



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería en Sistemas

**“CREACIÓN DE UNA INFRAESTRUCTURA DE DATOS ESPACIALES DE LA
PROVINCIA DEL AZUAY”**

Monografía previa a la obtención del título de

Ingeniero de Sistemas

Autores: Johnatan Fabricio Astudillo Llerena

Diego Sebastián Ñauta Tapia

Director: Ing. Diego Pacheco

Cuenca, Ecuador

2012

Dedicatoria

Dedicado a Nadia.

Johnatan Astudillo

Dedico este trabajo a mis padres Enrique y Zoila por haberme apoyado en todo momento, por sus consejos, sus valores que hicieron de mí una persona de bien y alcanzar mis ideales, pero más que nada por su infinito amor. A mis hermanos por guiarme y darme las fuerzas para seguir cumpliendo con todos mis sueños.

A todos ustedes se lo dedico por que sin su apoyo no hubiera podido concluir con una de mis más grandes metas que es alcanzar el título de Ingeniero de Sistemas para así poder desenvolverme ahora y en futuro como un gran profesional.

Sebastián Ñauta

Agradecimiento

De manera muy especial queremos agradecer al Ing. Diego Pacheco director de nuestra monografía, nos ha guiado evaluando y corrigiendo durante el desarrollo de este proyecto. A todos nuestros profesores quienes a lo largo de este tiempo nos han compartido sus conocimientos y experiencias que nos ayudarán en nuestra vida profesional.

Índice de contenidos

Dedicatoria.....	ii
Agradecimiento.....	iii
Índice de contenidos	iv
Índice de ilustraciones y cuadros	vii
Resumen	ix
Abstract.....	x
Introducción	11
Capítulo 1: Infraestructura de datos espaciales (IDE)	12
1.1 Introducción	12
1.2 Conceptos y objetivos de una IDE.....	12
1.3 Principios	13
1.3.1 Marco institucional	13
1.3.2 Estándares	13
1.3.3 Tecnología	13
1.3.4 Política de datos	13
1.4 Componentes	13
1.4.1 Datos	14
1.4.2 Metadatos.....	14
1.4.3 Servicios web.....	14
1.4.4 Estructura organizativa	16
Capítulo 2: Software utilizado para el montaje de una IDE	19
2.1 Introducción	19
2.2 Fundamentos del software libre	19
2.3 Software del lado del servidor	21
2.3.1 Servidores de base de datos espaciales	21

2.3.2	Servidor de mapas.....	23
2.3.3	Herramientas de metadatos	24
2.4	Software del lado del cliente.....	26
2.4.1	Clientes pesados.....	26
2.4.2	Clientes ligeros web.....	28
2.5	Conclusiones.....	34
Capítulo 3: Creación de la infraestructura de datos espaciales de la provincia de Azuay		35
3.1	Introducción.....	35
3.2	Publicación de la geoinformación como servicio <i>wms</i> en <i>mapserver</i>	35
3.2.1	Datos	35
3.2.2	Estándar <i>ogc</i> del servicio <i>wms</i>	37
3.2.1.1	Operaciones web <i>mapservice</i>	38
3.2.1.1.1	<i>GetCapabilities</i>	38
3.2.1.1.2	<i>GetMap</i>	39
3.2.1.1.2	<i>GetFeatureInfo</i>	41
3.2.3	Publicación en <i>mapserver</i>	43
3.2.3.1	Creación de la base de datos en <i>postgreSQL</i>	43
3.2.3.2	Exportar capas.....	44
3.2.3.3	Crear y definir el <i>mapfile</i> requerido por <i>mapserver</i>	47
3.3	Visualización del servicio <i>wms</i> a través del cliente ligero <i>p.mapper</i>	47
3.3.1	Instalación y configuración de requisitos.....	47
3.3.2	Instalación y configuración de la aplicación.....	48
3.3.2.1	Instalación.....	48
3.3.2.2	Adaptación del <i>mapfile</i>	49
3.3.2.3	Configuraciones de la aplicación	50
3.3.3	Personalización de la aplicación	52
3.3.4	Funcionalidades y herramientas de <i>p.mapper</i>	53

3.3.4.1 Funcionalidades	53
3.3.4.2 Herramientas	55
3.4 Publicación de metadatos con <i>geonetwork</i>	56
3.4.1 Configuraciones <i>geonetwork</i>	56
3.4.2 Personalizar <i>geonetwork</i>	59
3.4.3 Creación de nuevos grupos de usuarios	59
3.4.4 Creación de nuevos usuarios	59
3.4.5 Insertar plantilla <i>xml</i>	59
3.4.6 Construcción de los metadatos	60
3.4.6.1 Datos	60
3.4.6.2 Creación de metadatos	61
3.4.6.3 Creación de una vista en mapa miniatura en los metadatos	61
CONCLUSIONES	62
RECOMENDACIONES	64
BIBLIOGRAFÍA	65
Anexo 1: Instalación y configuración de <i>mapserver</i> en el sistema operativo <i>linux centos 5.x</i>	68
Anexo 2: Exportación de las capas con <i>gvSIG</i>	78
Anexo 3: Publicación de las capas con <i>gvSIG</i>	80
Anexo 4: Configuración del <i>mapfile</i> atlas_azuay.map	83
Anexo 5: Configuración del archivo .xml de <i>p.mapper</i>	86
Anexo 6: Instalación y configuración de <i>geonetwork</i> en el sistema operativo <i>linux centos 5.x</i>	91
Anexo 7: Personalización de <i>geonetwork</i> configurando el archivo <i>config-gui.xml</i>	94

Índice de ilustraciones y cuadros

Tabla 1. Características del cliente ligero web para SIG <i>OpenLayers</i>	31
Tabla 2. Características del cliente ligero web para SIG <i>MsCross</i>	32
Tabla 3. Características del cliente ligero web para SIG <i>MapBender</i>	33
Tabla 4. Características del cliente ligero web para SIG <i>p.mapper</i>	34
Tabla 5. Operaciones <i>WMS</i>	37
Tabla 6. Parámetros de la Operación <i>GetCapabilities</i>	38
Tabla 7. Parámetros de la Operación <i>GetMap</i>	40
Tabla 8. Parámetros de la Operación <i>GetFeatureInfo</i>	42
Tabla 9. Reporte de las capas tipo <i>shape</i> importadas en la base de datos.....	46
Tabla 10. Reporte de las capas tipo <i>raster</i>	47
Figura 1. Geoservicios.	16
Figura 2. Jerarquía entre diferentes IDEs. Fuente: http://www.ideo.es	17
Figura 3. Dependencias entre clientes web para SIG de código abierto. Fuente: GeoTux.....	29
Figura 4. Grupos de Datos Geográficos de la IDE del Azuay.	36
Figura 5. Niveles de Datos Geográficos de la IDE del Azuay.....	36
Figura 6. Archivos principales de configuración.....	49
Figura 7. Archivos principales de configuración.....	49
Figura 8. Archivos principales de configuración.....	49
Figura 9. Sección <i><pmapper></i> del archivo de configuración <i>config_azuguay.xml</i>	50
Figura 10. Sección <i><config></i> del archivo de configuración <i>config_azuguay.xml</i>	50
Figura 11. Sección <i><map></i> del archivo de configuración <i>config_azuguay.xml</i>	51
Figura 12. Sección <i><category></i> del archivo de configuración <i>config_azuguay.xml</i>	51
Figura 13. Resultado de la configuración del archivo <i>config_azuguay.xml</i>	52
Figura 14. Funcionalidad de búsqueda.	53

Figura 15. Funcionalidad de escala de presentación.....	53
Figura 16. Funcionalidad de escala gráfica.....	54
Figura 17. Funcionalidad de coordenadas.	54
Figura 18. Funcionalidad de <i>zoom</i>	54
Figura 19. Referencia de mapa.	54
Figura 20. Barra de herramientas.....	55
Figura 21. Página de administración de <i>Geonetwork</i>	56
Figura 22. El correo de las opciones de configuración del servidor	58
Figura 23. Opciones de configuración de autenticación	58
Figura 24. Importación del archivo XML.....	60
Figura 25. Vista en mapa miniatura en geonetwork	61
Figura 26. Archivo de configuración <i>pg_hba.conf</i>	70
Figura 27. Archivo de configuración <i>postgresql.conf</i>	71
Figura 28. Archivo de configuración <i>ld.so.conf</i>	76
Figura 29. Añadir capas	78
Figura 30. Añadir conexión a base de datos.	78
Figura 31. Exportación de capas.....	79
Figura 32. Añadir capas desde la base de datos.....	80
Figura 33. Herramienta de publicación.....	81
Figura 34. Herramienta de publicación.....	81
Figura 35. Pestaña Servidor	82
Figura 36. Pestaña Servicio	82
Figura 37. Pantalla inicial de Tomcat.	93
Figura 38. Pantalla inicial de Geonetwork.....	93

Resumen

El presente trabajo tiene como objetivo construir una Infraestructura de Datos Espaciales de la provincia del Azuay que esté acorde a las leyes y estándares nacionales e internacionales de interoperabilidad. La información cartográfica publicada en esta investigación, fue entregada por el Instituto de Estudios de Régimen Seccional del Ecuador (*IERSE*) y posteriormente preparada para la correcta integración. En el proceso se utilizó *Mapserver* como servidor de mapas, *postgreSQL* con extensión *postgis* para el almacenamiento de la geoinformación, *p.mapper* como cliente ligero web para el acceso, visualización y administración del servicio *Web Map Service (WMS)* y *Geonetwork* para la gestión de los metadatos. En el documento también se incluye los procedimientos de instalación y configuración de las herramientas requeridas para la creación de la IDE.

Abstract

The goal of the present research is to create a Spatial Data Infrastructure of the province of Azuay according to national and international laws and standards. The cartographic information published in this research was provided by the Institute of Sectional Studies of Ecuador (IERSE), which was then organized for its correct integration to the system. *Mapserver* was used as a server, *postgreSQL* with a *postgis* extension was used for storing geographical information, *p.mapper* as a Web thin client in order to access, visualize and administer *Web Map Service (WMS)*, and *Geonetwork* for the management of metadata. The project also includes the procedures followed for the installation and configuration of the tools required for the creation of the SDI.



Diana Lee Rodas
Translated by,
Diana Lee Rodas

Introducción

Las Infraestructuras de Datos Espaciales (IDE) se fundamentan como una forma de presentar, compartir, tratar y analizar los datos geográficos de manera remota; los avances en el Internet han permitido que esta tecnología obtenga una gran acogida en todo el sector interesado en la geoinformación, haciendo posible el progreso tecnológico, social y económico del lugar donde se lo implemente. En apoyo al desarrollo local, esta investigación expone la creación de una infraestructura de datos espaciales para la provincia del Azuay.

El documento está compuesto por tres capítulos, tanto teóricos como prácticos, que proponen una guía para crear una IDE con una estructura organizativa de nivel provincial, basada en leyes y estándares nacionales e internacionales.

En el primer capítulo brevemente se explican, con propósitos de introducción a las IDEs, los conceptos, objetivos y componentes más indispensables por considerar al momento en que se propone construir este tipo de infraestructura.

En el segundo capítulo se aborda los temas referentes al *software* libre necesario, tanto del lado del servidor como del lado del cliente: servidores de mapas, servidores de bases de datos espaciales, herramientas de metadatos, clientes ligeros web y clientes pesados; además, se presentan las características técnicas de las herramientas de mayor uso en la actualidad.

En el capítulo tres se identifica el software a utilizarse para la creación, que para este estudio se ha seleccionado los mismo elementos software que posee el servidor de la Universidad del Azuay, en el cual será implementado la IDE; conjuntamente se plantea la distribución por categorías de los mapas a crearse a partir de las información cartográfica provista por el IERSE (Instituto de Estudios de Régimen Seccional del Ecuador); de la misma manera se revisa el estándar internacional *Open Geospatial Consortium (OGC)* para la creación del servicio *Web Map Service (WMS)*; seguidamente se explica la forma de crear los metadatos con la herramienta *Geonetwork* y finalmente los procedimientos, instalaciones y configuraciones requeridos para la creación.

Capítulo 1: Infraestructura de datos espaciales (IDE)

1.1 Introducción

En la actualidad las IDEs son verdaderas infraestructuras de información para la búsqueda, evaluación y explotación de la información geoespacial. Estas infraestructuras han llegado al punto de flexibilidad que ahora están disponibles para usuarios y proveedores de todos los niveles de la Administración, sectores empresarial, académico, y ciudadanos en general. En el primer capítulo se dará a conocer brevemente los conceptos, principios y componentes principales de una IDE.

1.2 Conceptos y objetivos de una IDE

Se puede definir una IDE como un sistema informático integrado por un conjunto de recursos (catálogos, servidores, programas, datos, aplicaciones, páginas web, etc.) dedicados a gestionar información geográfica (mapas, ortofotos, imágenes de satélite, topónimos, etc.), disponibles en Internet, que cumplen una serie de condiciones de interoperabilidad (normas, especificaciones, protocolos, interfaces, etc.) que permiten que un usuario, utilizando un simple navegador, pueda utilizarlos y combinarlos según sus necesidades. Una IDE puede implementarse en una empresa, un centro de investigación, un organismo oficial, como ayuda para la gestión de su propia información espacial, y también puede implantarse como servicio público con la posibilidad de combinarse con otras IDE.

Los objetivos de una IDE son: facilitar el acceso y la integración de la información espacial, tanto a nivel institucional y empresarial como de los propios ciudadanos, lo que permitirá extender el conocimiento y el uso de la información geográfica y la optimización de la toma de decisiones; promover los metadatos estandarizados como método para documentar la información espacial, lo que permitirá la reducción de costos y evitar la duplicación de esfuerzos; y animar a la cooperación entre los agentes, favoreciendo un clima de confianza para el intercambio de datos.

1.3 Principios

Para la creación de una IDE se debe tener en consideración los siguientes principios:

1.3.1 Marco institucional

Entre los productores de información geográfica se deben establecer acuerdos, para generar y mantener los datos espaciales fundamentales para la mayoría de las aplicaciones basadas en sistemas de información geográfica, y así garantizar la calidad de la geoinformación publicada.

1.3.2 Estándares

Uno de los objetivos de una IDE es el intercambio e interoperación con sistemas similares y por ello es necesario el establecimiento de normas, a las que deberá ajustarse la información y servicios geográficos.

1.3.3 Tecnología

El establecimiento de la red y mecanismos informáticos que permitan: buscar, consultar, encontrar, acceder, suministrar y usar los datos espaciales o geográficos.

1.3.4 Política de datos

El establecimiento de las políticas, alianzas y acuerdos de colaboración necesarios para aumentar la disponibilidad de datos espaciales y compartir los desarrollos tecnológicos.

1.4 Componentes

Los componentes necesarios para la construcción de una IDE son los siguientes:

1.4.1 Datos

Los datos son el conjunto de información de una determinada temática que tiene la propiedad de ser georeferenciable. Los datos pueden ser a su vez datos de referencia o datos temáticos. Los datos de referencia son los que forman el Mapa Base o mapa sobre el que se referencian los datos temáticos. Forman parte del Mapa Base el sistema de coordenadas, las redes de transporte, la red hidrológica, la altimetría, los límites administrativos, las construcciones, etc. Los datos temáticos son los valores de las distintas capas de información geográfica que no forman el Mapa Base y que superponen a ella. Serían datos temáticos capas relativas a clima, edafología, hidrología, vegetación, población, etc.

1.4.2 Metadatos

Los metadatos describen los datos y se utilizan para tomar decisiones acerca de los mismos. Los metadatos constituyen la información, en forma de documentación que permite que los datos sean bien entendidos, compartidos y explotados de manera eficaz por todo tipo de usuarios a lo largo del tiempo.

Los metadatos de la información geográfica informan a los usuarios sobre los datos existentes describiendo: el sistema de referencia espacial, la calidad, su distribución, el formato, restricciones de seguridad, frecuencia de actualización, etc., de tal forma que sirven para describir un conjunto de datos geográficos.

1.4.3 Servicios web

Los servicios son las funcionalidades accesibles mediante un navegador de Internet que una IDE ofrece al usuario para aplicar sobre los datos geográficos. Estas funcionalidades se organizan en diferentes servicios: servicios de visualización de mapas, de descarga, de consulta, etc.

En la actualidad los servicios más comunes que se ofrecen son:

a) El Servicio de mapas en la web (WMS)

Permite la visualización de una imagen cartográfica generada a partir de una o varias fuentes: mapa digital, datos de un SIG, ortofoto, etc., provenientes de uno o varios servidores.

b) El Servicio de fenómenos en la web (WFS)

Permite acceder a los datos mismos, mediante el empleo del formato GML. Así se puede acceder al archivo que define la geometría de un objeto cartográfico, como un río, una ciudad, una parcela, etc., y disponer de esa información vectorial en el propio ordenador.

c) El Servicio de coberturas en web (WCS)

Es un servicio similar al WFS pero para datos raster, como es el caso de las imágenes digitales tomadas desde satélite o mediante una cámara fotogramétrica digital, y los modelos digitales del terreno.

d) Servicio de Nomenclátor (Gazetteer)

Permite localizar un fenómeno geográfico de un determinado nombre. Este servicio admite como entrada el nombre del fenómeno y nos muestra la localización a través de las coordenadas del mismo. Además, la consulta por nombre permite fijar otros criterios como la extensión espacial en la que se necesita buscar el tipo de fenómeno dentro de una lista disponible. En el caso de que existan varios que cumplan con la condición de búsqueda, el servicio ofrece una lista de los nombres encontrados con algún atributo adicional para que el usuario pueda escoger el que necesita.

e) Servicio de Geoparser

Analiza palabra a palabra un texto digital determinado, efectuando comparaciones con un conjunto de nombres geográficos dado y crea los vínculos o enlaces necesarios para que exista una referencia permanente en el texto original a los fenómenos

geográficos indicados. En definitiva, transforma el texto original en un hipertexto con vínculos geográficos y para ello ha de utilizar un Servicio Nomenclátor.

f) Servicio de catálogo (CSW)

Permite la publicación y búsqueda de información (metadatos) y es necesario para proporcionar capacidades de búsqueda y solicitud sobre los recursos registrados dentro de una IDE

En el siguiente gráfico muestra el comportamiento de una solicitud a los geoservicios.

