clientes, la eficiencia podría mejorarse comprimiendo este tipo de dato, por medio de compresores existentes, en este caso para .Net.
- ✓ El prototipo ofrece seguridad a nivel de autenticación y autorización de los usuarios que se conectan a consultar las capas de un mapa.

## 5. CONCLUSIONES

1. La propuesta realizada como fruto del Proyecto de Grado, de una Guía Metodológica para la construcción de servicios en línea, es una forma de ayudar a los usuarios con conocimientos en SIG y aplicaciones Web, a desarrollar aplicaciones que generen servicios en línea, con los elementos necesarios para proveer seguridad y eficiencia.
2. El levantamiento de los requerimientos funcionales y no funcionales por medio de los formatos desarrollados en la guía metodológica, facilitaron el manejo y el entendimiento de cada uno de los requerimientos basados en los estándares WMS y WFS, por lo cual se logró llevar un seguimiento de los mismos en las fases de Análisis, Diseño, Implementación y en la fase de pruebas para las validaciones de cada una de las operaciones que se soportan en los requerimientos adquiridos.
3. A través del desarrollo de éste trabajo de grado, se concluyó que desde el punto de vista de la *interoperabilidad* de los servicios geográficos, los estándares WMS y WFS escritos por el OGC (Open GIS Consortium), aún no son muy conocidos, y por ende menos utilizados en el ambiente Colombiano, a pesar de ser de vital importancia para desarrollar servicios en línea, que puedan integrarse con otros servicios, de acuerdo a las especificaciones de comunicación descritas por cada uno de los estándares.
4. El formato SVG para la visualización de imágenes es un formato que ofrece bastantes ventajas como la calidad de color, la inclusión de scripts que modifican el gráfico dinámicamente en función de las necesidades y la facultad de edición del texto que se incluye. Por otro lado, al estar basado en XML, el cual es un formato extensible, los fabricantes pueden adoptar SVG como formato nativo de sus aplicaciones, añadiendo las características específicas que deseen, manteniendo la compatibilidad básica y universal con toda aplicación que reconozca el formato, lo cual provee de igual forma interoperabilidad.

5. El diseño de la arquitectura planteado por la guía metodológica, basado en 4 capas: Capa Cliente, Adaptador de Servicios, Lógica de Servicios, y Capa de Datos, provee un mayor desempeño tanto para el desarrollo del sistema, como para su implementación y mantenimiento, debido a que cada capa se especializa en sus funciones, de modo que se aumente la cohesión y se reduzca el acoplamiento entre los componentes del sistema.
6. La guía metodológica propuesta, permitió determinar que con el uso de software libre, es posible generar aplicaciones de gran tamaño y de alta calidad, particularmente con el manejo de información geográfica, la cual posee cierta complejidad debido a su gran tamaño y estructura de los datos. A pesar de que actualmente existen herramientas comerciales que ofrecen funciones para generar servicios geográficos en la Web, el uso de software libre se plantea como una alternativa viable, en la medida en que se cuente con las condiciones aptas para el uso adecuado del mismo, como conocimiento por parte del recurso humano de dichas herramientas de software libre.
7. Como resultado de la comprobación de la Guía Metodológica en el caso de estudio se encontró que un correcto seguimiento de la guía, permite conseguir los resultados esperados, y por otra parte se logró determinar que la entidad a la cual se realizó el caso de estudio tomó conciencia de la importancia del uso de estándares geográficos, con el objetivo de conseguir calidad, eficiencia, seguridad e interoperabilidad en los servicios que ofrecen.

## 6. BIBLIOGRAFÍA

Sistemas de Información Geográfica

<http://www.ncgia.ucsb.edu> Fecha de Consulta: Agosto de 2003

Fundamentos de SIG

MICHAEL N. DeMers, Fundamentals of geographic information systems. New York : John Wiley, 1997 Fecha de Consulta: Agosto de 2003

[http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/vector\\_raster.htm#comparacion](http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/vector_raster.htm#comparacion). Fecha de Consulta: Agosto de 2003.

Bases de Datos espaciales

RIGAUX Philippe, Spatial Databases: with Applications for GIS. San Francisco, CA : Morgan Kaufmann Publishers, c2002. Fecha de Consulta Febrero 2004

Bases de Datos Geográficas

<http://www.esri.com> Fecha de Consulta: Enero de 2004

Administración de Sistemas de Información Geográfica

ARONOFF Stanley Geographic information systems: a management perspective. Ottawa : WDL, 1995. Fecha de Consulta: Abril de 2004.

Análisis y Diseño de Sistemas de Información

LESZEK A Maciaszek Requirements analysis and system design : developing information systems with UML Harlow, England : Addison-Wesley, 2001. Fecha de Consulta: Octubre de 2003.

Web Services

<http://www.w3.org/TR/ws-arch/#whatis> Fecha de Consulta: Octubre 2003

Web Services (XML, SOAP, WSDL, UDDI)

GRAHAM Steve, SIMEINOV Simeon, BOUBEZ Toufic, DAVIS Doug, DANIELS Glen, NAKAMURA Yuichi y NEYAMA Ryo. Building Web Services with Java TM Making sense of XML, SOAP, WSDL, and UDDI. 2002. SAMS Publishing Fecha de Consulta Febrero de 2003

#### Generalidades Web Services

CHAPARRO, Hilda Web Services. Nota Técnica. Universidad de los Andes, Facultad de Ingeniería. Departamento de Ingeniería de Sistemas y Computación 2002. Fecha de Consulta. Agosto de 2003.

#### Web Services Microsoft

<http://msdn.microsoft.com/webservices/> Fecha de Consulta: Noviembre de 2003

#### WSDL

<http://www.w3.org/TR/wsdl> Fecha de Consulta: Diciembre de 2003.

#### WSDL Definiciones Abstractas

<http://www.microsoft.com/spanish/msdn/articulos/archivo/091101/voices/wsdlexplained.asp> - wsdlexplained\_topic11 Fecha de Consulta: Noviembre de 2003.

#### SOAP

<http://www.w3.org/TR/soap12-part1/> Fecha de Consulta: Febrero de 2004.

#### Definición de SOAP Microsoft

<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art51.asp>  
Fecha de Consulta: Febrero de 2004.

#### Esquema de funcionamiento de SOAP

<http://www.revista.unam.mx/vol.3/num1/art3/img01.html> Fecha de Consulta: Marzo de 2004.