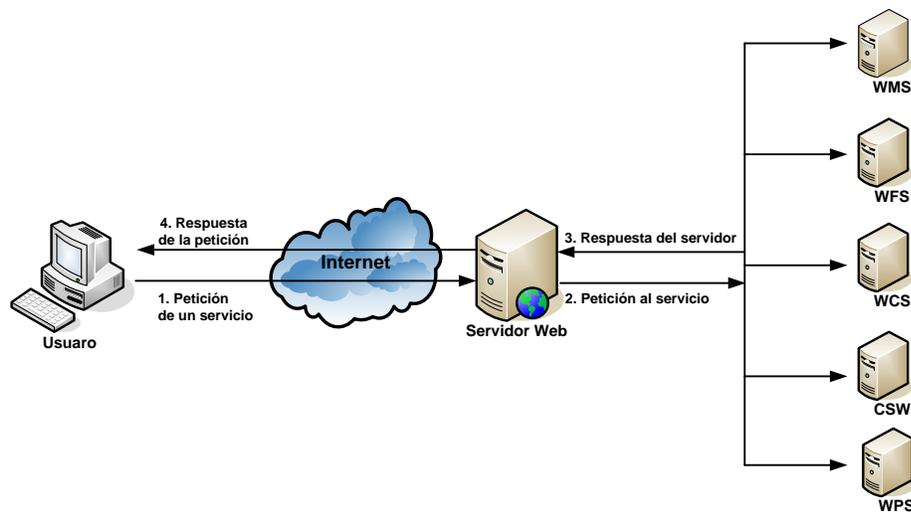


Figura 1. Geoservicios.

1.4.4 Estructura organizativa

Esta incluye una componente política para toma de decisiones, una organizativa del reparto de trabajo y un conjunto de estándares y normas que hacen que los sistemas IDE puedan interoperar. Los marcos en los que se desarrolla esta componente suelen venir definidos mediante legislaciones específicas.

Todos los componentes son necesarios, pero la organización es de especial importancia en una IDE porque ordena, regula, estructura y armoniza todos los demás. De ahí, también, su complejidad, pues debe adaptarse al tipo de organización política y social de cada Estado.

Las Infraestructuras de Datos Espaciales deben construirse para fundamentar a otras que a su vez pueden ser la base para menores. Las IDE deben desarrollarse armónicamente de manera que cada una de ellas garantice la sustentabilidad de las que se apoyan en ella. Las infraestructuras ubicadas en niveles superiores se componen de datos espaciales de niveles inferiores. Los niveles superiores, abarcan todos los componentes de los niveles inferiores; a su vez cualquier nivel sirve como punto de partida para niveles superiores. Esta jerarquía hace posible que la toma de decisiones en cierto nivel, sea más específica, puesto que cualquier nivel recurre a los datos de los otros niveles. Estos niveles estructurales de las IDE se representan de manera piramidal en la siguiente figura.

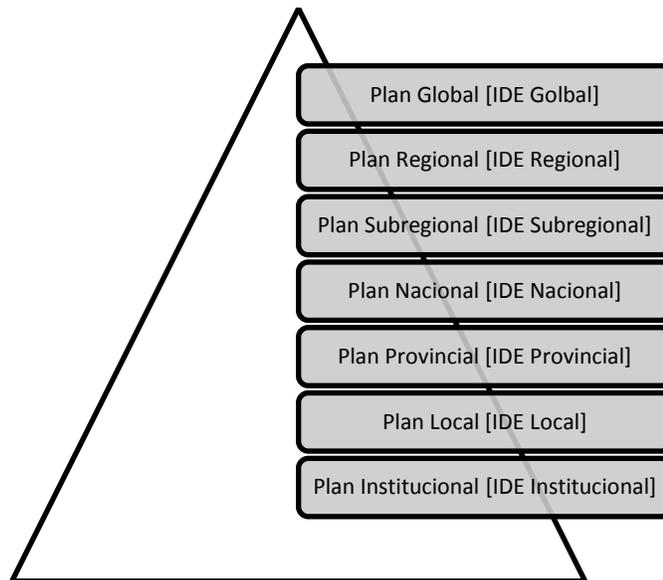


Figura 2. Jerarquía entre diferentes IDEs. Fuente: <http://www.idee.es>

También entra dentro del ámbito de los aspectos organizativos el establecimiento y adopción de estándares y acuerdos, imprescindible para la coherencia, compatibilidad e interoperabilidad necesarias para que los datos, servicios y recursos de una IDE puedan ser utilizados, combinados y compartidos

Una de las entidades productoras de servicios para IDEs es el *Open Geospatial Consortium (OGC)*. Se trata de un consorcio internacional de empresas, instituciones públicas y universidades que trabajan para el desarrollo de especificaciones estándares

para el procesamiento de la información geográfica. Los Servicios *OGC* pueden ser encadenados y combinados en un Geoportal, ofreciendo la posibilidad de: buscar un fenómeno por nombre (Nomenclátor) y visualizar el resultado sobre unos datos de referencia (*WMS*); localizar un producto seleccionando algunas características (Catálogo) y visualizarlo en pantalla (*WMS o WCS*).

Las especificaciones más importantes surgidas del *OGC* son:

- *GML (Geographic Markup Language)*.
- *WFS (Web Feature Service)*.
- *WMS (Web Map Service)*.
- *WCS (Web Coverage Service)*.
- *CS-W (Catalog Service Web)*.

Capítulo 2: Software utilizado para el montaje de una IDE

2.1 Introducción

En abril del 2008 el gobierno ecuatoriano emitió el decreto 1014 que establece como política para las Entidades de la Administración Pública la utilización de Software Libre en sus sistemas y equipamientos. Para estar en acorde con la tendencia que sigue la Administración Pública; este capítulo enfocado a las herramientas de software libre disponibles y más populares en la actualidad para la creación de una IDE, tanto del lado del servidor como del lado del cliente.

2.2 Fundamentos del software libre

Desde hace 40 años, el *software* era un complemento que los vendedores de los ordenadores (*hardware*) aportaban a sus clientes para que pudieran usarlos. Era común que los programadores compartieran sus programas. Pero desde principios de los años setenta nos hemos acostumbrado a que quien comercializa un programa pueda imponer las condiciones bajo las que puede usarse. Debido a este inconveniente surge el término software libre (o programas libres) tal como fue concebido por *Richard Matthew Stallman* quien fue el fundador de la *Free Software Foundation (FSF)*, hace referencia a las libertades que puede ejercer quien lo recibe, significa que los usuarios de programas, tienen las siguientes libertades esenciales:

- a) Libertad para ejecutar el programa en cualquier sitio, con cualquier propósito y para siempre.
- b) Libertad para estudiarlo y adaptarlo a nuestras necesidades. Esto exige el acceso al código fuente.
- c) Libertad de redistribución, de modo que se nos permita colaborar con vecinos y amigos.
- d) Libertad para mejorar el programa y publicar sus mejoras. Esto también exige el código fuente.

Estas libertades hacen referencia a un tipo de programas informáticos que se distribuye con las libertades necesarias para que los usuarios puedan mejorarlo, distribuirlo y utilizarlo.

Algunas de las ventajas al implementar con software libre son:

- Inversión mínima en compra de software y desarrollo del proyecto
- Redistribución y libertad de uso del software
- Administración de estándares reconocidos mundialmente como la *OGC*.

Sin embargo el Software Libre no tiene por qué ser necesariamente gratuito, de hecho en muchos casos se distribuye comercialmente y se realizan negocios con su mantenimiento y desarrollo. La diferencia con respecto al Software propietario es que el cliente paga por un producto y recibe no sólo un programa ejecutable, sino también su código fuente y el derecho a distribuirlo o mejorarlo si el usuario lo cree conveniente.

Hoy en día los proyectos y productos conocidos como *Free and Open Source (FOSS)*, que están en relación con la geomática están evolucionando de una forma rápida. De esta manera, además de los conocidos proyectos de clientes pesados, servicios de mapas o bases de datos espaciales, están creciendo otra serie de elementos asociados como servicios de publicación, clientes ligeros, servicios de geoprocésamiento o estándares como *WMS*, *WPS*, entre otros.

En la geomática se encuentran diferentes proyectos relacionados, que se agrupan en diversas temáticas: proyectos del lado del servidor y del lado del cliente.

2.3 Software del lado del servidor

2.3.1 Servidores de base de datos espaciales

Hoy en día estos proyectos son muy importantes, debido a la disposición de la tecnología que hace trabajar con una mayor cantidad de información y por ello lo más adecuado es utilizar sistemas gestores de bases de datos geográficos relacionales.

El *OGC* ha dispuesto una especificación del conjunto de tipos de datos y funciones que una base de datos geográfica debe cumplir, conocida como *Simple Features*. Esta norma regula las llamadas a funciones en *SQL (Structured Query Language)* para poder realizar aplicaciones independientes del Sistema de Gestión de Base de Datos Relacional (SGBDR).

Los servidores de bases de datos espaciales libres más importantes en la actualidad son los siguientes:

PostGIS

PostGIS es un producto muy utilizado en la actualidad ya que es un sistema de código abierto de Sistema de Información Geográfico, que proporciona soporte para el manejo de datos geográficos en la base de datos relacional de *PostgreSQL*. *PostGIS* implementa las características esenciales (*Simple features*) para la especificación *SQL* del *Open Geospatial Consortium*.

PostGIS habilita a *PostgreSQL* para el manejo de datos espaciales, permitiéndole ser usado como un respaldo de base de datos espacial para los Sistemas de Información Geográfica (SIG).

PostGIS implementa las siguientes características:

- Operadores espaciales para determinar las medidas geoespaciales como el área, distancia, longitud y perímetro.
- Operadores espaciales para determinar el conjunto de operaciones geoespaciales, como la unión, diferencia, simétrica y buffer.
- Índices espaciales de tipo árbol R para rápidas consultas espaciales.
- Soporte para índices selectivos, para proporcionar alto desempeño en la planeación de consultas para consultas combinadas espaciales y noespaciales.

MySQL

Se trata de un programa de licencia *OpenSource* y gratuito pero que, sin embargo, está mantenido por una empresa, *MySQL AB. MySQL*; es una de las bases de datos más conocidas en aplicaciones web, aunque no se puede considerar completamente *FOSS* ya que no es un producto totalmente libre, puesto que existe una versión comercial. Un inconveniente que presenta es que no cumple la norma *Simple Features Specification for SQL (SFSS)* que es un Estándar *OGC* de funciones de procesamiento y consulta de bases de datos geoespaciales; y por tanto actúa solo como un “contenedor” de información geográfica.

MySQL implementan las siguientes características:

- Posibilidad de crear y configurar usuarios, asignando a cada uno de ellos permisos diferentes.
- Facilidad de exportación e importación de datos, incluso de la base de datos completa.
- Posibilidad de ejecutar conjuntos de instrucciones guardadas en ficheros externos a la base de datos.

2.3.2 Servidor de mapas

Un servidor de mapas es un Sistema de *Hardware* y *Software* capaz de enviar vía *Web*, mapas digitales de forma dinámica, de acuerdo a las consultas realizadas por el cliente a través del navegador de internet o de un software SIG que soporte el protocolo *WMS*.

El *OGC* ha promovido la estandarización de servicios de interoperabilidad, que ha posibilitado desde el inicio de los proyectos *FOSS*, la puesta en marcha de aplicaciones relacionadas con la publicación de cartografía a través de la *web*.

Los servidores de mapas más importantes y de libre uso en la actualidad son los que se exponen a continuación.

UMN MapServer

Es un entorno de desarrollo en código abierto que permite utilizar datos del SIG desde un explorador *web* para poder visualizar, consultar y analizar información geográfica. *MapServer* funciona directamente con *shapefiles* y bases de datos, es totalmente autosuficiente; no necesita de otro programa servidor para procesar datos o crear informes.

Entre las características de *MapServer* destacan las siguientes:

- Dibujo y etiquetado dependiente de la escala.
- Formatos raster soportados: *JPG, PNG, GI, TIFF*
- Valores de escala.
- Símbolo y color adaptables.
- Acceso en función de las características a datos sobre atributos.
- Generación automática de leyendas.
- Utilización de datos en forma de mosaico.

GeoServer

Es un servidor de código abierto que le permite conectar su información a la *Web Geoespacial*, con *GeoServer* se puede publicar y editar los datos mediante estándares abiertos; su información está disponible en una variedad de formatos como mapas o imágenes reales o de datos geoespaciales.

Entre las principales características de *GeoServer* se destacan las siguientes:

- Enteramente compatible con las especificaciones *WMS*, *WCS* e *WFS*.
- Fácil utilización mediante la herramienta de administración web.
- Soporte amplio de formatos de entrada *PostGIS*, *Shapefile*, *ArcSDE* y *Oracle*.
- Soporte de formatos de salidas como *JPEG*, *GIF*, *PNG*, *SVG* y *GML*.
- Diseñado para ser compatible con extensiones.
- Incluye un cliente integrado *OpenLayers* para pre visualizar capas de datos.

2.3.3 Herramientas de metadatos

Un Servidor de Catálogo es una aplicación que facilita la publicación en Internet de un conjunto de metadatos acerca de distintos conjuntos de datos. En el ámbito de la geomática, dichos datos constituyen distintos tipos de información geográfica, esto es, temas raster y vectoriales, fotografías aéreas, mapas digitalizados, etc. Dicho catálogo se muestra como un portal que facilita las búsquedas mediante distintos criterios (espaciales y/o alfanuméricos).

Entre los más importantes, se encuentran los siguientes:

Geonetwork

Es un sistema basado en estándares, *software* libre y código abierto para la administración de catálogos de datos geoespaciales por *Web*. *Geonetwork* integra creación y administración de Metadatos, publicación y distribución de datos,

intercambiar y compartir información con otras organizaciones y búsqueda de metadatos.

Entre las principales características de *Geonetwork* se destacan las siguientes:

- Plantillas de metadatos.
- Descarga de datos espaciales.
- Almacena los registros de metadatos (archivos *XML*) como un atributo en una tabla de la base de datos y otra información como: la fuente, el propietario, los índices, etc.
- Visualización dinámica online.
- Servicios de mapas dinámicos.
- Autenticación de usuarios en metadatos y servicios de administración de metadatos.
- Módulo de manejo para datos y metadatos.

CatMDEdit

Es un software de código abierto, multiplataforma y multilenguaje que permite la gestión de metadatos de información geográfica, de acuerdo a los estándares de la norma ISO 19115. Aporta herramientas que facilitan la generación, manipulación y publicación de registros de metadatos.

Características principales de *CatMDEdit*:

- Edición de metadatos de acuerdo a la norma internacional “ISO19115. *Geographic Information-Metadata*”
- Diferentes estilos de presentación de registros de metadatos en *HTML* y *Excel*.
- Generación automática de metadatos para algunos formatos de transferencia de datos como *Shapefile*, *DGN*, *ECW*, *FICC*, *GeoTiff*, *GIF/GFW*, *JPG/JGW* o *PNG/PGW*

- Multiplataforma (Windows, Unix), gracias al uso de Java como lenguaje de programación

2.4 Software del lado del cliente

2.4.1 Clientes pesados

Las aplicaciones de escritorio han sido las más conocidas entre las herramientas para la gestión de la información geográfica. Estas aplicaciones facilitan la manipulación de información geográfica, así como su edición, análisis y explotación. A continuación se exponen algunos tipos *FOSS* para este tipo de herramientas:

QuantumGIS

Esta herramienta permite que usuarios con necesidades básicas puedan trabajar con un SIG en un entorno “amigable”, permitiéndoles el análisis vectorial y raster, además de las opciones comunes de visualización.

Las principales características son:

- Transforma el sistema de coordenadas de forma inmediata.
- Edición y selección de elementos geoespaciales.
- Edición, visualización y búsqueda de atributos.
- Herramienta de etiquetado.
- Herramientas de análisis espacial.
- Publicación a servidor *web* de mapas.

GvSIG

Es una herramienta orientada al manejo de la información geográfica, permite acceder a información vectorial o raster y que integra en una vista datos tanto locales como remotos a través de un origen *WMS*, *WCS* o *WFS*.

Este software está dirigido a usuarios finales de información geográfica de distintos ámbitos, sean de ámbitos privados o de administraciones públicas, ofreciéndose de manera gratuita en diferentes idiomas.

Como características más relevantes, se destacan las siguientes:

- Funciona en distintas plataformas hardware / software.
- Es de código abierto.
- No necesita de licencias; de esta manera, una vez finalizado el desarrollo no hay que abonar ningún coste por cada instalación que se realice.
- Sujeto a los estándares marcados por la OGC.
- Capaz de acceder a los datos de otro software propietario, como ArcView, AutoCAD o Microstation, sin necesidad de cambiarlos de formato.

GvSIG cuenta con las siguientes funcionalidades:

- Lectura de formatos vectoriales: *shapefiles*, *dxf*, *dwg (2000)*, *dgn (v7)*, *PostGIS*, *MySQL*, *WFS*, *ArcIMS* vectorial y *Oracle* vectorial.
- Lectura de formatos raster: *WMS*, *WCS*, *ECW*, *MrSID*, *geoTIFF*, *ArcIMS*, *IMG(Erdas)*, formatos *RAW*, etc.
- Geoprocesos vectoriales: los más comunes son unión, intersección, o área de influencia, entre otros.
- Geoprocesos raster: destaca el análisis de MDE, hidrología o geoestadística.
- Maquetación de mapas.
- Edición avanzada de cartografía.
- Gestión avanzada de sistemas de coordenadas y sistemas de referencia.
- Versión para dispositivos móviles.
- GIS 3D.
- Topología y gestión de redes.

2.4.2 Clientes ligeros web

Los clientes web para SIG son aplicaciones de internet que se encargan de visualizar información geográfica y permiten su manipulación a través de herramientas básicas de navegación y análisis. Existen varios proyectos de software libre y de código abierto que facilitan la administración, el desarrollo y la personalización de este tipo de aplicaciones, las cuales consumen servicios web y comunican al usuario con tareas avanzadas que se realizan en el servidor.

El *Open Geospatial Consortium (OGC)* ha promovido el uso de estándares para servicios web de mapas que han ayudado a establecer un marco común de trabajo para acceder a información geográfica en la internet (*Web Map Service, Web Feature Service, Web Coverage Service*), presentarla por medio de estilos (*Style Layer Descriptor*), filtrarla (*Filter encoding*), almacenarla, transportarla (*Geography Markup Lenguaje y Keyhole Markup Language*) y procesarla (*Web Processing Service*).

El siguiente gráfico ilustra las dependencias entre los clientes web OpenSource más conocidos en la aplicación de geoservicios.

DEPENDENCIA ENTRE CLIENTES WEB PARA SIG

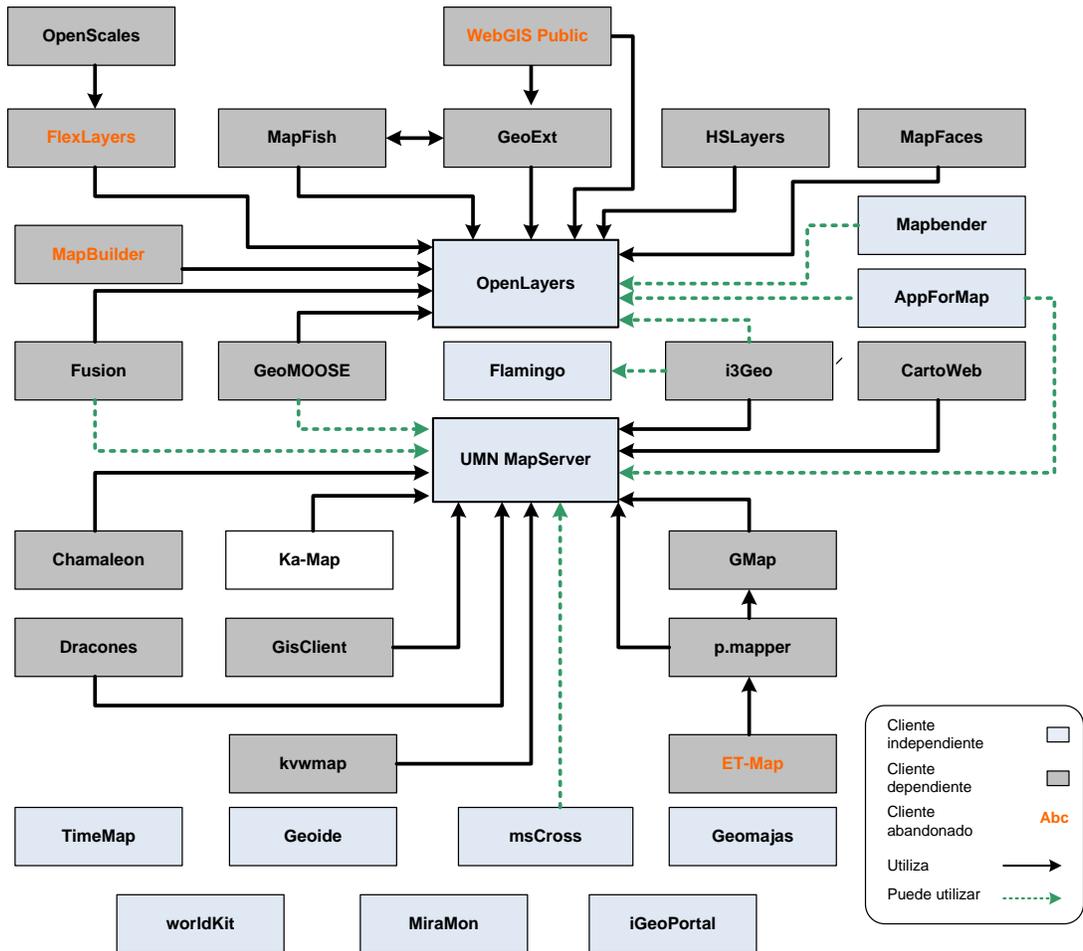


Figura 3. Dependencias entre clientes web para SIG de código abierto. Fuente: GeoTux

La mayoría de proyectos giran en torno a dos clientes base: *UMN MapServer* y *OpenLayers*. Los clientes que utilizan como base *UMN MapServer* fueron creados años atrás aprovechando las características que este cliente dispone: mapa, escala, mapa de referencia, herramientas de navegación básica, identificación de objetos espaciales; y su Interfaz de Programación de Aplicaciones (API) llamada *MapScript* que ha sido implementada en diferentes lenguajes de programación como *PHP*, *Python*, *Perl* y *Ruby*, y que continua su desarrollo adicionando funcionalidades como el etiquetado y la generación de gráficos de barra y de pastel. Por otra parte, la nueva generación de

clientes utiliza *OpenLayers* debido a su óptimo rendimiento en tareas de renderización en la *web*. Diferentes empresas contribuyen a su desarrollo y proyectos como *MapBuilder* han finalizado para acelerar su progreso.

Existen clientes que no se han basado en otros sino que han sido originados de manera independiente, como el caso de *Geomajas*, *iGeoPortal*, *Mapbender*, *TimeMap*, *MiraMon*, *Geoide* y *worldKit*. Algunos clientes utilizan *opcionalmente UMN MapServer* por medio de *MapScript* (*AppForMap*, *GeoMOOSE* y *msCross*) y otros permiten elegir una manera adicional para renderizar sus mapas con *OpenLayers* (*AppForMap*, *Mapbender e i3Geo*) y *Flamingo* (*i3Geo*).

Existe un número creciente de proyectos que aprovechan Flash para la construcción de aplicaciones enriquecidas de internet (*RIAs*), como por ejemplo *Flamingo*, *worldKit*, *OpenScales* y *Geoide*, brindando una nueva experiencia para los usuarios que buscan consultar mapas en línea.

A continuación se muestran las características generales y técnicas de los clientes web para SIG más utilizados en el medio.

OpenLayers

Es un cliente ligero construido con clases de *Javascript*, que permite al usuario elaborar mapas, haciendo uso de su propia base de información cartográfica o utilizando su estructura para hacer uso de otros servicios o incorporarlo en la web, sin dependencias de algún servidor en particular.

OpenLayers ofrece un API para acceder a diferentes fuentes de información cartográfica en la red: *Web Map Services*, Mapas comerciales (tipo *Google Maps*, *Bing*, *Yahoo*); el tipo de licencia de esta herramienta es *Berkeley Software Distribution (BSD)*.

Nombre del cliente web	OpenLayers
Licencia	BSD-style
País de origen	Estados Unidos
Entidad o empresa de origen	MetaCarta
Documentación	Idiomas: Español; Francés; Inglés; Portugués; Ruso Formatos: Blog; HTML; Trac; Wiki Niveles: Usuarios; Desarrolladores
Apoyo de OSGeo	Si (Graduado)
Observaciones	Librería Javascript sin dependencias en el servidor. Sirve de base para varios proyectos en la web. Soporta reproyección. Soporta SLD. Funcionalidades básicas de edición en línea. Desarrollo rápido. Gran cantidad de ejemplos.
Lenguaje en el que está escrito	Javascript
Lenguaje de programación que admite su API	Javascript
Servicios OGC que consume	WMS; WFS; Soporta las peticiones GetCapabilities, GetFeatureOfInterest y GetObservation de la especificación SOS. Recientemente se ha agregado soporte para la especificación WMTS.
Soporte de mapas basados en teselas	Si, Soporta gran cantidad de fuentes de mapas basados en tiles.
¿Requiere plug-ins privativos?	No
¿Incluye componente de metadatos?	Si, Soporta las peticiones GetDomain y GetRecords de la especificación CSW.
Listas de correo	Si, Tiene listas de correo para: Usuarios, desarrolladores, commits, anuncios, tilecache y trac, entre otras.

Tabla 1. Características del cliente ligero web para SIG *OpenLayers*.

MsCross

Fue desarrollado para permitir a los usuarios visualizar dinámicamente capas de información geográfica en la web. *MsCross* es un GIS construido en *JavaScript*.

Nombre del cliente web	MsCross
Licencia	GNU GPL
País de origen	Italia
Entidad o empresa de origen	Center for Advanced Studies, Research and Development in Sardinia
Documentación	Idiomas: Francés; Inglés; Italiano Formatos: HTML; Tracker Niveles: Usuarios; Desarrolladores Documentación incompleta.
Apoyo de OSGeo	No
Observaciones	Creado como cliente AJAX para UMN MapServer, actualmente puede funcionar prescindiendo de este. Consta de un solo archivo Javascript.
Lenguaje en el que está escrito	Javascript
Lenguaje de programación que admite su API	Javascript
Servicios OGC que consume	WMS; WFS Solo soporta puntos para el WFS.
Soporte de mapas basados en teselas	No
¿Requiere plug-ins privativos?	No
¿Incluye componente de metadatos?	No
Listas de correo	Si (Usuarios)

Tabla 2. Características del cliente ligero web para SIG *MsCross*.

MapBender

Es una aplicación programada en *PHP* y *JavaScript* que permite integrar servicios web basados en los estándares del *OGC* como *WMS*, *WFS* y *WMC*; forma parte de los proyectos por la *OSGeo* que consiste de un entorno para la publicación de Geoportales en la web que permite registrar, ver, publicar, controlar y garantizar el acceso seguro a servicios de infraestructura de datos espaciales.

Nombre del cliente web	Mapbender
Licencia	GNU GPL; Simplified BSD License
País de origen	Alemania
Entidad o empresa de origen	CCGIS
Documentación	Idiomas: Alemán; Inglés; Actualmente se trabaja sobre traducciones a varios idiomas: Sueco, español, búlgaro, polaco, italiano, francés, checo, esloveno, entre otros. Formatos: ODP; PDF; Trac; Video; Wiki Niveles: Usuarios; Desarrolladores
Apoyo de OSGeo	Si (Graduado)
Observaciones	Es un framework. Provee herramientas para la gestión de seguridad de servicios web e interfaces para la administración de usuarios y grupos. Recientemente se agregado OpenLayers como alternativa para el renderizado y JQuery para mejorar la integración con AJAX. Tiene cliente de servicios Gazetteer.
Lenguaje en el que está escrito	javascript; PHP
Lenguaje de programación que admite su API	javascript
Servicios OGC que consume	WMS; WFS; WFS-T; CSW
Soporte de mapas basados en teselas	Si, Aprovecha esta característica de OpenLayers.
¿Requiere plug-ins privativos?	No
¿Incluye componente de metadatos?	Si (Cumple con la especificación ISO-19119; cliente CSW)
Listas de correo	Si (Usuarios; Desarrolladores; Commits)

Tabla 3. Características del cliente ligero web para SIG *MapBender*.

P.mapper

Es un *framework* libre basado en *Mapserver* y en *PHPMapscript*. Nos brinda algunas herramientas listas para usar, y además, la capacidad de extender las funcionalidades por medio de *plugins*.

El desarrollo está a cargo de Armin Burger, quién lo liberó bajo una licencia GPL. El sitio web es: <http://www.pmapper.net/>

Algunas de las funciones incluidas en *p.mapper* son las siguientes:

- Con todas las funciones (consulta, seleccionar, buscar).
- Configuración es muy flexible de las funciones, comportamiento y el diseño.
- *Plugin* API para agregar funcionalidad personalizada.

Nombre del cliente web	p.mapper
Licencia	GNU GPL
País de origen	
Entidad o empresa de origen	Desarrollado por Armin Burger.
Documentación	Idiomas: Inglés La interfaz de usuario se encuentra en 20 idiomas diferentes. Formatos: HTML; PDF; Trac; Wiki Niveles: Usuarios
Apoyo de OSGeo	No
Observaciones	Es un framework. Está basado en UMN MapServer y PHP/MapScript. Provee un buen conjunto de herramientas listas para usar. Tiene una API de plugins para agregar funcionalidades. Utiliza jQuery para las funciones AJAX y para la interfaz. Se ofrece soporte comercial desde varios países.
Lenguaje en el que está escrito	Javascript; PHP
Lenguaje de programación que admite su API	Javascript; PHP
Servicios OGC que consume	WMS; WFS
Soporte de mapas basados en teselas	No
¿Requiere plug-ins privativos?	No
¿Incluye componente de metadatos?	No
Listas de correo	Si (Usuarios)

Tabla 4. Características del cliente ligero web para SIG *p.mapper*.

2.5 Conclusiones

Una vez entendidas las herramientas SIG como tal, podemos darnos cuenta de la funcionalidad que nos ofrecen y las ventajas de acceso a información. Para aprovechar mejor estas herramientas, es importante analizar las diferencias entre estas y las ventajas que tienen unas sobre otras y según sea el caso y las necesidades, utilizar la más adecuada de ellas.

En este capítulo se ha explicado, de forma general las características y funcionalidades del *software* libre para la generación de los geoservicios de una IDE; con la finalidad de ampliar el conocimiento y así abordar al siguiente capítulo.

Capítulo 3: Creación de la infraestructura de datos espaciales de la provincia de Azuay

3.1 Introducción

En el capítulo tres se revisará características importantes de las diferentes herramientas de software libre para la creación e implementación de una IDE. Se describirán los procedimientos para la creación del servicio *WMS* siguiendo la estandarización del *OGC*, la configuración del cliente ligero para la publicación de la geoinformación y consumo del servicio, tanto en clientes web como en clientes pesados; y finalmente la forma de publicar los metadatos.

Como se había mencionado en el capítulo anterior, todo el software que se utilizará es libre, para estar en consonancia con las últimas tendencias que sigue la Administración Pública del Ecuador.

El software a utilizarse del lado del servidor:

- Servidor de base de datos geográficos *PostGIS*
- Servidor de mapas *UMN MapServer*
- Servidor de metadatos *Geonetwork*

El software a utilizarse del lado del cliente:

- Cliente ligero web para SIG *p.mapper*
- Cliente pesado *gvSIG*

3.2 Publicación de la geoinformación como servicio *wms* en *mapserver*

3.2.1 Datos

La información que se publicó en la IDE del Azuay se encuentra agrupada como muestra la figura 4.

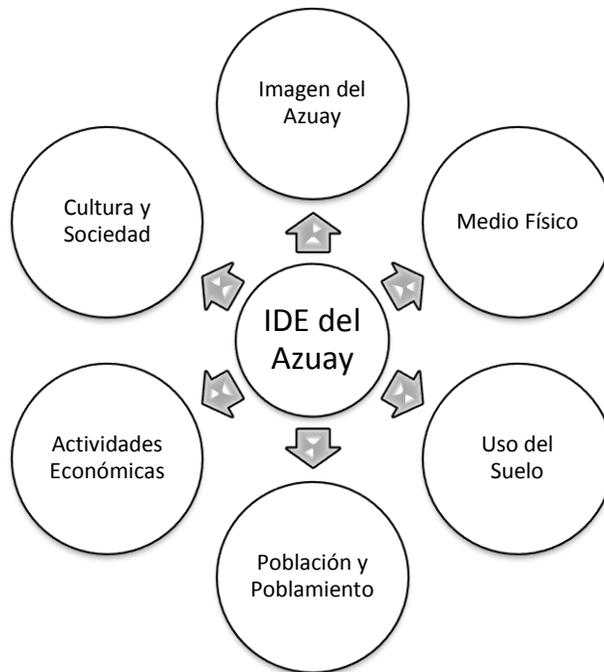


Figura 4. Grupos de Datos Geográficos de la IDE del Azuay.