#### UDDI

<http://www.uddi.org/> Fecha de Consulta: Abril de 2004-08-03

#### UDDI y Web Services

JASNOWSKI Mike, Java, XML, and Web Services Bible. 2002. Hungry Minds, Inc, (paginas 918)

#### Autenticación y Autorización

MSDN Training. 2310B: Developing Microsoft ASP.Net Web Application using Visual Studio .NET Fecha de Consulta: Mayo de 2004

#### WFS

[https://portal.opengeospatial.org/files/index.php?artifact\\_id=7176](https://portal.opengeospatial.org/files/index.php?artifact_id=7176) Fecha de Consulta: Abril de 2003

#### WMS

[http://portal.opengeospatial.org/files/index.php?artifact\\_id=5316](http://portal.opengeospatial.org/files/index.php?artifact_id=5316) Fecha de Consulta: Abril de 2003

#### GML

<http://opengis.net/gml/01-029/GML2.html>. Fecha de Consulta: Agosto de 2003

#### SVG

<http://www.w3.org/TR/SVG/> Fecha de Consulta: Marzo de 2004

#### SVG Explora Datos GML

<http://www.svgopen.org/2003/papers/SvgExplorerOfGmlData/> Fecha de Consulta: Junio de 2004

#### Convertir de archivos Shape

<http://www.mycgiserver.com/~amri/converter.cocoon.xml> Fecha de Consulta: Marzo de 2004

#### XSLT

<http://www.w3.org/TR/xslt> Fecha de Consulta: Marzo de 2004

#### Especificación XSLT en español

<http://www.sidar.org/recur/desdi/traduc/es/xml/xml-stylesheet-19990629-es.htm>

#### XML

<http://www.w3.org/TR/REC-xml/> Fecha de Consulta: Diciembre 2003

#### Diseño de Objetos

JOYANES AGUILAR, Luis. Programación Orientada a Objetos. Osborne McGraw-Hill.(Páginas 895). Fecha de consulta: Noviembre de 2003.

#### Diseño de Patrones

“Design Patterns: Elements of Reusable Object – Oriented Software”, de Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, 1995, Addison-Wesley, también conocido como el LIBRO GOF(Gang-Of-Four Book). Fecha de Consulta: Mayo de 2004

#### Arquitectura de aplicaciones de .NET: Diseño de aplicaciones y servicios

<http://www.microsoft.com/spanish/msdn/arquitectura/das/distapp.asp>  
Fecha de Consulta: Abril de 2004.

#### MySQL

<http://www.mysql.com/> Fecha de Consulta: Abril de 2004

Herramientas para MySQL

<http://crlab.com/download.html> Fecha de Consulta: Julio de 2004

Oracle

<http://www.oracle.com/database/index.html> Fecha de Consulta: Mayo de 2004

Oracle Spatial

<http://www.oracle.com/technology/software/products/spatial/index.html> Fecha de Consulta: Mayo de 2004

SqlServer

<http://www.microsoft.com/sql/default.msp> Fecha de Consulta: Junio de 2004

Especificación HTTP

<http://www.w3.org/Protocols/HTTP/> Fecha de Consulta: Julio de 2004

Arquitectura Web .NET

<http://www.microsoft.com/spanish/msdn/arquitectura/das/distapp.asp> . Fecha de Consulta: Junio de 2004

Microsoft .NET

RICHTER Jeffrey, Applied Microsoft.NET framework programming. Redmond, Wash. : Microsoft Press, c2002. Fecha de Consulta: Agosto de 2004

Introducción a .NET

Platt, David S., Introducing Microsoft.NET. Redmond, WA : Microsoft Press, 2002 Fecha de Consulta: Junio de 2004

Arquitectura Web J2EE

<http://java.sun.com> Fecha de Consulta: Julio de 2004

Diseño de aplicaciones con J2EE

Designing Enterprise Application with the J2EE Platform, 2ª Edicion

SVG Acrobat

<http://www.adobe.com/support/downloads/product.jsp?product=46&platform=Windows> Fecha de Consulta: Agosto de 2004.

## Anexo A. Parámetros del Request para la operación Get Map de Web Map Service (WMS)

Parámetro Request	Requerido/ Opcional	Descripción
VERSION=version	R	Versión del Request
REQUEST=GetMap	R	Nombre del Request
LAYERS=layer_list	R	Lista separada por comas de una o mas capas del mapa. Opcional si el parámetro SLD está presente.
STYLES=style_list	R	Lista separada por comas de estilos renderizados para capa solicitada. Opcional si el parámetro SLD está presente.
SRS=namespace :identifier	R	Sistema de Referencia Espacial
BBOX=minx,miny,maxx,maxy	R	Esquinas del Bounding box(lower left, upper right) en unidades SRS.
WIDTH=output_width	R	Ancho en pixeles de la ilustración del mapa.
HEIGHT=output_height	R	Alto en pixeles de la ilustración del mapa.
FORMAT=output_format	R	Formato de salida del mapa.
TRANSPARENT=TRUE FALSE	O	Transparencia del fondo del mapa (default=FALSE).
BGCOLOR=color_value	O	Valor hexadecimal del color para el color del fondo (default=0xFFFFFFFF).
EXCEPTIONS=exception_format	O	El formato en el cual las excepciones serán reportadas por el WMS (default=SE_XML).
TIME=time	O	Valor de las veces de la capa deseada.
ELEVATION=elevation	O	Elevación de la capa deseada.
Other sample dimension(s)	O	Valor de otras dimensiones como sea apropiado.
Vendor-specific parameters	O	Parámetros opcionales experimentales.
Los siguientes parámetros son usados solamente con Web Map Services que soportan la especificación de Styled Layer Descriptor		
SLD=styled_layer_descriptor_URL	O	URL del Styled Layer Descriptor (como se define en la especificación SLD).
WFS=web_feature_service_URL	O	URL del Web Feature Service proporcionando los razgos para ser simbolizados usando SLD.

## Anexo B. Parámetros del Request para la operación Get Feature Info de Web Map Service (WMS)

Parámetro Request	Requerido/Opcional	Descripción
VERSION=version	R	Versión del Request
REQUEST=GetFeatureInfo	R	Nombre del Request
<map_request_copy>	R	Copia parcial de los parámetros de la solicitud del mapa, que genera el mapa para aquella información que es deseada.
QUERY_LAYERS=layer_list	R	Lista separada por comas de una o mas capas para ser consultadas.
INFO_FORMAT=output_format	O	Retorna el formato de la información del rasgo(tipo MIME)
FEATURE_COUNT=number	O	Número de rasgos acerca de los cuales retornar información(default=1)
X=pixel_column	R	Coordenada X en pixeles del rasgo (medida sobre el ángulo izquierdo =0)
Y=pixel_row	R	Coordenada Y en pixeles del rasgo (medida sobre el ángulo izquierdo =0)
EXCEPTIONS=exception_format	O	El formato en el cual las excepciones son reportadas por el WMS (default=application/vnd.ogc.se_xml)
Parámetros específicos del vendedor	O	Parámetros opcionales experimentales.

### Anexo C. Parámetros del Request para la operación Get Feature de Web Feature Service (WFS)

Parámetro Request	Requerido/Opcional	Descripción
versión = 1.0.0	R	Versión del Request
service = WFS	R	Servicio
OUTPUTFORMAT = GML2	O	Formato de salida por defecto es GML2
maxFeatures	O	Usado para limitar el número de features (rasgos) que el request devolverá.
typeName	R	Es usado para indicar el nombre del tipo de feature o clase que será consultada.
featureVersion	O	Incluido en el orden que lo acomoda que el sistema que soporta las versiones del rasgo. Un valor de <b>ALL</b> indica que todas las versiones de un rasgo serán obtenidas. Un entero, <i>n</i> , puede ser especificado para retornar la n-ésima versión del rasgo. El número de la versión comienza en 1, la cual es la versión más vieja.
<Query>	R	define cuales serán los tipos de rasgos de la consulta, que propiedades se devolverán y que restricciones (espaciales y no espaciales) serán aplicadas a estas propiedades
<PropertyName>	O	usado para enumerar las propiedades del rasgo que será seleccionada durante la consulta cuyo valor será incluido en la respuesta del request <b>GetFeature</b>
<Filter>	O	Puede ser usado para definir restricciones en un consulta
<GetFeature WithLock>	O	Es funcionalmente similar al elemento <b>&lt;GetFeature&gt;</b> , excepto que este indica que un servicio web feature intenta un candado para los rasgos que serán seleccionados; presumiblemente para update en los rasgos