<p>Imagen del Azuay</p> <ul style="list-style-type: none"> Mapa topográfico Imagen satelital Relieve Cuencas y sub cuencas hidrográficas División político-administrativa cantonal División político-administrativa parroquial 	<p>Uso del suelo</p> <ul style="list-style-type: none"> Cobertura del suelo Pisos zoogeográficos
<p>Medio físico</p> <ul style="list-style-type: none"> Suelos Precipitación Modelo digital de elevaciones Pendientes 	<p>Población y poblamiento</p> <ul style="list-style-type: none"> Población cantonal Densidad poblacional por cantón Densidad de asentamientos poblacionales
	<p>Actividades económicas</p> <ul style="list-style-type: none"> Parque Nacional Cajas Sitios turísticos
	<p>Cultura y sociedad</p> <ul style="list-style-type: none"> Sitios arqueológicos Centros históricos

Figura 5. Niveles de Datos Geográficos de la IDE del Azuay.

3.2.2 Estándar *ogc* del servicio *wms*

El servidor de mapas *mapServer* utiliza la estandarización del *OGC*, por dicho motivo es necesario e importante conocer las especificaciones del servicio *WMS*.

El servicio *Web Map Service (WMS)* definido por el *OGC (Open Geospatial Consortium)* produce mapas de datos referenciados espacialmente, de forma dinámica a partir de información geográfica. Este estándar internacional define un mapa como una representación de la información geográfica en forma de un archivo de imagen digital conveniente para la exhibición en una pantalla de ordenador. Un mapa no consiste en los propios datos. Los mapas producidos por *WMS* se generan normalmente en un formato de imagen como *PNG, GIF o JPEG*, y opcionalmente como gráficos vectoriales en formato *SVG (Scalable Vector Graphics)* o *WebCGM (Web Computer Graphics Metafile)*.

El estándar define tres operaciones:

Operación	Descripción
<i>GetCapabilities</i>	Devolver metadatos del nivel de servicio.
<i>GetMap</i>	Devolver un mapa cuyos parámetros geográficos y dimensionales han sido bien definidos.
<i>GetFeatureInfo</i>	Devolver información de características particulares mostradas en el mapa (opcionales).

Tabla 5. Operaciones *WMS*.

Las operaciones *WMS* pueden ser invocadas usando un navegador estándar realizando peticiones en la forma de *URLs (Uniform Resource Locators)*. El contenido de tales *URLs* depende de la operación solicitada. Concretamente, al solicitar un mapa, la *URL* indica qué información debe ser mostrada en el mapa, qué porción de la tierra debe dibujar, el sistema de coordenadas de referencia, la anchura y la altura de la imagen de salida. Cuando dos o más mapas se producen con los mismos parámetros geográficos y tamaño de salida, los resultados se pueden solapar para producir un mapa compuesto. El uso de formatos de imagen que soportan fondos transparentes permite que los mapas subyacentes sean visibles. Además, se puede solicitar mapas individuales de diversos servidores.

El servicio *WMS* permite así la creación de una red de servidores distribuidos de mapas, a partir de los cuales los clientes pueden construir mapas a medida. Las operaciones *WMS* también pueden ser invocadas usando clientes avanzados SIG, realizando igualmente peticiones en la forma de *URLs*.

3.2.1.1 Operaciones *web mapservice*

Las tres operaciones definidas por el *WMS* son: *GetCapabilities*, *GetMap* y *GetFeatureInfo*; que a continuación se describirán.

3.2.1.1.1 *GetCapabilities*

Consiste en la solicitud al servidor de un documento en formato *XML* donde se detallan las 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 *GetCapabilities* se debe armar un *URL Request* con ciertos parámetros.

Componentes	Obligatoriedad	Descripción
VERSION	Opcional	Versión de la especificación OGC
SERVICE=WMS	Obligatorio	Tipo de Servicio
REQUEST=GetCapabilities	Obligatorio	Nombre de la operación
FORMAT	Opcional	Formato de salida del metadato
UPDATESEQUENCE	Opcional	Número de secuencia o cadena para el control del caché.

Tabla 6. Parámetros de la Operación *GetCapabilities*.

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)*¹ 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.

3.2.1.1.2 GetMap

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.

¹Su función básica es la descripción de la estructura de datos, para usar una estructura común y mantener la consistencia entre todos los documentos que utilicen la misma *DTD*.

GetMapRequest

La solicitud es típicamente incrustada en el *URL* que es invocado en el *WMS* usando la operación *HTTP GET*. Los parámetros de un *RequestGetMap* se encuentran en la siguiente tabla.

Componentes	Obligatoriedad	Descripción
VERSION	Obligatorio	Versión de la especificación OGC
REQUEST=GetMap	Obligatorio	Nombre de la Petición
LAYERS	Obligatorio	Lista de una o más capas
STYLES	Obligatorio	Estilo de visualización por capa requerida
CRS=EPSG:identificador	Obligatorio	Sistema de Coordenadas de Referencia
BBOX=minx,miny,maxx,maxy	Obligatorio	Esquinas del ámbito
WIDTH	Obligatorio	Ancho del mapa en píxeles
HEIGHT	Obligatorio	Alto del mapa en píxeles
FORMAT	Obligatorio	Formato de salida del mapa
TRANSPARENT=TRUE FALSE	Opcional	Transparencia del fondo del mapa
BGCOLOR=color_value	Opcional	Valor del color del fondo RGB
EXCEPTIONS=exception_format	Opcional	Formato de las excepciones
TIME=time	Opcional	Valor de Tiempo en las capas deseadas
ELEVATION=elevation	Opcional	Elevación de las capas deseadas
Othersampledimension(s)	Opcional	Valor de otras dimensiones adecuadas

Tabla 7. Parámetros de la Operación *GetMap*.

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.

3.2.1.1.2 *GetFeatureInfo*

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 *requestGetFeatureInfo* 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 *requestGetMap*, 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 *graphicelement*, 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*.

GetFeatureInfoRequest

Los parámetros de un *RequestGetFeatureInfo* se encuentran en la siguiente tabla.

Componentes	Obligatoriedad	Descripción
VERSION	Obligatorio	Versión de la especificación OGC
REQUEST=GetFeatureInfo	Obligatorio	Nombre de la Petición
Parámetros del mapa	Obligatorio	Copia parcial de una petición
QUERY_LAYERS	Obligatorio	Lista de una o más capas
INFO_FORMAT	Obligatorio	Formato de respuesta
FEATURE_COUNT	Opcional	Número de objetos sobre los que se devuelve información
I=pixel_column	Obligatorio	Coordenada <i>i</i> del objeto en el Map CS, en píxeles.
J=pixel_row	Obligatorio	Coordenada <i>j</i> del objeto en el Map CS, en píxeles.
EXCEPTIONS	Opcional	Formato de las excepciones

Tabla 8. Parámetros de la Operación *GetFeatureInfo*.

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).

El *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.

3.2.3 Publicación en *mapserver*

La instalación y configuración de todos los componentes software necesarios para la publicación de la cartografía en *MapServer* se los realizó en el Sistema Operativo *centOS 5.3* de *Linux*; en el Anexo 1 se encuentran los pasos a seguir de la instalación y configuraciones para: el servidor web *Apache*, la base de datos *PostgreSQL* con la extensión *Postgis*, el entorno *PHP*, y el servidor de mapas *MapServer*.

3.2.3.1 Creación de la base de datos en *postgreSQL*

En primer lugar necesitamos crear un template tipo *postgis* en el cual se encuentren todas las estructuras y funciones geográficas.

Ingresar como usuario *postgres* al *template1*

```
[root@localhost ~]# sudo -u postgres psql template1
```

Crear la base de datos que servirá como *template*

```
template1=# create database template_postgis with template = template1;
```

Actualizar

```
template1=# UPDATE pg_database SET datistemplate = TRUE where datname = 'template_postgis';
```

Ingresar a la base *template_postgis*

```
template1=# \c template_postgis
```

Crear lenguaje *plpgsql*

```
template_postgis=# CREATE LANGUAGE plpgsql;
```

Importar

```
template_postgis=# \i /usr/share/pgsql/contrib/lwpostgis-64.sql;
```

Importar

```
template_postgis=# \i /usr/share/pgsql/contrib/spatial_ref_sys.sql;
```

Dar todos los permisos sobre la tabla “geometry_columns”

```
template_postgis=# GRANT ALL ON geometry_columns to PUBLIC;
```

Dar todos los permisos sobre la “spatial_ref_sys”

```
template_postgis=# GRANT ALL ON spatial_ref_sys to PUBLIC;
```

Liberar el almacenamiento de las tablas

```
template_postgis=# VACUUM FREEZE;
```

Una vez creado el template, se tiene que crear la base de datos.

Ingresamos con el usuario postgres

```
[root@localhost paquetes]# supostgres
```

Crear la base de datos “atlas_azuary”:

```
bash-3.2$ createdb -E UTF8 -T template_postgisatlas_azuary
```

```
bash-3.2$ createlangplpgsqlatlas_azuary
```

3.2.3.2 Exportar capas

Las capas tipo *shape* que se exportaron se enlistan en el siguiente reporte de la base de datos.

Tabla	Tamaño
Agua	728 kB
altoandino_azuary_utm_sam56	224 kB
ant_1parroquias_azuary_odeplan_250k_utm_sam56	416 kB
bosque_neblina_montano_norte_occidental_azuary_100k_utm_sam56	184 kB
bosque_neblina_montano_sur_occidental_azuary_100k_utm_sam56	224 kB
bosque_neblina_montano_sur_oriental_azuary_100k_utm_sam56	144 kB
bosque_semideciduo_montano_bajo_sur_occidental_azuary_100k_utm_s	56 kB
bosque_siempre_verde_montano_alto_norte_occidental_azuary_100k_u	296 kB
bosque_siempre_verde_montano_alto_sur_oriental_azuary_100k_utm_s	520 kB

bosque_siempre_verde_montano_bajo_800_1200_norte_occidental_azu	40 kB
bosque_siempre_verde_piemontano_0_300_norte_occidental_azuay_10	168 kB
bosque_siempre_verde_piemontano_300_1600_norte_occidental_azuay	288 kB
bosque_siempre_verde_piemontano_sur_occidental_azuay_100k_utm_s	88 kB
bosques_protectores_azuay_50k_utm_sam56	808 kB
buffer_n_vias_1_orden_azuay_50k_utm_sam56	1272 kB
buffer_n_vias_2_orden_azuay_50k_utm_sam56	432 kB
buffer_n_vias_3_orden_azuay_50k_utm_sam56	2624 kB
cabeceras_cantoniales_azuay_50k_utm_sam56	64 kB
cabeceras_parroquiales_azuay_odeplan_250k_utm_sam56	168 kB
cantones_azuay_hcpa_250k_utm_sam56	368 kB
cerros_lomas_azuay_250k_utm_sam56	40 kB
colostethus_vertebralis_azuay_500k_utm_sam56	96 kB
concesiones_mineras_existentes_azuay_50k_utm_sam56	256 kB
corredor_encanto_natural_50k_utm_sam56	176 kB
corredor_flores_frutas_50k_utm_sam56	104 kB
corredor_incas_encantados_50k_utm_sam56	48 kB
corredor_parque_nacional_cajas_50k_utm_sam56	320 kB
corredor_pueblos_caminos_pintorescos_50k_utm_sam56	232 kB
cuencas_hidrograficas_azuay_50k_utm_sam56	504 kB
isotermas_azuay_250k_utm_sam56	568 kB
isoyetas_azuay_250k_utm_sam56	528 kB
lagunas_azuay_50k_utm_sam56	192 kB
limite_provincial_legal	136 kB
limites_no_definidos_azuay_250k_utm_sam56	88 kB
matorral_humedo_montano_sur_occidental_azuay_100k_utm_sam56	88 kB
matorral_humedo_montano_sur_oriental_azuay_100k_utm_sam56	664 kB
matorral_seco_montano_sur_occidental_1_azuay_100k_utm_sam56	144 kB
matorral_seco_montano_sur_occidental_azuay_100k_utm_sam56	232 kB
meses_secos_azuay_250k_utm_sam56	536 kB
metallura_baroni_azuay_500k_utm_sam56	176 kB
Mosaicos	123 MB
n_vias_1_orden_azuay_50k_utm_sam56	352 kB
n_vias_2_orden_azuay_50k_utm_sam56copy	176 kB
n_vias_3_orden_azuay_50k_utm_sam56	1408 kB
norte_centro_occidental_azuay_100k_utm_sam56	88 kB
Paramo	104 MB
paramo_arbustivo_sur_oriental_azuay_100k_utm_sam56	168 kB
paramo_herbaceo_norte_occidental_azuay_100k_utm_sam56	88 kB
paramo_herbaceo_sur_occidental_azuay_100k_utm_sam56	112 kB
paramo_herbaceo_sur_oriental_azuay_100k_utm_sam56	416 kB
parques_nacionales_azuay_50k_utm_sam56	56 kB
Pastos	87 MB

poblacion_cantonal_2007	376 kB
pro_pob07	8192 bytes
provincias_area_estudio_hcpa_250k_utm_sam56	672 kB
quebradas_permanentes_azuary_50k_utm_sam56	3296 kB
quichua_domain_azuary_500k_utm_sam56	104 kB
rangos_pendientes_porcentaje	27 MB
rios_azuary_50k_utm_sam56	1216 kB
rios_principales_azuary_50k_utm_sam56	560 kB
rios_simples_dobles_azuary_50k_utm_sam56	1896 kB
sistema_hidrico_azuary_50k_utm_sam56	440 kB
st_ecoturismo_azuary_50k_utm_sam56	40 kB
st_lugares_visitar_azuary_50k_utm_sam56	40 kB
subcuencas_hidrograficas_azuary_hcpa_50k_utm_sam56	808 kB
subsistema_hidrico_azuary_50k_utm_sam56	440 kB
subtropical_azuary_utm_sam56	104 kB
subtropical_seco_azuary_utm_sam56	72 kB
suelo_desnudo	30 MB
suelos_azuary_aee_250k_utm_sam56	2144 kB
superparamo_norte_occidental_azuary_100k_utm_sam56	32 kB
sur_occidental_azuary_100k_utm_sam56	64 kB
sur_oriental_azuary_100k_utm_sam56	88 kB
templado_azuary_utm_sam56	232 kB
tropical_noroccidental_azuary_utm_sam56	112 kB
veg_leniosa	61 MB
zonficacion_ambiental_azuary_utm_sam56	80 kB
Capas no consideradas en el atlas del Azuay	
cumbres_azuary_250k_utm_sam56	40 kB
centros_poblados_azuary_250k_utm_sam56	72 kB
acequias_canales_azuary_50k_utm_sam56	368 kB
esteros_azuary_50k_utm_sam56	40 kB
nacimiento_fuentes_azuary_50k_utm_sam56	64 kB
pantano_inundacion_azuary_50k_utm_sam56	152 kB
piscina_reservorio_azuary_50k_utm_sam56	48 kB
quebradas_intermitentes_azuary_50k_utm_sam56	1720 kB
geologia_azuary_odeplan_500k_utm_sam56	224 kB

Tabla 9. Reporte de las capas tipo *shape* importadas en la base de datos.

El total es de 86 tablas en la base de datos de *PostgreSQL*. Las 4 capas tipo raster utilizadas para la elaboración de los mapas son los siguientes:

Tabla	Tamaño
azuay_17_c453_2001_sam56	65 MB
mdt_azuary	4 MB
Relieve	1 MB
densidad_poblacional_asentamientos	5 MB

Tabla 10. Reporte de las capas tipo *raster*.

En el Anexo 2 se encuentra el procedimiento para la exportación de toda la información cartográfica tipo *shape* en la base de datos *PostgreSQL / PotGIS* utilizando la herramienta *gvSIG*.

3.2.3.3 Crear y definir el *mapfile* requerido por *mapserver*

El archivo principal de configuración de *MapServer* es un archivo de texto, con extensión “.map”, en el que se incluye una serie de parámetros que definen las capas disponibles en el servicio, el estilo con que se representarán, su simbología, formato en el que se generará la imagen, el sistema de referencia, etc.

En archivo .map cada sección inicia con el nombre de la sección y termina con la palabra *END*. El contenido de las secciones consiste en la definición de determinados parámetros del tipo atributo - valor.

Para generar el *mapfile* nos podemos ayudar de la extensión de publicación del programa *gvSIG*. En el Anexo 3 encontramos el procedimiento para generar el archivo requerido por *mapserver*.

3.3 Visualización del servicio *wms* a través del cliente ligero *p.mapper*

3.3.1 Instalación y configuración de requisitos

Es necesario configurar el entorno PHP

Editar el archivo *php.conf*

```
[root@localhost ~]# vi /etc/httpd/conf.d/php.conf
```

Confirmar y agregar las líneas para que queden de la siguiente manera

```
AddHandler php5-script .php .phtml
```

```
AddType text/html .php .phtml
```

Reiniciar el servicio

```
[root@localhost ~]# servicehttpdrestart
```

Instalar librería requerida MDB2

```
[root@localhost ~]# yuminstall php-pear-MDB2
```

3.3.2 Instalación y configuración de la aplicación

3.3.2.1 Instalación

El visor de mapas *p.mapper* se lo puede descargar del sitio oficial:

<http://www.pmapper.net>

Desempaquetamos el archivo descargado y se lo puede renombrar; en nuestro caso lo llamaremos “visorAtlas”. Esta carpeta debe estar situada en el servidor web bajo la ruta `/var/www/html/`

Asignar permisos a la carpeta *legend*

```
[root@localhost ~]# chmod 777 /var/www/html/visorAtlas/images/legend/
```

Crear la carpeta temporal de imágenes

```
[root@localhost ~]# mkdir /var/www/html/imgpMapper
```

Asignar permisos a la carpeta temporal de imágenes

```
[root@localhost ~]# chmod 777 /var/www/html/imgpMapper
```

Para empezar, las principales configuraciones del visor están bajo el subdirectorio *config* en el archivo *config_default.xml*, tomamos como base los archivos que *p.mapper* tiene para testear la aplicación; para ello efectuamos una copia del archivo *config_default.xml* y de la carpeta *default* en el mismo directorio.

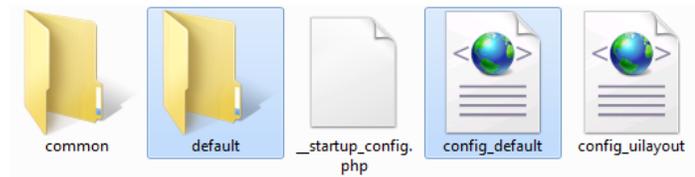


Figura 6. Archivos principales de configuración

La carpeta y el archivo copiado lo renombramos de acuerdo al tema del nuevo mapa; de la siguiente manera.

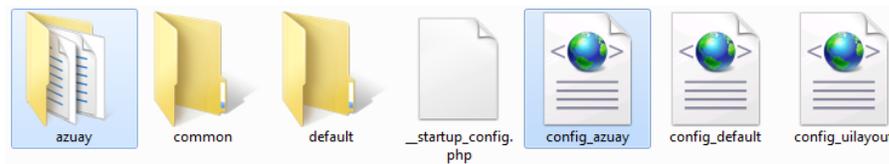


Figura 7. Archivos principales de configuración.

3.3.2.2 Adaptación del *mapfile*

Dentro de la carpeta *azuay* es necesario colocar el *mapfile* de nuestro servicio y realizar las adaptaciones necesarias para que funcione correctamente con *p.mapper*.

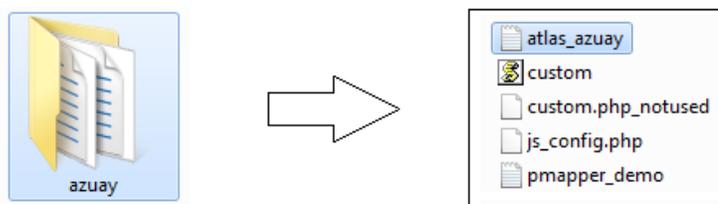


Figura 8. Archivos principales de configuración

Las secciones del *mapfile* que son indispensable ajustarlas, para que se visualice, son las siguientes:

MAP

SHAPEPATH

(FONTSET)

(SYMBOLSET)

WEB

IMAGEPATH "var/www/html/imgpMapper/"

IMAGEURL "/imgpMapper/"

Es recomendable utilizar como guía el ejemplo *pmapper_demo.map* que está incluido en el cliente ligero. En el ANEXO 4 se encuentre la configuración básica del *mapfile*.

3.3.2.3 Configuraciones de la aplicación

Configuraciones del archivo *config_azuay.xml*. Este fichero de configuración se encuentra separado por secciones, a continuación se explicarán las más importantes:

En la sección *<pmapper>* están las configuraciones de la aplicación en sí.

```
<pmapper>
  <pmTitle>Atlas Azuay</pmTitle>  ───────────▶ Titulo de la aplicación
  <debugLevel>3</debugLevel>
  <plugins>export</plugins>
  <plugins>scalebar</plugins>
  <plugins>transparency</plugins>
</pmapper>
```

Figura 9. Sección *<pmapper>* del archivo de configuración *config_azuay.xml*

En la sección *<config>* están las configuraciones relacionadas con las rutas.

```
<config>
  <pm_config_location>azuay</pm_config_location>  ───────────▶ carpeta en donde se encuentra el mapfile
  <pm_javascript_location>javascript</pm_javascript_location>
  <pm_print_configfile>common/print.xml</pm_print_configfile>
  <pm_search_configfile>inline</pm_search_configfile>
</config>
```

Figura 10. Sección *<config>* del archivo de configuración *config_azuay.xml*

En la sección *<map>* se configuran características de visualización y comportamiento de las capas.

Ubicar el mapfile

Existen dos maneras de ubicar el *mapfile*:

- 1) Usar el path absoluto en donde se encuentre el *mapfile*.
- 2) Solamente poner el nombre del *mapfile*, si este se encuentra bajo el directorio especificado en '*pm_config_location*'

```
<mapFile>atlas_azuary.map</mapFile> → Nombre del mapfile bajo el directorio  
<tplMapFile>common/template.map</tplMapFile> → declarado en pm_config_location
```

Figura 11. Sección `<map>` del archivo de configuración *config_azuary.xml*

Configuración de categorías

El usuario debe especificar las categorías y el nombre de las capas.

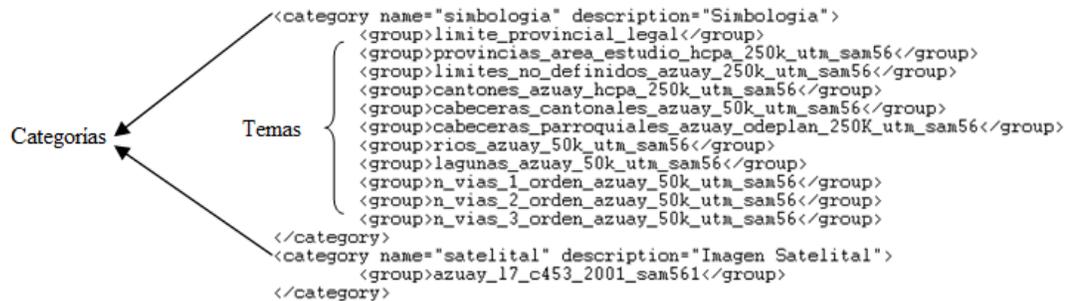


Figura 12. Sección `<category>` del archivo de configuración *config_azuary.xml*

Como resultado de la configuración de las categorías y capas, se obtiene la siguiente estructura tipo árbol que permitirá activar y desactivar las capas.

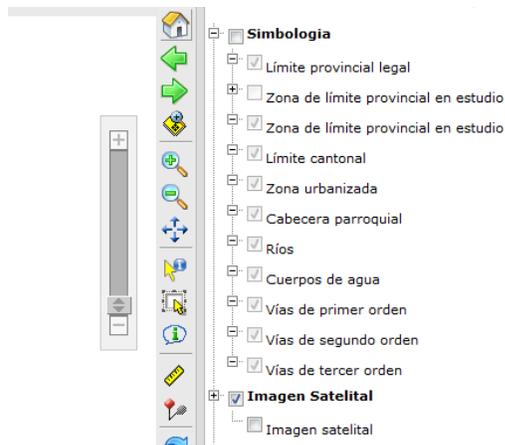


Figura 13. Resultado de la configuración del archivo *config_azuary.xml*

En el Anexo 5 se puede consultar todas las secciones del archivo de configuración *config_azuary.xml*

3.3.3 Personalización de la aplicación

Logos

Para personalizar el título y el logo de la aplicación; se necesita modificar el archivo *uiement.php* que se encuentra en el subdirectorio *incphp* del aplicativo.

En la función *public static function pmHeader()* configuramos:

```

$pmLogoUrl      =      array_key_exists('pmLogoUrl',      $_SESSION)      ?
$_SESSION['pmLogoUrl'] : "http://www.uazuay.edu.ec/";

$pmLogoTitle    =      array_key_exists('pmLogoTitle',    $_SESSION)      ?
$_SESSION['pmLogoTitle'] : "Universidad del Azuay";

$pmLogoSrc      =      array_key_exists('pmLogoSrc',      $_SESSION)      ?
$_SESSION['pmLogoSrc'] : "images/logos/uazuay.png";

```

Fondo

Para cambiar el color de fondo de la página, modificar el archivo *templates/layout.css*. En los estilos *[.pm-header]* y *[.pm-footering, .pm-footer div]* agregamos *[background-color:#005595;]*

De la misma manera editar el archivo *templates/jquery.layout.css*. En los estilos *[.ui-layout-north]* y *[.ui-layout-south]* agregar y/o modificar la etiqueta *background* con el color deseado.

Para configuraciones de layout, coordenadas, herramientas, consultas y otros ajustes básicos, consultar el siguiente sitio:

<http://geotux.tuxfamily.org/index.php/es/component/k2/item/252-algunas-configuraciones-b%C3%A1sicas-en-pmapper-faq>

3.3.4 Funcionalidades y herramientas de *p.mapper*

3.3.4.1 Funcionalidades

El visor web de mapas tiene a disposición las siguientes funcionalidades:

Búsqueda: Permite realizar la búsqueda de la capas que se encuentren disponibles para consulta.

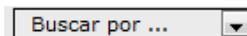


Figura 14. Funcionalidad de búsqueda.

Escala de representación: Muestra la escala de visualización de las coberturas según la vista o zoom realizado.



Figura 15. Funcionalidad de escala de presentación.

Escala gráfica: Muestra la escala gráfica en metros de la visualización según la vista o zoom realizado. Permite personalizar la escala de la visualización.



Figura 16. Funcionalidad de escala gráfica.

Coordenadas: Muestra la ubicación del puntero según las coordenadas geográficas.

X: 608086 Y: 9597043

Figura 17. Funcionalidad de coordenadas.

Barra de acercamiento (zoom): Permite realizar un acercamiento o alejamiento de la vista.



Figura 18. Funcionalidad de *zoom*.

Referencia de mapa: Muestra la ubicación de la vista en el mapa general.



Figura 19. Referencia de mapa.

3.3.4.2 Herramientas

Las herramientas que llegan por defecto en el aplicativo *pmapper* son las que se presentan en la figura 21.

Para agregar nuevas funcionalidades y comportamientos se puede consultar en el sitio oficial de la aplicación: <http://svn.pmapper.net/trac/wiki/AvailablePlugins>. Aquí encontraremos la forma de configurar los *plugins* que trae consigo *pmapper* en lo referente a personalizaciones en la leyenda, en la visualización, en consultas, en la interfaz, en el mapa, etc.

	Visualización completa: Muestra la visualización completa de la vista o coberturas.
	Vista anterior: Muestra la vista anteriormente visualizada.
	Vista siguiente: Muestra la vista posteriormente visualizada.
	Zoom a la selección: Permite realizar un zoom al objeto seleccionado.
	Acercar: Permite realizar un zoom de acercamiento.
	Alejar: Permite realizar un zoom de alejamiento.
	Mover: Permite arrastrar la vista hacia donde queramos.
	Identificar: Identifica la información de las capas consultables activadas en la selección.
	Seleccionar: Permite consultar la información del área seleccionada.
	Autoidentificar: Identifica automáticamente la información de donde se encuentra el puntero.
	Medida: Permite realizar segmentos y muestra las medidas de los mismos en perímetro (Km) y área (Km ²).
	Punto de interés: Con un clic podemos colocar y etiquetar puntos de interés.
	Refrescar el mapa: Actualiza la vista

Figura 20. Barra de herramientas.

3.4 Publicación de metadatos con *geonetwork*

La instalación y configuración de todos los componentes software necesarios para la herramienta de metadatos *Geonetwork* se los realizó en el Sistema Operativo *centOS 5.3* de *Linux*; en el Anexo 6 se encuentran los pasos a seguir de la instalación y configuraciones para: *JAVA*, servidor *Apache–Tomcat* y *Geonetwork*.

3.4.1 Configuraciones *geonetwork*

Para la configuración de la aplicación, se debe iniciar sesión como administrador con usuario y la contraseña “admin”, es importante cambiar estos valores desde la página de administración, una vez que se ha iniciado la sesión. Desde el menú en la página “Administración” seleccionar “Configuración del sistema”.

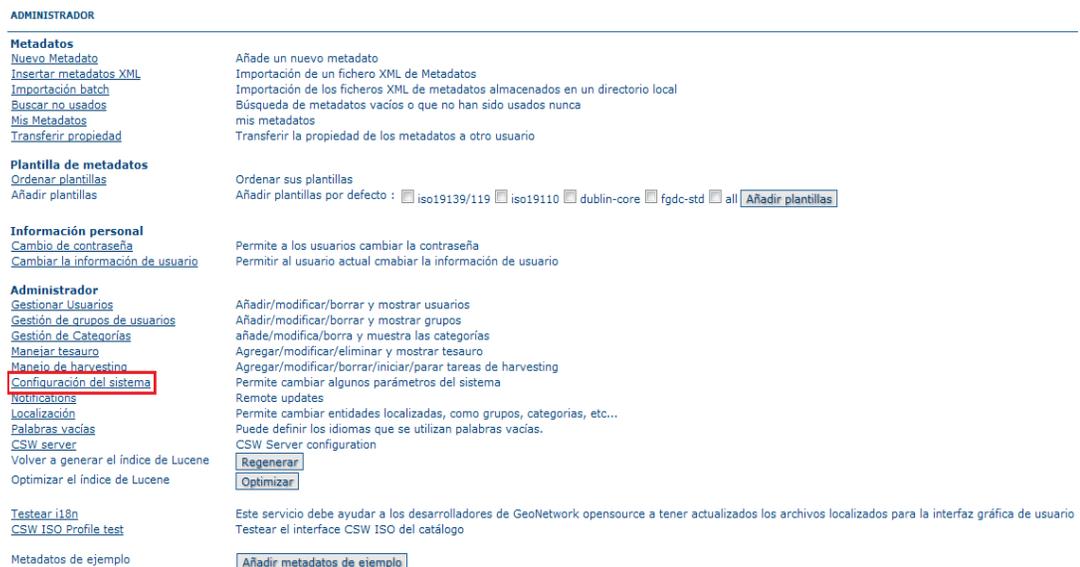


Figura 21. Página de administración de *Geonetwork*.

Al seleccionar “Configuración del sistema” aparecerá un conjunto de parámetros que se pueden cambiar, a continuación una descripción detallada de los mismos.

Durante una sesión de edición, al agregar enlaces de datos a una de metadatos. El anfitrión y el puerto serán utilizados para construir los enlaces de descarga para almacenar dentro de los metadatos. Durante peticiones *CSW (Catalog Service Web)*.

La operación *GetCapabilities* devuelve un documento *XML* con enlaces *HTTP* a los servicios de la *CSW*. Estos enlaces son de forma dinámica con el anfitrión y los valores de puerto.

Parámetros generales del sitio

Nombre: El nombre de la instalación de *GeoNetwork*. Este nombre se utiliza para identificar el nodo de operaciones.

Organización: La organización pertenece al nodo. Solo por razones informativas.

Servidor: del nodo o número IP. Si el nodo es accesible al público a través de Internet, se debe de usar el dominio de la máquina / dirección. Una configuración típica es tener un servidor apache en la dirección A, que es de acceso público y redirige las peticiones a un servidor *Tomcat* en una dirección privada B (generalmente localhost).

Puerto: El nodo del puerto (generalmente es 80 o 8080).

Red: La red interna de direcciones IP en forma (generalmente 127.0.0.0).

Máscara: De la máscara de la red (generalmente 255.0.0.0)

Servidor Z39.50: *GeoNetwork* puede actuar como un servidor Z39.50, que es un protocolo de comunicación *OGC* para consultar y recuperar los metadatos, normalmente esta activa la casilla.

Puerto Z39.50: Este es el puerto en el que *GeoNetwork* escucha peticiones entrantes Z39.50. Por lo general, el valor de 2100 es un estándar.

Correo electrónico y notificación

Si se desea enviar correo electrónico se debe configurar el servidor de correo *GeoNetwork*.

RETROALIMENTACIÓN

Correo

Servidor SMTP

Puerto SMTP

Figura 22. El correo de las opciones de configuración del servidor

Email: Esta es la dirección de correo electrónico que se utilizara para enviar el correo electrónico (la dirección del remitente).

Servidor SMTP: la dirección del servidor de correo para utilizar al enviar correo electrónico.

Puerto SMTP: el servidor de correo SMTP puerto (normalmente 25).

Eliminación de Metadatos

Define el directorio utilizado para almacenar una copia de seguridad de los metadatos y los datos después de una acción de eliminación. Este directorio se utiliza como un directorio de copia de seguridad para permitir a los administradores del sistema recuperar los metadatos y, posiblemente, los datos relativos después de la eliminación errónea. Por defecto, el directorio eliminado se crea en la carpeta de datos.

Autenticación

En esta sección se define la fuente contra la cual *GeoNetwork* autentica los usuarios.

AUTENTICACIÓN

Usar Login:

GeoNetwork Autenticación
Activar Auto-Registro de usuarios

LDAP Autenticación

Permitir a otros usuarios hacer login usando:

Shibboleth Autenticación

Figura 23. Opciones de configuración de autenticación

De manera predeterminada, los usuarios se autentican con información celebrada en la base de datos *GeoNetwork*. Cuando la base de datos *GeoNetwork* se utiliza como fuente de autenticación, auto registro de usuarios puede ser activada.