## Anexo D. Parámetros del Request para la operación Transaction de Web Feature Service (WFS)

Parámetro Request	Requerido/ Opcional	Descripción
versión = 1.0.0	R	Versión del Request
service = WFS	R	Servicio
releaseAction		Usado para el control de candados del rasgo, cuando el request de la transacción se completa. Un valor <b>ALL</b> indica que los candados en todas las instancias de rasgo usan un candado especificado por el <b>&lt;LockId&gt;</b> devuelto cuando la transacción se complete.
<Transaction>		Puede contener cero o más elementos <b>&lt;Insert&gt;</b> , <b>&lt;Update&gt;</b> , o <b>&lt;Delete&gt;</b> que describan operaciones de creación, modificación o destrucción de instancias de rasgo.
<LockId>		Usado para especificar que la transacción será aplicada previamente al conjunto de candados las instancias de rasgo.

## Anexo E. Parámetros del Request para la operación Lock Feature de Web Feature Service (WFS)

Parámetro Request	Requerido/Opcional	Descripción
versión = 1.0.0	R	Versión del Request
service = WFS	R	Servicio
<Lock>	R	Define un candado de operación en una instancia de rasgo de un tipo de rasgo.
<Filter>	O	Usado para definir el conjunto de instancias de rasgo del tipo especificado a las que se les pondrá candado.
Expiry	O	Es usado para determinar el límite de largo que un servicio web feature tendrá un candado en una instancia de rasgo en el evento en que una transacción nunca resulte se levantará el candado. El límite de <i>expiración</i> es especificado en minutos.
LockAction	O	Es usado para el control de como el candado del rasgo será adquirido. Una acción ALL del candado indica que un servicio web feature tratará de adquirir un candado en todos los request de instancias de rasgos.

**Anexo F. Formato de Cuestionario aplicado a las entidades de Bogotá que producen Información Geográfica**

<b>Nombre de la Entidad</b> _____	
<b>Persona Contactada</b> _____	
<b>Área</b> _____	<b>Fecha</b> _____
<ol style="list-style-type: none"><li>1. ¿La entidad en la cual usted labora genera servicios en línea y cuáles son?</li><li>2. ¿Cuál es la interoperabilidad que ofrecen los servicios que genera la entidad, entendiendo por interoperabilidad la manera en que varias fuentes de información interactúan independientemente de la ubicación, plataforma o lenguaje en el cual se encuentra?</li><li>3. ¿Qué fuentes de información utilizan para generarlos?</li><li>4. ¿Quién o quiénes son los encargados de generarlos?</li><li>5. ¿Cómo es el proceso para generarlos?</li><li>6. ¿Cómo se detecta la necesidad de generar nuevos servicios?</li><li>7. ¿Qué tecnologías, estándares u otras herramientas son utilizadas para generarlos?</li><li>8. ¿Existe algún proyecto encaminado a generar servicios en línea basados en información geográfica, de ser así en que están trabajando actualmente?</li><li>9. ¿Quiénes conforman el comité generador de servicios en línea basados en información geográfica y cuáles son sus funciones?</li><li>10. ¿Contemplan la posibilidad de generar otros nuevos, teniendo en cuenta factores como recursos, tecnología y fuentes de información entre otros?</li><li>11. La entidad estaría dispuesta a apoyar un proyecto de investigación para generación de servicios en línea basados en información geográfica? SI / NO ¿Por qué?</li><li>12. ¿Con qué entidad le gustaría compartir información y que tipo de información?</li></ol>	

## Anexo G. Cuadro de criterios de selección para el caso de estudio

CRITERIOS ENTIDADES	Genera servicio línea	Cuáles servicios genera	Manejo interoperabilidad	Fuentes de información	Encargados de generarlos	Proceso de generación	Herramientas y estándares	Proyecto actual	Probabilidad de generar SL	Entidades compartir información	Apoyo a éste proyecto
ICA	Si	Sanidad agro_pecuaria	Poca	Mapas y bases de datos de la entidad	Dependencia Epidemiología Veterinaria	Levantamiento información, generación mapas, tabulación y publicación.	GPS, ArcSIG, ArcView	No	Si existe	Gremios productores pecuarios, Min. Agricultura, DANE, Usuarios	Si
EEAB	Si	Consulta información técnica, mapas interactivos de Bogotá y Georeferenciador	Media	Cartografía con nomenclatura de Catastro, IGAC y Planeación	Dirección SIG, Topografía y Geodesia, Sistema de Información Documental	Gestión de obtención y gestión de uso de datos, para posterior publicación.	Oracle 8i, ArcSD, monitor de datos espaciales, ArcIMS para servicios Web, ArcSIG 8x : depende del cliente, ArcView, ArcInfo, Browser:	Si	Si existe	Entidades que generen beneficios a la empresa	Si
IGAC	Si	Consulta de metadatos(inhabilitado) y mapas estáticos.	Baja	Mapas topográficos y temáticos(catastrales, de suelos), y datos geográficos del instituto.	Oficina de Sistemas	Generación y validación de archivos, y por ultimo la publicación en la página Web	Herramientas comerciales gratuitas para manejo de mapas en la Web. Normas ICONTEC (NTC4611) para consulta de metadatos y XML	Si, outsourcing	Si existe	ICDE y entidades que lo requieran	Si
ETB	Si	Internos: Determinación y consulta de redes, malla pública Externos: AlaFija.com(paginas amarillas de ETB)	Alta	Información de catastro, de las entidades del sitio.	Administración de Soporte y Aprovisionamiento, encargados del SIG.	Planeación, análisis, diseño, implementación y publicación.	DeskMap, ArcInfo, MapGuide, Dreamweaver, ColdFusion, MapObjects Estándares informáticos	Si	Si existe	Planeación Distrital, Catastro Distrital, IDU, Encargados del Espacio Público	Si
CATASTRO	Si	Mapa digital de Bogotá.	Nula	40 niveles de información de cartografía	Departamento de informática y división de cartografía.	Fotografía, fotocontrol, edición final y publicación	Herramientas ESRI(ArcIMS, ArcInfo), estándares Nacionales y del IGAC	Si	Si existe	Alcaldía de Bogotá, IGAC	Si
GAS NATURAL	No	No aplica	Nula	GeoDataBase	Área de Servicio de Planificación Red	No aplica	SIG GeoDataBase Modelo Cartográfico	No	Muy poca	No	No
CODENSA	No	No aplica	No aplica	Base de datos de Codensa, SmallWorld	Dpto. de Gestión del conocimiento	No existe	SmallWorld de Oracle	No	Si existe	Departamentos internos de la empresa	Si