Los usuarios pueden elegir para autenticar inicios de sesión en contra de cualquiera de las tablas de bases de datos *GeoNetwork;LDAP* (el *Lightweight Directory Access Protocol*). También puede configurar otras fuentes de autenticación. En la actualidad, *Shibboleth* es la única fuente de autenticación adicional que se pueden configurar. *Shibboleth* se utiliza normalmente para las federaciones nacionales de acceso.

3.4.2 Personalizar *geonetwork*

Se puede personalizar la página inicial de *Geonetwork*, como las capas que se van a visualizar en el *mapviewer*; estos cambios se realizan dentro del archivo *config-gui.xml*, se puede observar en el Anexo 7 el contenido de este archivo.

3.4.3 Creación de nuevos grupos de usuarios

El administrador puede crear nuevos grupos de usuarios. Los grupos de usuarios pueden corresponder a las unidades lógicas dentro de una organización. Por ejemplo, grupos de BaseDatos, Programadores, Analistas, etc. Para crear nuevos grupos:

1. Seleccione la pestaña de Administrador y seleccione Gestión de grupos de usuarios.
2. Seleccione Agregar un nuevo grupo.
3. Rellene los detalles. La dirección de correo electrónico será utilizada para enviar información sobre la descarga de datos cuando se producen por los recursos que forman parte del grupo. Clic en guardar.

Nota: el nombre no debe contener espacios en blanco. Puede utilizar las funciones de localización para proporcionar los nombres localizados de los grupos.

3.4.4 Creación de nuevos usuarios

1. Seleccione la pestaña de Administrador y seleccione Gestionar usuarios.
2. Clic en el botón Añadir un nuevo usuario.
3. Proporcionar la información requerida para el nuevo usuario, asignar el perfil adecuado, asignar el usuario a un grupo. Clic en Guardar.

3.4.5 Insertar plantilla *xml*

1. Seleccione la pestaña de Administrador y seleccione Insertar metadatos XML.

2. Proporcionar la información requerida para la plantilla. En “tipo” asignar: plantilla de metadatos y asignar a un grupo. Clic en Insertar.

IMPORTACIÓN DE UN FICHERO XML DE METADATOS

Modo de inserción: Subir fichero Copiar/Pegar
Tipo de fichero: Fichero (XML, SLD, WMC...) Fichero MEF
Metadatos: PEM_Vector.xml
Tipo:
Acciones de importación: No realizar ninguna acción en la importación
 Sobrescribir los metadatos con el mismo UUID
 Generar UUID para el metadato insertado
Hoja de Estilo:
Validar:
Grupo de usuarios:
Categoría:

Figura 24. Importación del archivo XML.

Para elaboración de los metadatos de la IDE de la provincia del Azuay en *Geonetwork* se usó la plantilla, *PEM_Vector.xml*, plantilla basada en las normas de metadatos ISO 19115:2003 e ISO 199115-2:2009, que fue desarrollada por un grupo de instituciones especializadas las cuales acordaron el Perfil Ecuatoriano de Metadatos (PEM). Esta plantilla la podemos descargar del siguiente sitio:

<http://www.geoportaligm.gob.ec/portal/articulos-tecnicos/infraestructura-de-datos-espaciales/PEM-Vector.xml>

3.4.6 Construcción de los metadatos

3.4.6.1 Datos

Para llenar la plantilla de metadatos se usó el manual: Infraestructura ecuatoriana de datos geoespaciales, que está aprobada por el Consejo Nacional de Geoinformática (CONAGE). Para mayor información se le puede descargar del siguiente link: http://territorial.sni.gob.ec/images/stories/documentos/documentosconage/DOC_2011/metadatos_2011.pdf

Los metadatos creados para esta IDE del Azuay, por cuestiones técnicas, son únicamente los de la carpeta “Cartografía Base”, del Atlas de la provincia del Azuay. Está información fue proporcionada por la universidad.

3.4.6.2 Creación de metadatos

1. Seleccione la pestaña de Administrador y seleccione nuevo metadato.
2. Seleccionar la plantilla de metadatos.
3. Después de seleccionar la plantilla correcta, es necesario asignar un grupo de usuarios de los metadatos que pertenece; y finalmente hacer clic en Crear.

Un formulario de metadatos basados en la plantilla seleccionada se mostrará para ser llenada.

3.4.6.3 Creación de una vista en mapa miniatura en los metadatos

Para crear una vista del mapa en miniatura:

1. Seleccione su mapa y clic en Editar.
2. En el menú de edición, clic en el botón otras acciones que se encuentra en la parte superior de la página y luego clic en vista previa
3. Aparece el asiste de gestión de vistas previas. Para crear una miniatura pequeña o grande, clic en el botón seleccionar archivo. Se recomienda que utilice 180 píxeles de miniaturas pequeñas y 800 x 600 para imagines en miniatura de gran tamaño. Puede utilizar formatos GIF, PNG y JPEG para la imagen.
4. Clic en añadir y la imagen se agregará a la página.
5. A continuación, clic en volver y guardar el registro.

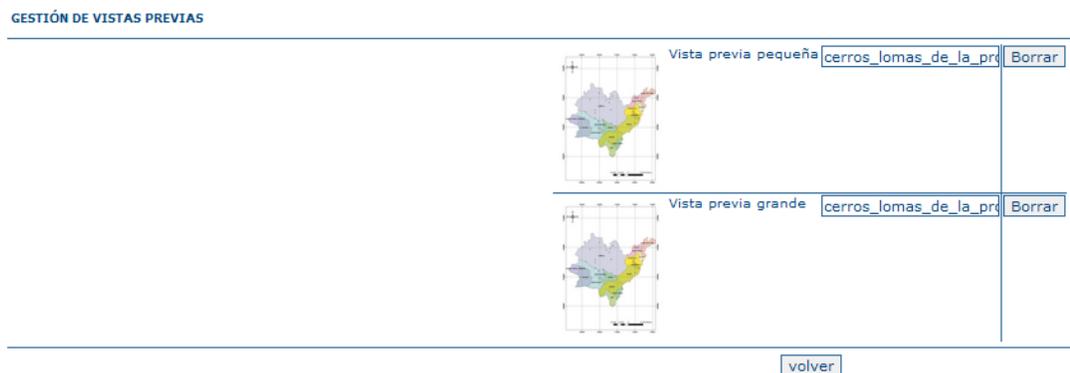


Figura 25. Vista en mapa miniatura en geonetwork

CONCLUSIONES

Una vez culminado el desarrollo de esta monografía podemos concluir:

1. En nuestra sociedad actual, la información geográfica está en un proceso de desarrollo muy importante, en la que gradualmente las IDEs comienzan a formar parte. Este hecho está permitiendo, a la administración pública y organismos privados, el mejoramiento en la toma de decisiones que tenga relación con el territorio y su planificación.
2. Una infraestructura de datos espaciales está constituida por metadatos, conjuntos de datos espaciales y los servicios de datos espaciales; los servicios y tecnologías de red, acceso y utilización. Además, mecanismos, procesos y procedimientos de coordinación y seguimiento que en nuestro país, por falta de una iniciativa de esta índole, aún no se han establecido.
3. Para creación del servicio *Web Map Service (WMS)* de la infraestructura de datos espaciales para la provincia del Azuay, se utilizó las especificaciones internacionales propuestas por la entidad reguladora *Open Geospatial Consortium (OGC)*; de la misma forma para la creación de los metadatos se siguió el documento desarrollado por el Perfil Ecuatoriano de Metadatos (PEM) que está basado en las normas ISO 19115:2003 e ISO 199115-2:2009.
4. Para facilitar el acceso y uso de la geoinformación de esta región, se centralizó la información espacial en la base de datos “atlas_azuguay” y se adaptó el cliente ligero web *pmapper* para consumir el servicio *WMS*.
5. La utilización de herramientas de software libre en la implementación, brindan una solución estable y económica; pero con cierto nivel de complicación en la instalación.

6. Los metadatos describen los datos, constituyen la información en forma de documentación, que permite que los datos sean bien entendidos, compartidos y explotados eficazmente por los usuarios y facilita en la toma de de los mismos.

7. El metadato es un elemento clave en una Infraestructura de Datos Espaciales debido a que es la única forma en la que el IDE se puede sustentar en un mundo totalmente comunicado como lo es en la actualidad.

RECOMENDACIONES

Al concluir éste trabajo y luego de realizar pruebas pertinentes de funcionamiento se plantean las siguientes recomendaciones:

1. Aplicar métodos de optimización para conseguir una mayor velocidad de respuesta en el servicio.
2. Crear un script para remover automáticamente todos los archivos temporales generados por *pmapper* en el directorio especificado en el *mapfile* en la sección *WEB-IMAGEPATH*.
3. Registrar los metadatos en el momento en que se generan o recogen los datos. Cuando esta captura se realiza de manera posterior, se corre un alto riesgo de obtener información menos precisa y el tiempo utilizado para esta actividad será mayor debido a que deben buscarse los datos que permitan un registro correcto.
4. Usar herramientas libres *Open Source*, para la integración de los metadatos en una Infraestructura de datos espaciales, ya que reducen notablemente los costos en la implantación.

BIBLIOGRAFÍA

Artículos

Subirana, Joan Capdevila i. <<Scripta Nova>> 1 de Agosto de 2004. REVISTA ELECTRÓNICA DE GEOGRAFÍA Y CIENCIAS SOCIALES. Recuperado el 1 marzo de 2012 <<http://www.ub.edu/geocrit/sn/sn-170-61.htm>>.

Gomariz, F.<<DUGImedia>> 11 de marzo de 2010. Propuesta de un SIG para su uso en la gestión de zonas costera. A 'IV Jornadas de SIG Libre'. Girona: Universitat. Recuperado el 1 de marzo de 2012 < <http://hdl.handle.net/10256.1/1560>>

Benavidez, César. <<Dspace>> 15 de octubre de 2010. ervidoresWebMapping para Bases de Datos Espaciales. Aplicativo: Sistema de Información Geográfica (SIG) Prototipo del Cuerpo de Bomberos de Riobamba. Recuperado 2 de marzo de 2012 < <http://dspace.esPOCH.edu.ec/handle/123456789/471>>

Carrillo, German. <<Geotux>> 30 de noviembre de 2010. Comparación de clientes web para SIG v.5. Recuperado el 2 marzo de 2012 < <http://geotux.tuxfamily.org/index.php/es/component/k2/item/265-comparacion-de-clientes-web-para-sig-v5#t1r5>>

GOMEZ GOMEZ, Wilfrido.<<Slideshare Present Yourself.>>27 de Septiembre de 2008. Recuperado el 4 de Marzo de 2012 <<http://www.slideshare.net/SIGSI/servidore-geograficos-presentation>>.

HINOJOSA, Jhon. <<SlidesharePresentYourself.>>22 de Enero de 2010. Recuperado el 28 de Febrero de 2012 < <http://www.slideshare.net/jhp1411/openlayers>>.

Páginas web

Cosejo superior cartográfico IDEE. Recuperado el 1 de Marzo de 2012, de http://www.idee.es/show.do?to=pideep_Info_IDEs.ES

CatMDEdit. Geoportal sobre METADATOS de Información Geográfica. 8 de Octubre de 2010. Recuperado el 7 de Marzo de 2012, de http://metadatos.ign.es/metadatos/Como_se_crean/2-b-herramientas/a-open-source/catmdedit/fhsfgh.

Free Software Foundation. GNU Operating System.9 de Diciembre de 2012.Recuperado el 2 de Marzo de 2012 <<http://www.gnu.org/philosophy/free-sw.es.html>>.

Geonetwork. (2011). Recuperado el 10 de Marzo de 2012, de <http://geonetwork-opensource.org/>.

Geoserver. (2009). Recuperado el 20 de Febrero de 2012, de <http://geoserver.org/display/GEOS/Welcome>.

MapBender. (2010). Recuperado el 5 de marzo de 2012, de http://www.mapbender.org/index.php/Utilizaci%C3%B3n_Mapbender.

MapServer. (2009). Recuperado el 20 de Febrero de 2012, de <http://mapserver.org/>.

MySQL. (2009). Recuperado el 14 de Febrero de 2012, de <http://dev.mysql.com/doc/refman/5.0/es/features.html>.

PostGIS. (2011). Recuperado el 14 de Febrero de 2012, de <http://postgis.refractor.net/>.

Wikipedia. (28 de Enero de 2012). Recuperado el 1 de Marzo de 2012, de <http://es.wikipedia.org/wiki/GeoServer>.

Wikipedia. (15 de Febrero de 2012). Recuperado el 5 de Marzo de 2012, de <http://es.wikipedia.org/wiki/MapServer>.

OGC. (2012). Recuperado el 20 Marzo de 2012, de <http://www.opengeospatial.org/standards/wms>

Geoportal Instituto Geográfico Militar. (2010). Recuperado el 20 Marzo de 2012, de <http://www.geoportaligm.gob.ec/portal/articulos-tecnicos/infraestructura-de-datos-espaciales/PEM-Vector.xml/view>

Infraestructura de datos espaciales de España (2009). Recuperado el 23 de Marzo de 2012, de http://www.idee.es/show.do?to=pideep_desarrollador_wms.ES

GeoTux. (15 de Abril de 2010). Recuperado el 24 de Marzo de 2012, de <http://geotux.tuxfamily.org/index.php/es/component/k2/item/252-algunas-configuraciones-b%C3%A1sicas-en-pmapper-faq>

Pmapper. (2011). Recuperado el 24 de Marzo de 2012, de <http://www.pmapper.net>

Perfil Ecuatoriano de Metadatos. (2009). Recuperado el 27 de Marzo de 2012, de <http://www.geoportalignm.gob.ec/portal/articulos-tecnicos/infraestructura-de-datos-espaciales/PEM-Vector.xml/view>

Consejo Nacional de Geomática. (2004). Recuperado el 5 de Febrero de 2012, de http://territorial.sni.gob.ec/images/stories/documentos/documentosconage/DOC_2011/metadatos_2011.pdf

Anexo 1: Instalación y configuración de *mapserver* en el sistema operativo *linux centos 5.x*

1. Configuraciones previas

Ingresar como administrador

```
[root@localhost ~]# su -
```

Agregar repositorio rpmforge

```
[root@localhost ~]# rpm -Uvh http://packages.sw.be/rpmforge-  
release/rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
```

Actualizar el yum

```
[root@localhost ~]# yum -y update
```

2. Instalación de Apache

En la distribución centOS de Linux el servidor apache ya viene instalado por defecto.

Instalar o actualizar el apache.

```
[root@localhost ~]# yum -y installhttpd
```

Iniciar el servicio.

```
[root@localhost ~]# servicehttpdstart
```

Configurar para arranque automático.

```
[root@localhost ~]# chkconfighttpdon
```

3. Instalación de postgresQL 8.4

Desactivar postgresQL del repositorio de centOS

```
[root@localhost ~]# vi /etc/yum.repos.d/CentOS-Base.repo
```

En la parte inferior de las secciones de [base] y [updates] agregar la siguiente línea

```
exclude=postgresql*
```

Crear una carpeta para descargar los paquetes necesarios.

```
[root@localhost ~]# mkdir /usr/local/paquetes
```

Ubicarse en la nueva carpeta creada.

```
[root@localhost ~]# cd /usr/local/paquetes
```

Importar llave necesaria

```
[root@localhost paquetes]# rpm --import  
http://yum.pgsqlrpms.org/RPM-GPG-KEY-PGDG
```

Descargar la configuración del yum para la instalación de PostgreSQL

```
[root@localhost paquetes]#  
wgethttp://yum.pgsqlrpms.org/reporpms/8.4/pgdg-centos-8.4-  
2.noarch.rpm
```

Instalar la configuración descargada para el yum.

```
[root@localhost paquetes]# rpm -i pgdg-centos-8.4-2.noarch.rpm
```

Eliminar el protocolo NFS

```
[root@localhost paquetes]# yum -y removenfs-utils
```

Actualizar el yum para los nuevos cambios

```
[root@localhost paquetes]# yum -y update
```

Instalar PostgreSQL

```
[root@localhost paquetes]# yum -y installpostgresqlpostgresql-server  
postgresql-devel
```

Iniciar el cluster de datos

```
[root@localhost paquetes]# servicepostgresqlinitdb
```

Iniciar el servicio

```
[root@localhost paquetes]# servicepostgresqlstart
```

Cambiar la contraseña de súper usuario.

```
[root@localhost paquetes]# supostgres
bash-3.2$ psql -d template1
template1=# ALTER USER postgres WITH PASSWORD '*****';
template1=# \q
bash-3.2$ exit
```

Editar archivo de configuración.

```
[root@localhost paquetes]# vi /var/lib/pgsql/data/pg_hba.conf
```

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all ident
local all all md5
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 ident
```

Figura 26. Archivo de configuración pg_hba.conf.

En esta sección agregamos la dirección IP desde cual puede ser accedida remotamente la base de datos.

Reiniciamos el servicio

```
[root@localhost paquetes]# servicepostgresqlrestart
```

Editar archivo de configuración

```
[root@localhost paquetes]# vi /var/lib/pgsql/data/postgresql.conf
```

Verificar y des comentar estos valores

```
listen_addresses = '*'           # what IP address(es) to listen on;  
                                # comma-separated list of addresses;  
                                # defaults to 'localhost', '*' = all  
                                # (change requires restart)  
port = 5432                      # (change requires restart)
```

Figura 27. Archivo de configuración postgresql.conf.

Configurar para arranque automático

```
[root@localhost paquetes]# chkconfigpostgresqlon
```

4. Instalar el entorno PHP

Requisitos

```
[root@localhost]# yum -y installunixODBCunixODBC-devel
```

Instalar el entorno

```
[root@localhost ]# yum -y installphpphp-devel
```

Reiniciamos el apache

```
[root@localhost ~]# servicehttpdrestart
```

5. Instalar pre-requisitos para *mapServer*

Instalar librerías de proyecciones.

```
[root@localhost paquetes]# yum -y installprojproj-develproj-epsgproj-nad
```

Instalar librería para topología espacial.

```
[root@localhost paquetes]# yum -y install geos-3.1.0-1.el5.rf
```

Instalar compilador

```
[root@localhost paquetes]# yum -y installgcc-c++
```

Instalar Flex

```
[root@localhost paquetes]# yum -y install flex
```

6. Instalación de postGIS y módulos necesarios

Instalar postGIS

```
[root@localhost paquetes]# yum -y install postgis
```

Módulo postgresQL

```
[root@localhost paquetes]# yum -y install php-pgsql
```

7. Creación de la Base de datos espacial en postgresQL 8.3 sin usar template postgis

Creación de la base de datos

```
[root@localhost paquetes]# supostgres
bash-3.2$ createdb -E UTF8 nombreBase
bash-3.2$ createlang plpgsql nombreBase
```

Agregamos la referencia espacial a la base creada

```
bash-3.2$ psql -d nombreBase -f /usr/share/pgsql/contrib/lwpostgis-
64.sql
bash-3.2$ psql -d nombreBase -f
/usr/share/pgsql/contrib/spatial_ref_sys.sql
bash-3.2$ exit
```

8. Instalar MapServer

Instalar requisitos

```
[root@localhostpaquetes]# yum -y install xerces-c-devel
[root@localhostpaquetes]# yum -y install libgssapi-devel
[root@localhost paquetes]# yum -y install krb5-devel
[root@localhost paquetes]# yum -y install e2fsprogs-devel
```

```
[root@localhost paquetes]# yum -y install libidn-devel
[root@localhost paquetes]# yum -y install openssl-devel
[root@localhost paquetes]# yum -y install gdgd-devel
[root@localhost paquetes]# yum -y install libxml2 libxml2-devel
[root@localhost paquetes]# yum -y install curlcurl-devel
[root@localhost paquetes]# yum -y install giflibgiflib-devel
```

Descargar los archivos, en caso de 32 bits buscarlos.

```
[root@localhost paquetes]# wget
http://dag.wieers.com/rpm/packages/netcdf/netcdf-3.6.2-
1.el5.rf.x86_64.rpm
[root@localhost paquetes]# wget
http://dag.wieers.com/rpm/packages/netcdf/netcdf-devel-3.6.2-
1.el5.rf.x86_64.rpm
```

Instalar los archivos descargados

```
[root@localhost paquetes]# rpm -i netcdf-3.6.2-1.el5.rf.x86_64.rpm
[root@localhost paquetes]# rpm -i netcdf-devel-3.6.2-
1.el5.rf.x86_64.rpm
```

Cambiar de directorio

```
[root@localhost paquetes]# cd /usr/local/src
```

Descargar el comprimido

```
[root@localhostsrc]# wget http://download.osgeo.org/gdal/gdal-
1.6.0.tar.gz
```

Descomprimir

```
[root@localhostsrc]# tar -xvzf gdal-1.6.0.tar.gz
```

Ingresar en el directorio descomprimido

```
[root@localhostsrc]# cd gdal-1.6.0
```

Instalar el descomprimido con los siguientes pasos

```
[root@localhost gdal-1.6.0]# ./configure --with-pg=/usr/bin/pg_config
```

```
[root@localhost gdal-1.6.0]# make
```

```
[root@localhost gdal-1.6.0]# make install
```

Nota: Si hay un error en la lectura libxpat.so porque es de 64 bits, entonces editar GDALmake.opt y cambiar /usr/lib a /usr/lib64

Instalar otros requisitos

```
[root@localhost gdal-1.6.0]# yum -y install freetypefreetype-devel
```

```
[root@localhost gdal-1.6.0]# yum -y install libpnglibpng-devel
```

```
[root@localhost gdal-1.6.0]# yum -y install libjpeglibjpeg-devel
```

```
[root@localhost gdal-1.6.0]# yum -y install libtiffibtiff-devel
```

```
[root@localhost gdal-1.6.0]# yum -y install aggagg-devel
```

Instalación de MAPSERVER 5.2.1

Cambiar de directorio

```
[root@localhost gdal-1.6.0]# cd /usr/local/src
```

Descargar el comprimido

```
[root@localhostsrc]# wget
```

```
http://download.osgeo.org/mapserver/mapserver-5.2.1.tar.gz
```

Descomprimir

```
[root@localhostsrc]# tar -xvzf mapserver-5.2.1.tar.gz
```

Ingresar en el directorio descomprimido

```
[root@localhostsrc]# cd mapserver-5.2.1
```

Compilación de Mapserver

```
[root@localhost mapserver-5.2.1]# ./configure --with-gd=/usr --with-freetype=/usr/bin/freetype-config --with-tiff=/usr --with-proj=/usr --with-curl-config=/usr/bin/curl-config --with-httpd=/usr/sbin/httpd --with-postgis=/usr/bin/pg_config --with-php=/usr --with-eppl --with-wfs --with-wcs --with-wmsclient --with-wfsclient --with-sos --with-ogr=/usr/local/bin/gdal-config --with-gdal=/usr/local/bin/gdal-config --with-geos=/usr/bin/geos-config --prefix=/usr
```

```
[root@localhost mapserver-5.2.1]# make
```

```
[root@localhost mapserver-5.2.1]# yum install gdal
```

Copiararchivosnecesarios

```
[root@localhost mapserver-5.2.1]# cp -f legend /usr/local/bin/legend
```

```
[root@localhost mapserver-5.2.1]# cp -f msencrypt  
/usr/local/bin/msencrypt
```

```
[root@localhost mapserver-5.2.1]# cp -f scalebar /usr/local/bin/scalebar
```

```
[root@localhost mapserver-5.2.1]# cp -f shp2img /usr/local/bin/shp2img
```

```
[root@localhost mapserver-5.2.1]# cp -f shp2pdf /usr/local/bin/shp2pdf
```

```
[root@localhost mapserver-5.2.1]# cp -f shptree /usr/local/bin/shptree
```

```
[root@localhost mapserver-5.2.1]# cp -f shptreetst  
/usr/local/bin/shptreetst
```

```
[root@localhost mapserver-5.2.1]# cp -f shptreevis  
/usr/local/bin/shptreevis
```

```
[root@localhost mapserver-5.2.1]# cp -f sortshp /usr/local/bin/sortshp
```

```
[root@localhost mapserver-5.2.1]# cp -f tile4ms /usr/local/bin/tile4ms
```

```
[root@localhost mapserver-5.2.1]# cp -f mapserv /var/www/cgi-  
bin/mapserv
```

```
[root@localhost mapserver-5.2.1]# cp -f
mapscript/php3/php_mapscript.so
/usr/lib64/php/modules/php_mapscript.so
```

```
[root@localhost mapserver-5.2.1]# yum install php-gd
```

Incluir librerías de 64 bits

```
[root@localhost mapserver-5.2.1]# vi /etc/ld.so.conf
```

Verificar y agregar

```
Include ld.so.conf.d/*.conf
/usr/local/include
/usr/local/lib
include /usr/lib64
include /usr/local/lib
include /usr/lib
```

Figura 28. Archivo de configuración ld.so.conf.

```
[root@localhost mapserver-5.2.1]# ldconfig
```

9. Pruebas

Editar el archivo

```
[root@localhost mapserver-5.2.1]# vi /var/www/html/index.php
```

Y escribir dentro del archivo

```
<?php
    dl("php_mapscript.so");
    phpinfo();
?>
```

Y verificar desde el navegador que Mapserver esté instalado.

<http://localhost/index.php>

10. Instalación de Fuentes Windows requeridos por mapserver

```
[root@localhost mapserver-5.2.1]# yum -y install rpm-build

[root@localhost mapserver-5.2.1]# cdpaquetes

[root@localhostpaquetes]# wget
http://dag.wieers.com/rpm/packages/cabextract/cabextract-1.2-
1.fc3.rf.x86_64.rpm

[root@localhostpaquetes]# rpm -i cabextract-1.2-1.fc3.rf.x86_64.rpm

[root@localhost paquetes]# wget
http://corefonts.sourceforge.net/msttcorefonts-2.0-1.spec

[root@localhostpaquetes]# rpmbuild -ba msttcorefonts-2.0-1.spec

[root@localhostpaquetes]# rpm -ivh
/usr/src/redhat/RPMS/noarch/msttcorefonts-2.0-1.noarch.rpm

[root@localhost paquetes]# servicexfsreload
```

Anexo 2: Exportación de las capas con gvSIG

1. En un nuevo proyecto añadimos todas las capas que deseamos exportar a la base de datos. Desde el menú Vista – AñadirCapa.

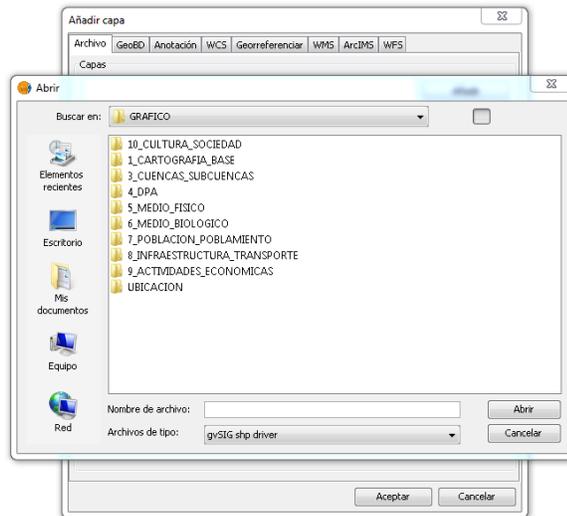


Figura 29. Añadir capas

2. Añadir una nueva conexión hacia la base de datos. Desde el menú Ver – Gestor de conexiones a BD espaciales, y llenar los campos requeridos para la conexión.

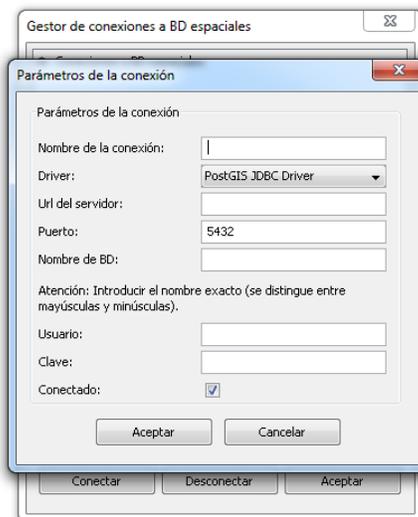


Figura 30. Añadir conexión a base de datos.

Una vez añadida la conexión. Pulsamos conectar.

3. Exportar todas las capas. Desde el menú Capa – Exportar a – PostGIS.

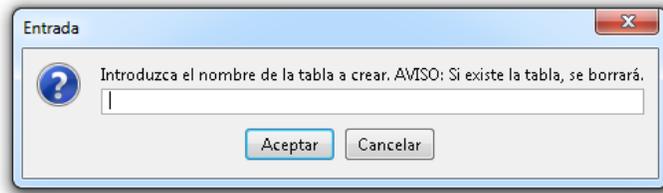


Figura 31. Exportación de capas.

Solo se necesita colocar el nombre de la tabla a crear.

Anexo 3: Publicación de las capas con gvSIG

4. En un nuevo proyecto añadimos todas las capas que están en la base de datos de PostreSQL que deseamos publicar en el servicio. Desde el menú Vista – Añadir Capa – pestaña GeoBD.

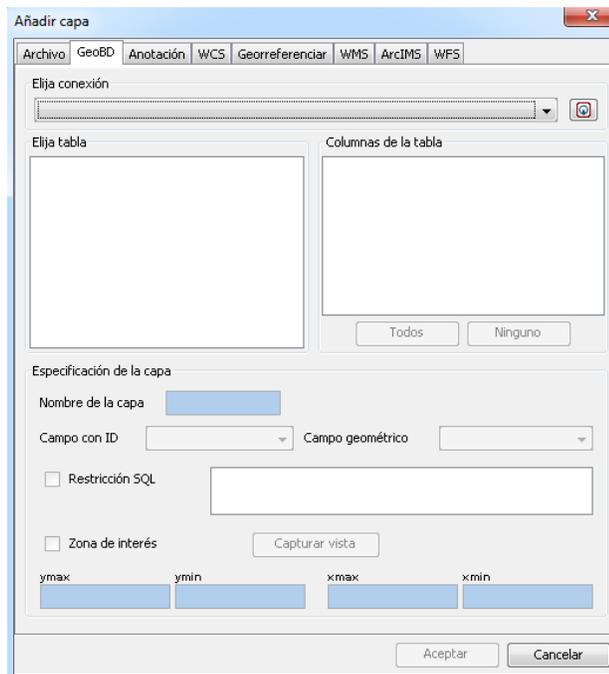


Figura 32. Añadir capas desde la base de datos

Seleccionamos la conexión de la base de datos de la cual queremos recuperar las capas; se en listaran todas las tablas disponibles en esa base de datos; seleccionamos y agregamos las que deseemos.

5. Es preciso colocar atributos visuales de acuerdo a la temática de la capa. La simbología puede ser formateada en las propiedades de cada capa. Desde el menú contextual de la capa – Propiedades – pestaña Simbología.
6. Una vez establecido el correcto estilo de visualización de las capas; se puede generar el mapfile, con todas las capas que están en la vista.

En la ventana Gestor de proyectos, se encuentra la extensión Publicación, que generará el archivo .map.



Figura 33. Herramienta de publicación

7. Pulsar en nuevo, para crear un nuevo documento de publicación, y abrirlo. Digitar el URL del servidor, seleccionar el servidor Mapserver y el tipo de servicio WMS.

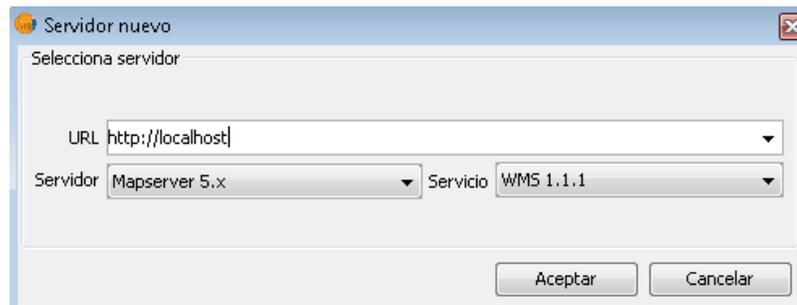


Figura 34. Herramienta de publicación

8. En la pestaña Servidor colar la ruta en donde se almacenará el mapfile generado.

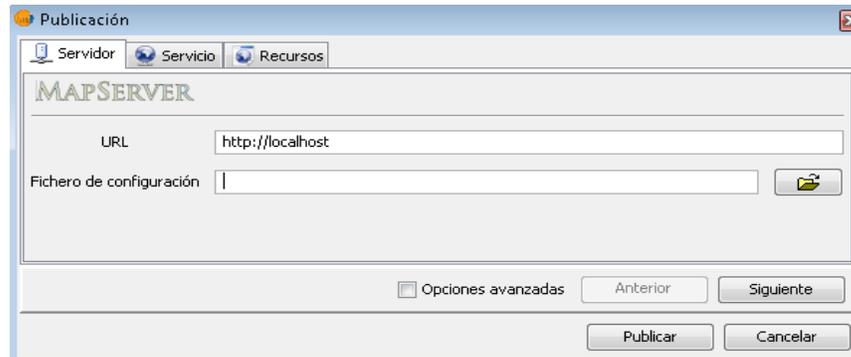


Figura 35. Pestaña Servidor

9. En la pestaña Servicio llenar los campos de Propiedades OGC principales. Mantener la URL del servicio por defecto.

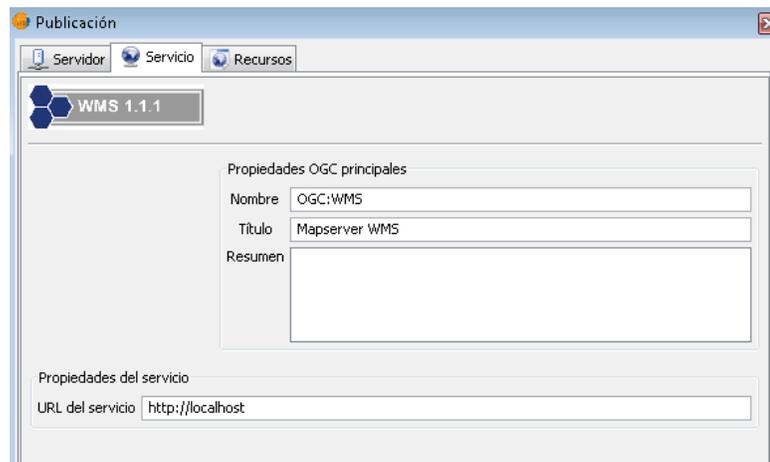


Figura 36. Pestaña Servicio

10. Finalmente en la pestaña Recursos seleccionar todas las capas que se desea publicar.

Anexo 4: Configuración del *mapfile atlas_azuary.map*