## Anexo H. Formato de Especificación de Requerimientos del Negocio

<b>ESPECIFICACIÓN DE REQUERIMIENTOS DEL NEGOCIO</b>					
<b>Información General</b>					
<b>Fecha</b>	(dd/mm/aaaa)	<b>Usuario</b>	<A quien se aplica el formato>	<b>Cargo</b>	<Cargo de usuario>
<b>Descripción global de requerimientos</b>					
<Se describe a grandes rasgos lo que la organización busca conseguir con el planteamiento de los requerimientos del negocio, es decir el aporte que haría para el cumplimiento de las metas organizacionales>					
<b>Nivel alto de los requerimientos de la Organización</b>					
Identificador	Nombre	Prioridad (%)	Actores	Procedimientos	Descripción
<b>Total de requerimientos</b>		<b>Fecha límite de entrega</b>		(dd/mm/aaaa)	
<b>Observaciones</b>					
<Observaciones generales acerca de los requerimientos a nivel de negocio, recopilados en este formato>					
<b>Aplicado por</b>				<b>Empresa</b>	
<b>Cargo</b>				<b>Firma</b>	

## Anexo I. Formato de Especificación de Requerimientos de Usuario en Casos de Uso

Factores	Descripción
<i>Identificación</i>	<Identificación del caso de uso>
<i>Actor</i>	<Actor que llevará a cabo el caso de uso>
<i>Nombre del Caso de Uso</i>	<Nombre que se le da al caso de uso con la convención: Verbo_Adjetivo>
<i>Meta(Objetivo)</i>	<Objetivo que se debe cumplir una vez se ha llevado a cabo el caso de uso>
<i>Pre Condiciones</i>	<Condiciones que se deben dar para poder llevar a cabo el caso de uso>
<i>Flujo Normal de Eventos</i>	<Flujo de eventos que sigue el caso de uso en el mejor caso, es decir sin errores ni excepciones>
<i>Puntos de Extensión</i>	<Puntos de extensión que tenga el caso de uso, por ejemplo especializaciones de una acción>
<i>Caminos de Excepción</i>	<Los caminos que se siguen cuando ocurren las excepciones>
<i>Post Condiciones</i>	<Las condiciones que se debieron obtener cuando se completara el caso de uso>

## Anexo J. Formato de Especificación de Requerimientos Funcionales

ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES					
<b>Información General</b>					
Fecha		Usuario		Cargo	
<b>Descripción global de requerimientos funcionales</b>					
<Se describe globalmente el objetivo general de los requerimientos funcionales que se van a recopilar en éste formato, teniendo en cuenta el aporte para la organización>					
<b>Nivel detallado de los requerimientos funcionales</b>					
<b>REQUERIMIENTO FUNCIONAL</b>					
Identificador		Nombre			
Actores		Áreas organizacionales			
<b>Listado de Procedimientos</b>					
Id	Nombre	Actores	Entradas	Procesos	Salidas
Prioridad (%)			Total Procedimientos asociados		
<b>REQUERIMIENTO FUNCIONAL</b>					
Identificador		Nombre			
Actores		Áreas organizacionales			
<b>Listado de Procedimientos</b>					
Id	Nombre	Actores	Entradas	Procesos	Salidas
Prioridad (%)			Total Procedimientos asociados		
...					
<b>Total de requerimientos</b>		<b>Fecha límite de entrega</b>			
Aplicado por				Empresa	
Cargo				Firma	

## Anexo K. Formato de Especificación de Requerimientos no Funcionales

<b>ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES</b>			
<b>Información General</b>			
Fecha		Usuario	Cargo
<b>Descripción global de requerimientos no funcionales</b>			
<Se describe globalmente el objetivo general de los requerimientos no funcionales que se van a recopilar en éste formato, teniendo en cuenta el aporte para la organización>			
<b>Nivel detallado de los requerimientos funcionales</b>			
<b>REQUERIMIENTO NO FUNCIONAL</b>			
Identificador		Nombre	
Actores		Recursos	
Restricciones		Prioridad	
Descripción General			
<b>REQUERIMIENTO NO FUNCIONAL</b>			
Identificador		Nombre	
Actores		Recursos	
Restricciones		Prioridad	
Descripción General			
...			
Total de requerimientos		Fecha límite de entrega	
Aplicado por		Empresa	
Cargo		Firma	

## Anexo L. Especificación de Requerimientos del Negocio para la EAAB

<b>ESPECIFICACIÓN DE REQUERIMIENTOS DEL NEGOCIO</b>					
<b>Información General</b>					
<b>Fecha</b>	30/03/2004	<b>Usuario</b>	Ing. Johan Rivera	<b>Cargo</b>	Dirección de SIG
<b>Descripción global de requerimientos</b>					
La entidad busca con el planteamiento de los requerimientos del negocio, determinar el aporte para la organización del prototipo que se va a implantar, teniendo en cuenta que el grupo investigador planteará un modelo el cual se deja abierto para quien lo quiera seguir, con el objetivo de mejorar eficiencia y calidad en la generación de servicios en línea geográficos.					
<b>Nivel alto de los requerimientos de la Organización</b>					
<b>Id</b>	<b>Nombre</b>	<b>Prioridad (%)</b>	<b>Actores</b>	<b>Procedimientos</b>	<b>Descripción</b>
RN_01	Atraer clientes Web	60 (media)	Cientes Web	Procesamiento de servicios en línea eficientes	Atraer clientes por calidad de información
RN_02	Compartir información	70 (media - alta)	Cientes y entidades	Estandarización de datos y de formatos de intercambio.	Utilización de estándares especializados
RN_03	Obtener ventajas competitivas	60 (media)	Cientes Web e internos	Por medio del uso de estándares ofrecer otros servicios en línea para SIG.	Variedad de servicios geográficos en línea
<b>Total de requerimientos</b>		<b>3</b>	<b>Fecha límite de entrega</b>		<b>No aplica</b>
<b>Observaciones</b>					
Los requerimientos consignados en éste formato son levantados con el objetivo de aplicar correctamente la guía metodológica, el alcance no es conseguir una implementación de los mismos en la entidad. Se utilizará información de la entidad para demostrar el cumplimiento de dichos requerimientos del negocio a través de la guía metodológica.					
<b>Aplicado por</b>	Bibiana Lara - Ana Maria Rodríguez		<b>Empresa</b>	Universidad Javeriana	
<b>Cargo</b>	Grupo investigador Ing. Sistemas		<b>Firma</b>	Bibiana Lara - Ana M <sup>a</sup> Rodríguez	

## Anexo M. Especificación de Requerimientos Funcionales para la EAAB

<b>ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES</b>					
<b>Información General</b>					
<b>Fecha</b>	06/04/2004	<b>Usuario</b>	Ing. Johan Rivera	<b>Cargo</b>	Dirección de SIG
<b>Descripción global de requerimientos funcionales</b>					
Los requerimientos funcionales que se especifican en este formato, buscan describir claramente la funcionalidad del sistema con respecto a las necesidades actuales de la entidad. Los requerimientos se establecen con base en el alcance del proyecto con el uso de los estándares del OpenGIS WMS y WFS.					
<b>Nivel detallado de los requerimientos funcionales</b>					
<b>REQUERIMIENTO FUNCIONAL</b>					
<b>Identificador</b>	RF_01	<b>Nombre</b>	Obtener las capacidades de un servidor Web Map Server		
<b>Actores</b>	Usuario Web	<b>Áreas organizacionales</b>	Dirección de SIG.		
<b>Listado de Procedimientos</b>					
<b>Id</b>	<b>Nombre</b>	<b>Actores</b>	<b>Entradas</b>	<b>Procesos</b>	<b>Salidas</b>
P01	Autenticarse ante el sistema	Usuario WMS	Usuario, contraseña	Validar el usuario ante el servidor WMS.	Acceso autorizado o denegado.
P02	Solicitar la operación GetCapabilities	Usuario WMS	Request GetCapabilities	Obtener las capacidades del servidor WMS basadas en el estándar WMS.	Documento XML con las capacidades de WMS.
P03	Guardar las capacidades del servidor WMS.	Usuario WMS	Solicitud para guardar las capacidades WMS.	Proveer ubicación y nombre del archivo como se va a guardar el documento.	Archivo almacenado en algún dispositivo.
<b>Prioridad (%)</b>		100	<b>Total Procedimientos asociados</b>		3

<b>REQUERIMIENTO FUNCIONAL</b>					
<b>Identificador</b>	RF_02	<b>Nombre</b>	Obtener un mapa de un servidor Web Map Server		
<b>Actores</b>	Usuario Web	<b>Áreas organizacionales</b>	Dirección de SIG.		
<b>Listado de Procedimientos</b>					
<b>Id</b>	<b>Nombre</b>	<b>Actores</b>	<b>Entradas</b>	<b>Procesos</b>	<b>Salidas</b>
P01	Autenticarse ante el sistema	Usuario WMS	Usuario, contraseña	Validar el usuario ante el servidor WMS.	Acceso autorizado o denegado.
P02	Solicitar la operación GetMap	Usuario WMS	Request GetMap	Obtener el mapa solicitado en formato GML y en formato gráfico liviano.	Documento SVG con los rasgos solicitados del mapa.
P03	Guardar los documentos GML y SVG.	Usuario WMS	Solicitud para guardar el mapa dado por WMS.	Proveer ubicación y nombre del archivo como se va a guardar el documento.	Archivo almacenado en algún dispositivo.
<b>Prioridad (%)</b>	100	<b>Total Procedimientos asociados</b>			3
<b>REQUERIMIENTO FUNCIONAL</b>					
<b>Identificador</b>	RF_03	<b>Nombre</b>	Obtener las capacidades de un servidor Web Feature Server		
<b>Actores</b>	Usuario Web	<b>Áreas organizacionales</b>	Dirección de SIG		
<b>Listado de Procedimientos</b>					
<b>Id</b>	<b>Nombre</b>	<b>Actores</b>	<b>Entradas</b>	<b>Procesos</b>	<b>Salidas</b>
P01	Autenticarse ante el sistema	Usuario WFS	Usuario, contraseña	Validar el usuario ante el servidor WFS.	Acceso autorizado o denegado.
P02	Solicitar la operación GetCapabilities	Usuario WFS	Request GetCapabilities	Obtener las capacidades del servidor WFS basadas en el estándar WFS.	Documento XML con las capacidades de WFS.
P03	Guardar las capacidades del servidor WFS.	Usuario WFS	Solicitud para guardar las capacidades WFS.	Proveer ubicación y nombre del archivo como se va a guardar el documento.	Archivo almacenado en algún dispositivo.
<b>Prioridad (%)</b>	100	<b>Total Procedimientos asociados</b>			3

<b>REQUERIMIENTO FUNCIONAL</b>					
<b>Identificador</b>	RF_04	<b>Nombre</b>	Obtener la descripción de los tipos de rasgos que soporta el servidor Web Feature Server.		
<b>Actores</b>	Usuario Web	<b>Áreas organizacionales</b>	Dirección de SIG		
<b>Listado de Procedimientos</b>					
<b>Id</b>	<b>Nombre</b>	<b>Actores</b>	<b>Entradas</b>	<b>Procesos</b>	<b>Salidas</b>
P01	Autenticarse ante el sistema	Usuario WFS	Usuario, contraseña	Validar el usuario ante el servidor WFS.	Acceso autorizado o denegado.
P02	Solicitar la operación Describe FeatureType	Usuario WFS	Request Describe FeatureType	Obtener la descripción de la estructura de cualquier tipo de rasgo que este pueda servir.	Documento GML con el esquema asociado de los tipos de rasgos.
P03	Guardar los documentos xsd y xml.	Usuario WFS	Solicitud para guardar los documentos obtenidos	Proveer ubicación y nombre del archivo como se va a guardar los documentos.	Archivos almacenados en algún dispositivo.
<b>Prioridad (%)</b>	100	<b>Total Procedimientos asociados</b>			3
<b>REQUERIMIENTO FUNCIONAL</b>					
<b>Identificador</b>	RF_05	<b>Nombre</b>	Obtener rasgos de capas de un mapa.		
<b>Actores</b>	Usuario Web	<b>Áreas organizacionales</b>	Dirección de SIG		
<b>Listado de Procedimientos</b>					
<b>Id</b>	<b>Nombre</b>	<b>Actores</b>	<b>Entradas</b>	<b>Procesos</b>	<b>Salidas</b>
P01	Autenticarse ante el sistema	Usuario WFS	Usuario, contraseña	Validar el usuario ante el servidor WFS.	Acceso autorizado o denegado.
P02	Solicitar la operación GetFeature	Usuario WFS	Request GetFeature	Obtener los rasgos solicitados de cada una de las capas, en formato GML.	Documento GML con las capacidades de WFS.
P03	Guardar los rasgos de las capas solicitadas.	Usuario WFS	Solicitud para guardar los rasgos por capas.	Proveer ubicación y nombre del archivo como se va a guardar el documento.	Archivo almacenado en algún dispositivo.
<b>Prioridad (%)</b>	100	<b>Total Procedimientos asociados</b>			3
<b>Total de requerimientos</b>	5	<b>Fecha límite de entrega</b>			No aplica
<b>Aplicado por</b>	Bibiana Lara - Ana María Rodríguez	<b>Empresa</b>	Universidad Javeriana		
<b>Cargo</b>	Grupo investigador Ing. Sistemas	<b>Firma</b>	Bibiana Lara - Ana M <sup>a</sup> Rodríguez		