```
MAP
    NAME wms_atlas_provincias_del_azuary
    EXTENT 631266.9257 9592341.6046 794312.7613 9730880.9854
    SYMBOLSET "../common/symbols/symbols-pmapper.sym"
    FONTSET "../common/fonts/msfontset.txt"
    RESOLUTION 96
    IMAGETYPE png
    INTERLACE OFF

# =====
# Sección WEB
# =====

    WEB
        TEMPLATE "map.html"
        IMAGEPATH "/usr/local/apache2/htdocs/imgpMapper/"
        IMAGEURL "/imgpMapper/"
        METADATA
            "wms_title" "WMS Atlas de la Provincia del Azuay"
            "wms_abstract" "Servicio WMS del Atlas de la Provincia del Azuay.
Prototipo desarrollado por Johnatan Astudillo - Sebastian Nauta"
            "wms_srs" "EPSG:24877"
        END
    END

# =====
# PROYECCION
# =====

    PROJECTION
        "proj=utm"
        "zone=17"
        "south"
        "ellps=intl"
        "units=m"
        "no_defs"
    END
```

```

# =====
# Shape: poblacion_cantonal_2007
# =====
LAYER
NAME "densidad_poblacion_cantonal"
STATUS ON
TYPE POLYGON
DATA "the_geom from public.poblacion_cantonal_2007 using unique gid using
srid=24877"
CONNECTIONTYPE POSTGIS
CONNECTION "user=postgres password=1234 host=localhost port=5432
dbname=atlas_azuary"
MAXSCALE -1.0
MINSSCALE -1.0
TRANSPARENCY 75
SIZEUNITS pixels
TEMPLATE "void"
    PROJECTION
        "proj=utm"
        "zone=17"
        "south"
        "ellps=intl"
        "units=m"
        "no_defs"
    END
LABELITEM "canton"
CLASS
    EXPRESSION ([densidad_poblacional] >=0 AND [densidad_poblacional] <= 30)
    STYLE
        COLOR 250 207 189
    END #end style
    NAME "0 - 30"
END
CLASS
    EXPRESSION ([densidad_poblacional] >=31 AND [densidad_poblacional] <= 60)
    STYLE
        COLOR 254 237 204
    END #end style
    NAME "31 - 60"
END
CLASS

```

```

        EXPRESSION ([densidad_poblacional] >=61 AND [densidad_poblacional] <= 90)
        STYLE
            COLOR 238 243 208
        END #end style
        NAME "61 - 90"
    END
    CLASS
        EXPRESSION ([densidad_poblacional] >=90)
        STYLE
            COLOR 157 176 124
        END #end style
        NAME "> 90"
    END
METADATA
    "DESCRIPTION" "Densidad poblacional - habitantes por Km2"
    "ows_title" "poblacion_cantonal_2007"
    "RESULT_FIELDS" "canton, pobl,area_km, densidad_poblacional"
    "RESULT_HEADERS" "Cantón, Población, Área (Km2), Densidad"
    "LAYER_ENCODING" "UTF-8"
END
END # Layer
END # Map File

```

Anexo 5: Configuración del archivo .xml de *p.mapper*

```
<pmapper>
  <ini>
    <pmapper>
      <pmTitle>Atlas de la provincia del  Azuay</pmTitle>
      <debugLevel>3</debugLevel>
      <plugins>export</plugins>
      <plugins>scalebar</plugins>
      <plugins>transparency</plugins>
      <plugins>transparency2</plugins>
      <plugins>roundedboxes</plugins>
    </pmapper>
  <config>
    <pm_config_location>azuay</pm_config_location>
    <pm_javascript_location>javascript</pm_javascript_location>
    <pm_print_configfile>common/print.xml</pm_print_configfile>
    <pm_search_configfile>inline</pm_search_configfile>
  </config>
  <map>
    <mapFile>atlas_azuay.map</mapFile>
    <tplMapFile>common/template.map</tplMapFile>
    <categories>
      <category name="densidadpoblacional" description="Densidad Pobla. Por Cantón"
closed="true">
        <group>densidad_poblacion_cantonal</group>
      </category>
    </categories>
  </map>
</pmapper>
```

```

<allGroups>
    <group>densidad_poblacion_cantonal</group>
</allGroups>
<defGroups>
    <group>densidad_poblacion_cantonal</group>
</defGroups>
    <mutualDisableList>
    </mutualDisableList>
<layerAutoRefresh>1</layerAutoRefresh>
<imgFormat>png</imgFormat>
<altImgFormat>jpeg</altImgFormat>
<altImgFormatLayers>
    <group>densidad_poblacion_cantonal</group>
</altImgFormatLayers>
<sliderMax>531710</sliderMax>
<sliderMin>25000</sliderMin>
</map>
<query>
    <limitResult>300</limitResult>
    <highlightColor>0 255 255</highlightColor>
    <highlightSelected>1</highlightSelected>
    <autoZoom>nquery</autoZoom>
    <autoZoom>search</autoZoom>
    <zoomAll>search</zoomAll>
    <zoomAll>nquery</zoomAll>
    <infoWin>dynwin</infoWin>
    <alignQueryResults>1</alignQueryResults>

```

```
<pointBuffer>10000</pointBuffer>
<shapeQueryBuffer>0.02</shapeQueryBuffer>
</query>
<ui>
  <tocStyle>tree</tocStyle>
  <legendStyle>attached</legendStyle>
  <useCategories>1</useCategories>
  <catWithCheckbox>1</catWithCheckbox>
  <scaleLayers>1</scaleLayers>
  <icoW>18</icoW>
  <icoH>14</icoH>
</ui>
<locale>
  <defaultLanguage>es</defaultLanguage>
  <defaultCharset>UTF-8</defaultCharset>
  <map2unicode>1</map2unicode>
</locale>
<print>
  <printImgFormat>png</printImgFormat>
  <printAltImgFormat>jpeg</printAltImgFormat>
  <pdfres>2</pdfres>
</print>
<download>
  <dpiLevels>150</dpiLevels>
  <dpiLevels>200</dpiLevels>
  <dpiLevels>300</dpiLevels>
</download>
```

```

<php>
    <pearDbClass>MDB2</pearDbClass>

    <defaultTimeZone>Europe/Vienna</defaultTimeZone>

</php>

<pluginsConfig>
    <export>
        <formats>CSV</formats>

        <formats>PDF</formats>

        <PDF>
            <defaultFont>FreeSans</defaultFont>

            <defaultFontSize>9</defaultFontSize>

            <headerFontSize>9</headerFontSize>

            <headerFontStyle>BI</headerFontStyle>

            <layerFontSize>11</layerFontSize>

            <layerFontStyle>UB</layerFontStyle>

        </PDF>
    </export>

    <transparency2>
        <useOpacity>off</useOpacity>
    </transparency2>

</pluginsConfig>

</ini>

<searchlist version="1.0">
    <dataroot>${</dataroot>

    <searchitem name="nombre" description="Nombre Parroquia">

    <layer type="postgis" name="ant_1parroquias_azuary_odeplan_250k_utm_sam56">

    <field type="s" name="nombre" description="Nombre Parroquia" wildcard="0">

```

```
<definition type="suggest" connectiontype="db" sort="asc" firstoption="*">
  <dsn encoding="UTF-8">pgsql://smapas:ideuda2012@localhost/atlas_azuary</dsn>
  <sql>SELECT nombre FROM
public.ant_1parroquias_azuary_odeplan_250k_utm_sam56 WHERE nombre IS NOT NULL
ORDER BY nombre</sql>
  <dblayer encoding="UTF-8" keyfield="nombre" showfield="nombre"/>
</definition>
</field>
</layer>
</searchitem>
</searchlist>
</pmapper>
```

Anexo 6: Instalación y configuración de *geonetwork* en el sistema operativo *linux centos5.x*

1. Descargar el archivo *Java Development Kit 5.0 Update 14*, a través de este link:
<http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase5-419410.html>
2. Abrir una terminal y asignar permisos de ejecución a `jdk-1_5_0_14-linux-i586-rpm.bin`

```
[root@localhost Desktop]# chmod 777 jdk-1_5_0_14-linux-i586-rpm.bin
```

3. Luego Descomprimos y desempaquetamos

```
[root@localhost Desktop]# ./jdk-1_5_0_14-linux-i586-rpm.bin
```

Nos muestra los términos legales de *Sun*, pulsamos varias veces la barra espaciadora para llegar al final y escribimos “yes” y se instala el paquete de *JAVA*. Y verificamos que en la carpeta `/usr` se nos creó automáticamente una carpeta llamada `java` con los paquetes ya instalados.

4. Ahora, Descargamos el archivo *Apache-Tomcat 5.5.35.tar.gz*, a través de este link:
<http://tomcat.apache.org/download-55.cgi>; y escogemos el Core: `tar.gz`
5. Luego de tener el archivo descargado, lo descomprimos con el siguiente comando

```
[root@localhost Desktop]# gunzip apache-tomcat-5.5.35.tar.gz
```

Y posteriormente el siguiente comando

```
[root@localhost Desktop]# tarxvf apache-tomcat-5.5.35.tar.gz
```

Luego movemos la carpeta descomprimida al directorio `/usr/java` con el siguiente comando

```
[root@localhost Desktop]# mv apache-tomcat-5.5.35 /usr/tomcat
```

6. Ahora necesitamos un script para ejecutar tomcat como un demonio lo descargamos del siguiente link: <http://www.satollo.net/wp-content/uploads/2009/06/tomcat.zip> y lo ubicamos en /etc/init.d

7. Damos permisos a tomcat con los siguientes comandos

```
[root@localhost Desktop]#chownroot:root /etc/init.d/tomcat
```

Con el comando chown asigno un propietario que va a ser el superroot, con eso nos aseguramos que no puedan modificar el script de ejecución.

```
[root@localhost Desktop]# chmod +a+x /etc/init.d/tomcat
```

Como es usuario root solo él puede ejecutar, pero con el comando chmod damos permisos de ejecución a todos los usuarios.

8. Para instalar el script para distintos niveles de ejecución (runlevels)

```
[root@localhost Desktop]# /sbin/chkconfig --add tomcat
```

9. Con esto ya tenemos JAVA y Tomcat en nuestro Linux, ahora debemos incluir en el PATH de nuestro entorno de trabajo, las librerías JAVA y tomcat para esto editamos el archivo /etc/profile e incluimos las siguientes líneas antes de “unset i”, lo siguiente:

```
JAVA_HOME=/usr/java/jdk1.5.0_14
```

```
PATH=$PATH:$JAVA_HOME/bin
```

```
CATALINA_HOME=/usr/java/tomcat
```

```
Export JAVA_HOME CATALINA_HOME PATH
```

10. Iniciamos el servicio de tomcat

```
[root@localhost Desktop]#/usr/java/tomcat/bin/startup.sh
```

Ahora vamos al navegador y colocamos el siguiente url: <http://localhost:8080/> y nos debe mostrar lo siguiente:

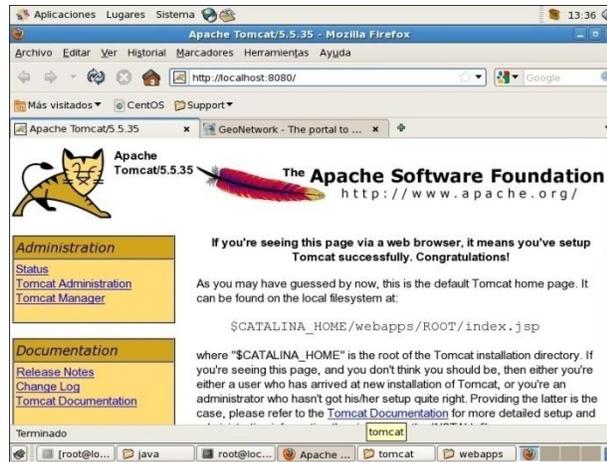


Figura 37. Pantalla inicial de Tomcat.

11. Por último Descargamos Geonetwork v2.6.4.war, a través de siguiente link:
http://sourceforge.net/projects/geonetwork/files/GeoNetwork_opensource/v2.6.4/

12. Descomprimos el archivo Geonetwork v2.6.4.war dentro de la carpeta
 /usr/java/tomcat/webapps/.

Renombramos la carpeta para que quede solo geonetwork.

13. Vamos al navegador y colocamos el siguiente url: <http://localhost:8080/geonetwork>
 y visualizara el entorno de geonetwork:

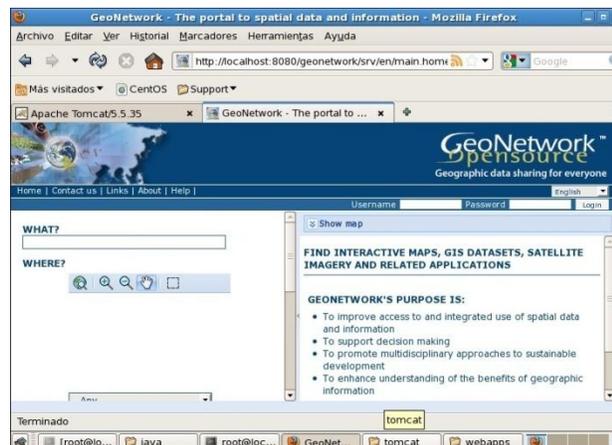


Figura 38. Pantalla inicial de Geonetwork.

El usuario por defecto es “admin” y la clave “admin”.

Anexo 7: Personalización de *geonetwork* configurando el archivo *config-gui.xml*

1. Para cambiar el mapa mundial se debe cambiar las siguientes líneas:

```
<mapSearch options="{projection: 'EPSG:4326', maxExtent: new OpenLayers.Bounds(-93,-6,-74,3), units: 'degrees', restrictedExtent: new OpenLayers.Bounds(-92,-5,-75,2)}">
  <layers>
    <layer server="http://geoportal.inpc.gob.ec:8080/geoinpc/wms"
      tocName="Qhapaqnan" params="{layers: 'inpc:qhapaq_nan_total',
      transparent: 'true', format: 'image/png'}" options="{ }" />
    <layer server="http://www.geoportalignm.gob.ec/nacional/wms"
      tocName="Provincias" params="{layers: 'igm:provincias', transparent:
      'true', format: 'image/png'}" options="{ }" />
    <layer server="http://gis.uazuay.edu.ec:8080/geoserver/crp/wms?"
      tocName="Cuenca del rioPaute" params="{layers:
      'crp:contorno_crp_50k_esf_wgs84', transparent: 'true', format:
      'image/png'}" options="{ }" />
    <layer server="http://geoserver.ambiente.gob.ec/conservacion/wms"
      tocName="Patrimonio Areas Conservacion" params="{layers:
      'mae:car_pane', format: 'image/jpeg'}" options="{isBaseLayer: true}" />
  </layers>
</mapSearch>
```

2. Para configurar el *mapviewer* se debe cambiar las siguiente líneas

```
<mapVieweroptions="{projection: 'EPSG:4326', maxExtent: new OpenLayers.Bounds(-93,-6,-74,3), units: 'degrees', restrictedExtent: new OpenLayers.Bounds(-92,-5,-75,2)}">
```

```

<layers>
  <layer server="http://geoportal.inpc.gob.ec:8080/geoinpc/wms"
  tocName="Qhapaqnan" params="{layers: 'inpc:qhapaq_nan_total',
  transparent: 'true', format: 'image/png'}" options="{ }" />
  <layer server="http://www.geoportaligm.gob.ec/nacional/wms"
  tocName="Provincias" params="{layers: 'igm:provincias', transparent:
  'true', format: 'image/png'}" options="{ }" />
  <layer server="http://gis.uazuay.edu.ec:8080/geoserver/crp/wms?"
  tocName="Cuenca del rioPaute" params="{layers:
  'crp:contorno_crp_50k_esf_wgs84', transparent: 'true', format:
  'image/png'}" options="{ }" />
  <layer server="http://geoserver.ambiente.gob.ec/conservacion/wms"
  tocName="Patrimonio Areas Conservacion" params="{layers:
  'mae:car_pane', format: 'image/jpeg'}" options="{isBaseLayer: true}" />
</layers>
<scales values="[7500000, 5000000, 2500000, 1000000, 750000, 500000,
250000, 100000, 75000, 50000, 25000, 10000, 7500, 5000, 2500, 1000]" />
<proj>
  <crs code="EPSG:4326" default="1" name="WGS84 (lat/lon)" />
</proj>
<servers>
<server name="NASA JPL OneEarth Web Mapping Server (WMS)"
url="http://wms.jpl.nasa.gov/wms.cgi?" />
  <server name="NASA Earth Observations (NEO) WMS"
  url="http://neowms.sci.gsfc.nasa.gov/wms/wms?" />
  <server name="DEMIS World Map Server"
  url="http://www2.demis.nl/mapserver/wms.asp?" />
  <server name="MAE"
  url="http://geoserver.ambiente.gob.ec/conservacion/wms?" />
  <server name="INPC" url="http://geoportal.inpc.gob.ec/geoinpc/wms?"
  />

```

```
<server name="CRP"  
url="http://gis.uazuay.edu.ec:8080/geoserver/crp/wms?" />  
</servers>  
</mapViewer>
```