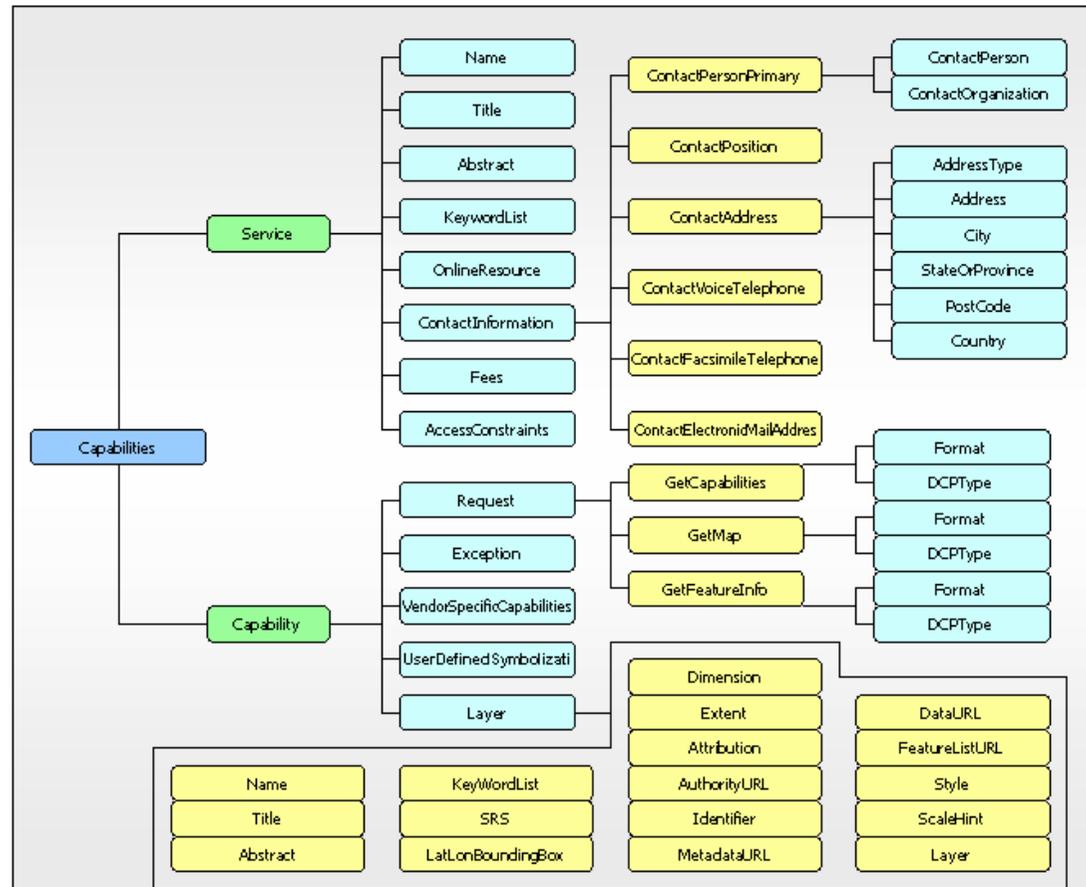
## Anexo N. Especificación de Requerimientos no Funcionales para la EAAB

<b>ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES</b>			
<b>Información General</b>			
<b>Fecha</b>	06/04/2004	<b>Usuario</b>	Ing. Johan Rivera
<b>Cargo</b>	Dirección de SIG		
<b>Descripción global de requerimientos no funcionales</b>			
Los requerimientos funcionales listados en este documento, están especificados con respecto al alcance del proyecto, para la comprobación de la efectividad de la guía metodológica planteada.			
<b>Nivel detallado de los requerimientos funcionales</b>			
<b>REQUERIMIENTO NO FUNCIONAL</b>			
<b>Identificador</b>	RNF_01	<b>Nombre</b>	Uso de una base de datos geográfica y relacional.
<b>Actores</b>	Usuarios del sistema	<b>Recursos</b>	Recursos de hardware y de software tales como un RDBMS.
<b>Restricciones</b>	La base de datos debe ser relacional.	<b>Prioridad (%)</b>	100
<b>Descripción General</b>			
Este requerimiento consigue garantizar el éxito de la generación de los servicios en línea, con respecto a la estructura, integridad y calidad de los datos que se almacenan en una base de datos geográfica, relacional en la cual se mantengan todas las entidades relacionadas de modo que el acceso a los datos sea eficiente y garantice la integridad de los mismos, en la medida en que cada capa se mantenga asociada con cada uno de los rasgos que la componen, así como la información adicional que se requiere para proveer los servicios WMS y WFS.			

<b>REQUERIMIENTO NO FUNCIONAL</b>			
<b>Identificador</b>	RNF_02	<b>Nombre</b>	Uso de estándares WMS y WFS para proveer servicios en línea geográficos
<b>Actores</b>	Usuarios del sistema	<b>Recursos</b>	Recursos de hardware y software requeridos para el sistema distribuido
<b>Restricciones</b>	El uso de estándares solo para las funciones básicas	<b>Prioridad</b>	100
<b>Descripción General</b>			
Se establece éste requerimiento, ya que la manipulación de información geográfica debe ser basada en procedimientos y formatos especializados para tal fin, por lo cual se plantea el uso de estándares internacionales para el manejo de información geográfica y servicios en línea desarrollados por el OpenGIS Consortium. Los estándares propuestos son WMS y WFS, los cuales proveen servicios para la manipulación de información geográfica en línea, dando la forma como se deben elaborar las solicitudes, la estructura de los procedimientos para cada una de las operaciones y la estructura y formato de las salidas que serán presentadas al cliente como respuesta al servicio que se ha solicitado.			
<b>REQUERIMIENTO NO FUNCIONAL</b>			
<b>Identificador</b>	RNF_03	<b>Nombre</b>	Uso de estándar GML para codificación e intercambio de información geográfica.
<b>Actores</b>	Usuario del sistema	<b>Recursos</b>	Recursos de hardware y software requeridos.
<b>Restricciones</b>	Se debe generar GML siempre que se acceda a información georeferenciada.	<b>Prioridad</b>	100
<b>Descripción General</b>			
La idea de utilizar GML para la manipulación e intercambio de información geográfica, es seguir una estandarización internacional creada para dicho fin. Teniendo en cuenta que se sale un poco del paradigma de obtener el mapa directamente, lo que se busca conseguir es obtener una codificación que permita manipular la imagen de manera más compleja con el objetivo de que la organización la utilice para análisis mas sofisticados de sus datos, a la vez que se pueda visualizar en el lado del cliente en un formato liviano y con propiedades que eviten falencias como distorsión de la imagen cuando se producen operaciones como cambios de tamaño.			

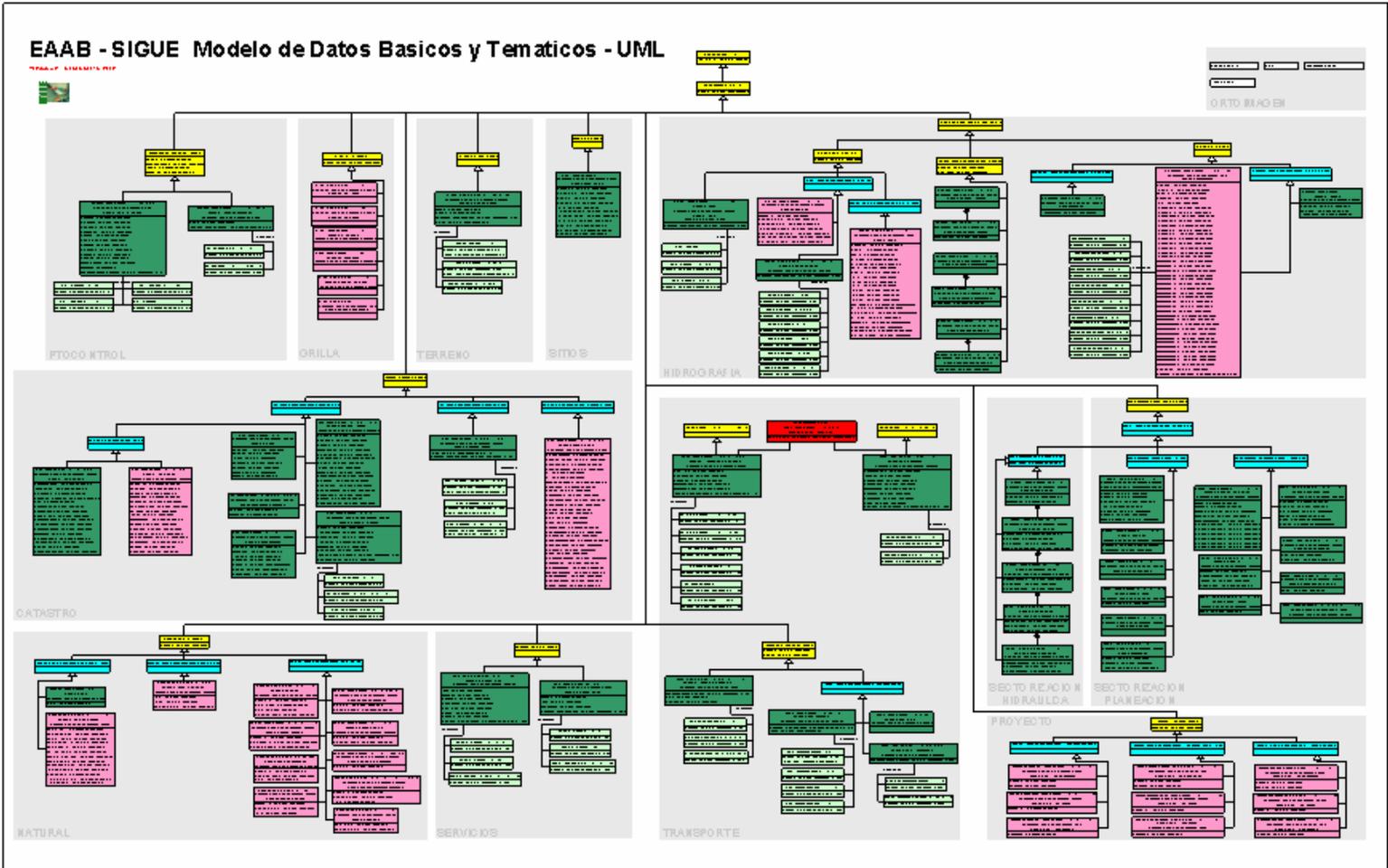
<b>REQUERIMIENTO NO FUNCIONAL</b>			
<b>Identificador</b>	RNF_04	<b>Nombre</b>	Eficiencia en la transmisión de los mapas al cliente.
<b>Actores</b>	Usuario del Sistema	<b>Recursos</b>	Recursos de hardware y software requeridos.
<b>Restricciones</b>	El cliente no debe recibir imagen estática	<b>Prioridad</b>	100
<b>Descripción General</b>			
<p>La eficiencia representada en tiempo y calidad cuando se provee un servicio a un cliente, se determinó debido a que se ha observado que la transmisión de imágenes por parte de los servidores a los clientes es un poco lenta, y más cuando se trata de mapas geográficos, ya que traen información de metadatos incluida, lo que hace más pesados los archivos. La eficiencia se requiere ya que el cliente lo que busca es rapidez, para poder hacer sus consultas y análisis en el menor tiempo posible, lo cual también hace que el cliente acceda con mayor frecuencia a los servicios en línea generados por la entidad. En la medida en que el cliente reciba mapas dinámicos, éste puede realizar análisis más complejos con la información de los mismos.</p>			
<b>REQUERIMIENTO NO FUNCIONAL</b>			
<b>Identificador</b>	RNF_05	<b>Nombre</b>	Seguridad en cuanto a permisos de acceso a las operaciones ofrecidas por WMS y WFS.
<b>Actores</b>	Usuario del Sistema	<b>Recursos</b>	Recursos de hardware y software requeridos.
<b>Restricciones</b>	Los usuarios deben estar suscritos a cada uno de los servidores.	<b>Prioridad</b>	100
<b>Descripción General</b>			
<p>La seguridad que se requiere se refiere a permisos de acceso a cada una de las operaciones de cada uno de los servidores WMS y WFS, y los permisos asociados para dichas operaciones tales como consulta o modificación de la información que se publica. La entidad tiene un manejo de usuarios y roles que se deben aprovechar para establecer los niveles de acceso a los diferentes niveles de la información con los privilegios asociados.</p>			
<b>Total de requerimientos</b>	5	<b>Fecha límite de entrega</b>	No aplica
<b>Aplicado por</b>	Bibiana Lara - Ana Maria Rodríguez	<b>Empresa</b>	Universidad Javeriana
<b>Cargo</b>	Grupo investigador Ing. Sistemas	<b>Firma</b>	<i>Bibiana Lara - Ana M<sup>a</sup>Rodríguez</i>

## Anexo O. Esquema de las Capacidades de los servidores WMS y WFS



Fuente: Adaptado del esquema de Capacidades de WMS y WFS

Anexo P. Modelos del aplicativo SIGUE de la EAAB



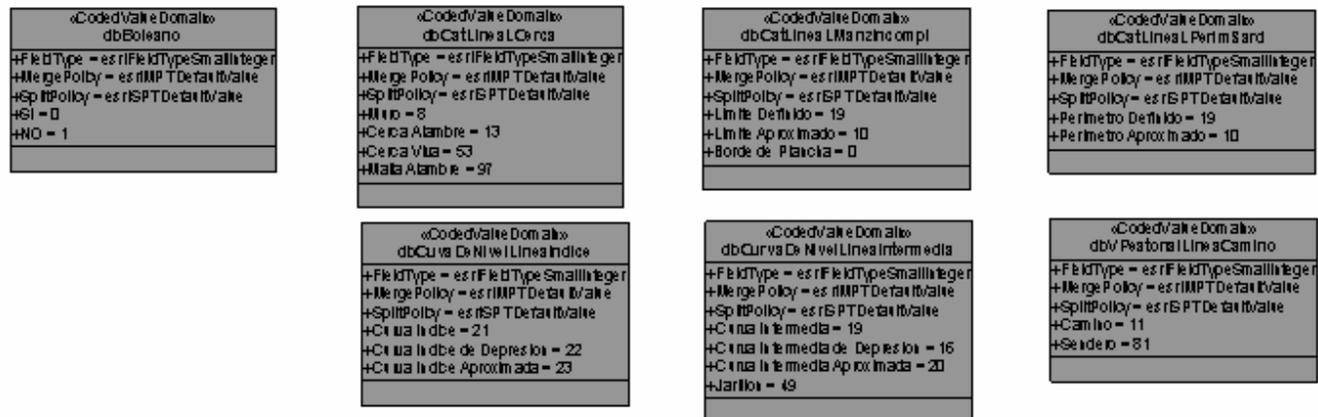
Modelo Temático - UML(Fuente: Proporcionado por la EAAB)

## EAAB - SIGUE Modelo de Datos Basicos y Tematicos - UML

Version 3.2 - Diciembre '16, 2012



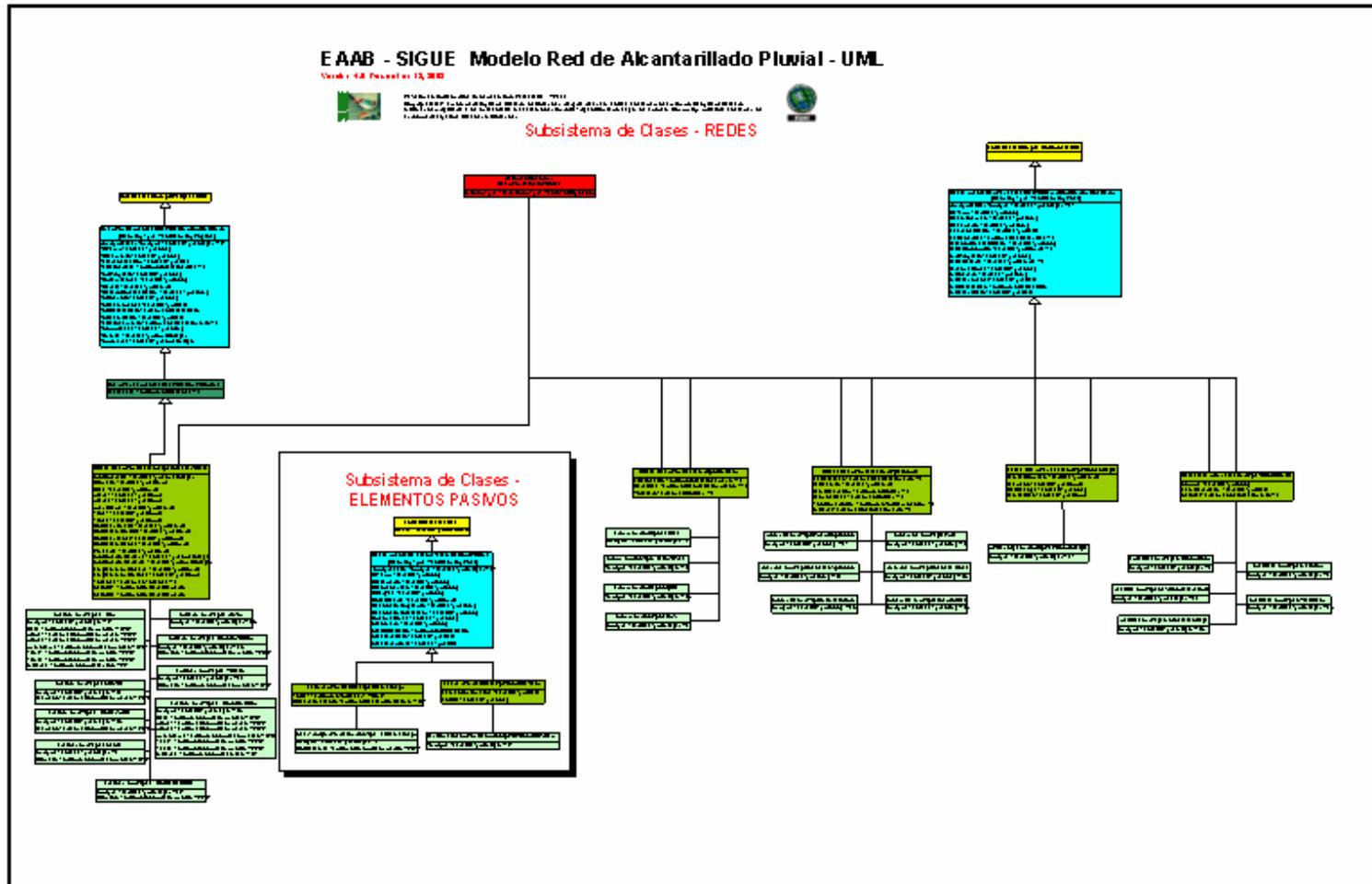
### DOMINIOS



Modelo Temático – Dominios (Fuente: Proporcionado por la EAAB)







Modelo Alcantarillado – UML (Fuente: Proporcionado por la EAAB)



# EAAB - SIGUE Modelo de Alcantarillado Pluvial - UML

Version 2.0 - December 12, 2002



## Diagrama de Dominios

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicClaseLinea
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-No. = <field type Small Integer> = 0
-Secundario = <field type Small Integer> = 1
-Local = <field type Small Integer> = 2

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicTipoLinea
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-Colector = <field type Small Integer> = 0
-Interceptor = <field type Small Integer> = 1

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicMaterialBancito
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-Concreto = <field type Small Integer> = 0
-Cra = <field type Small Integer> = 1
-Ladrillo = <field type Small Integer> = 2
-PVC = <field type Small Integer> = 3
-Herba = <field type Small Integer> = 4
-PiedraRegada = <field type Small Integer> = 5
-Diagonado = <field type Small Integer> = 6
-Otro = <field type Small Integer> = 7

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicMaterialBancito
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-Buena = <field type Small Integer> = 1
-Mala = <field type Small Integer> = 2
-Controlado = <field type Small Integer> = 3
-Diagonado = <field type Small Integer> = 4

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicMaterialBancito
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-No. = <field type Small Integer> = 0
-Si = <field type Small Integer> = 1

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicMaterialBancito
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-En Servicio = <field type Small Integer> = 0
-Fuera de Servicio = <field type Small Integer> = 1
-Abandonado = <field type Small Integer> = 2
-Diagonado = <field type Small Integer> = 3

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicMaterialBancito
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-No. = <field type Small Integer> = 00
-Bombas = <field type Small Integer> = 002
-Compuccion = <field type Small Integer> = 003
-Control de Garantia = <field type Small Integer> = 004
-Control de Seguro = <field type Small Integer> = 005
-Cuayacion = <field type Small Integer> = 006
-Inspeccion = <field type Small Integer> = 007
-Instalacion = <field type Small Integer> = 008
-Investigacion = <field type Small Integer> = 010
-Limpieza = <field type Small Integer> = 011
-Movimiento de tierra = <field type Small Integer> = 012
-Pruebas = <field type Small Integer> = 013
-Rehabilitacion material = <field type Small Integer> = 014
-Reparacion = <field type Small Integer> = 015
-Vivales = <field type Small Integer> = 016
-N/A. = <field type Small Integer> = N/A.

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicTipoCumanoStron
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-No Aplicar = <field type Small Integer> = 0
-Kulcado = <field type Small Integer> = 1
-Solado = <field type Small Integer> = 2

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicTipoCumanoStron
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-Cerros = <field type Small Integer> = 0
-Puro = <field type Small Integer> = 1
-No Aplicar = <field type Small Integer> = 2

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicCapacidadSumidero
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-100 = <field type Integer> = 100
-150 = <field type Integer> = 150
-200 = <field type Integer> = 200
-250 = <field type Integer> = 250
-N/A. = <field type Integer> = 0

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicTipoQuemil
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-Orden Ma Coactiva ALC = <field type String> = DM01
-Orden Ma Preventiva ALC = <field type String> = DM02
-Orden Aviso Ma ALC = <field type String> = DM03
-Orden Ma Pasivos ALC = <field type String> = DM04
-Orden Rehabilitacion Material ALC = <field type String> = DM05
-Otro = <field type String> = Otro
-N/A. = <field type String> = N/A.

```

```

<CodeOfValueDomain>
Domain APluvial_ClasapUcennicCalidadUlc
IF field type = numeric(field) = <field type Integer>
MergePolicy = numeric(field) = <mergePolicy default Value>
SplitPolicy = numeric(field) = <splitPolicy Duplicate>
-Buena = <field type Small Integer> = 1
-Regular = <field type Small Integer> = 2
-Mala = <field type Small Integer> = 3
-Diagonado = <field type Small Integer> = 4

```

Modelo Alcantarillado – Dominios (Fuente: Proporcionado por la EAAB)



## Anexo R. Descripción del contenido del CD

