

Universidad Nacional Autónoma de Nicaragua – León
UNAN – León

Facultad de Ciencias
Departamento de Computación



**“Instalación y Configuración de un Servidor de Mapas
utilizando Herramientas Libres y de Código Abierto”**

Trabajo Monográfico para optar al Título de Licenciado en Computación.

Autores:

- **Br. Donald Javier Caballero Sotelo.**
- **Br. José Ernesto Dávila Pantoja.**
- **Br. Luis Ernesto Díaz Beteta.**

Tutor:

- **Msc. Aldo René Martínez.**

León, Nicaragua
Junio del 2007

Índice de contenido

Agradecimientos.....	4
1 Introducción.....	5
2 Antecedentes.....	6
3 Justificación.....	7
4 Objetivos.....	8
4.1 Objetivo General.....	8
4.2 Objetivos Específicos.....	8
5 Marco Teórico.....	9
5.1 Los Sistemas de Información Geográfica (SIG).....	9
5.1.1 Aplicaciones de los Sistemas de Información Geográfica.....	10
5.1.2 ¿Qué es un Servidor de Mapas?.....	11
5.2 Introducción a las Proyecciones de Mapas.....	12
5.3 Sistema de Coordenadas Geográficas.....	12
5.3.1 Grados decimales versus grados minutos segundos	13
5.4 Proyecciones de Mapas: Aplanando el Esferoide.....	14
5.5 Sistema de Coordenadas Planas o Proyectadas.....	16
5.6 Introducción a los Mapas Digitales.....	17
5.6.1 El Poder de los Mapas Digitales.....	18
5.6.2 Las Dificultades de Hacer Mapas.....	19
5.6.2.1 Mapas Personales.....	19
5.6.2.2 Barreras Tecnológicas.....	20
5.6.3 Diferentes tipos de Web Mapping.....	21
5.6.3.1 Usuarios de Mapas Web.....	22
5.6.3.2 Tras la Página Web.....	23
5.6.3.3 Haciendo nuestro propio sitio de Web Mapping.....	24
5.6.4 Identificando los Tipos de Tareas para un Proyecto.....	25
5.6.4.1 Visualización y Creación de Mapas.....	26
5.6.4.2 Análisis.....	26
5.6.4.3 Creación y Manipulación.....	27
5.6.4.4 Conversión.....	27
5.6.4.5 Compartir.....	28
5.7 MapServer.....	28
5.7.1 Cómo Operan las Aplicaciones MapServer.....	29
5.7.2 Componentes Principales.....	30
5.7.2.1 Ejecutable de MapServer.....	31
5.7.2.2 Archivos de Mapas de MapServer.....	31
5.7.2.3 Fuentes de Datos.....	32
El modelo de datos raster.....	32
El modelo vectorial.....	32

Ventajas e inconvenientes de ambos modelos.....	33
5.7.2.4 Imagen del Mapa de Salida.....	34
5.8 Administración de Bases de Datos Espaciales.....	34
5.8.1 Introducción a PostGIS.....	35
5.8.2 SIG basados en Servidores.....	36
5.9 MapScript: El API de MapServer.....	38
6 Diseño Metodológico.....	39
6.1 Instalando MapServer.....	39
6.1.1 Plataformas.....	39
6.1.2 Adquirir los Binarios de MapServer.....	40
6.1.2.1 Binarios para Linux.....	40
6.1.2.2 Distribuciones SIG de Linux.....	40
6.1.2.3 Versiones de Debian Linux.....	41
6.1.3 Instalando MapServer en Ubuntu.....	41
6.2 Publicando Mapas Interactivos en la Web.....	42
6.2.1 MapServer en Linux con Apache.....	42
6.2.1.1 Ubicación del Directorio Temporal para Imágenes.....	43
6.3 Creando la Aplicación de Web Mapping.....	44
6.3.1 Creando el MapFile.....	44
6.3.2 Creando la Página de Inicialización.....	48
6.4 Instalación y Uso de PostGIS.....	49
6.4.1 Habilitando el acceso a PostgreSQL desde la red.....	52
6.4.2 Creando la base de datos geoespacial.....	54
6.4.3 Modificaciones en el mapfile.....	55
6.5 Creando Mapas con PHP-MapScript.....	56
6.5.1 Obtener la extensión de MapScript para PHP.....	56
7 Conclusiones.....	62
8 Recomendaciones.....	63
9 Bibliografía.....	64
Anexo A - Referencia de Objetos MapServer.....	65
Anexo B - Archivo de Configuración nicaragua.map.....	78
Anexo C - Archivo de Inicialización index.html.....	86
Anexo D - Plantilla global.html.....	87
Anexo E - PHP MapScript archivo index.php.....	89

Agradecimientos

A mi madre pilar fundamental y soporte incondicional a lo largo de toda mi vida, a mi padre que siempre dijo presente ante cualquier eventualidad y por supuesto al señor nuestro Dios que nunca me desamparó en mis sueños y ambiciones profesionales.

Donald Javier Caballero Sotelo.

A nuestro Padre Celestial por darnos la fuerza y constancia necesaria para realizar este trabajo monográfico. A mis padres por apoyarme económica y moralmente e impulsarme durante todos estos años de estudio.

José Ernesto Dávila Pantoja.

A Dios sobre todas las cosas, a toda mi familia, en especial a mis padres Luis Ernesto Díaz Blanco, Rosa Maria Beteta Medina y a mi esposa Karol Johana Argeñal por ser ese punto de apoyo que siempre necesite.

Luis Ernesto Díaz Beteta.

1 Introducción

Tanto para los novatos como para los expertos geoespaciales, las tecnologías de mapeo están transcurriendo por cambio tan significativo desde que por primera vez los mapas se hicieron digitalmente. La previa introducción de los Sistemas de Información Geográfica (SIG) y otras tecnologías de mapeo transformaron la manera tradicional de hacer los mapas e introdujeron una nueva era de especialistas en estas nuevas tecnologías geográficas. Hoy en día, un conjunto de avances tecnológicos aun más novedosos traen un cambio igualmente masivo a medida que el mapeo digital avanza. La disponibilidad de Sistemas de Posicionamiento Global (GPS), acceso a Internet de banda ancha, discos duros de almacenamiento masivo, dispositivos portátiles y - aun más importante - tecnologías Web están acelerando la posibilidad de incorporar información geográfica en nuestra vida diaria.

Paralelo a las excitantes oportunidades que las tecnologías modernas están proporcionando al universo geoespacial digital, un fenómeno menos conocido pero quizás mucho más importante ha emergido; un nuevo mundo de colaboración Open Source. Las comunidades de usuarios y el desarrollo Open Source, junto con un compromiso saludable por parte de la industria, están llenando la creciente necesidad y demanda de tecnologías espaciales para tomar mejores decisiones y proporcionar más información a las crecientes necesidades de mapeo de los usuarios de estas tecnologías. En un mundo multidimensional, la geografía forma un marco de trabajo común para diseminar información. La comunidad Open Source y la industria están llenando estas necesidades con un rango de crecimiento nunca alcanzado por la industria.

El diseño de mapas siempre ha resaltado las diferencias y variaciones que ocurren en el espacio; pero al mismo tiempo nos recuerda que compartimos este mundo con nuestros vecinos, y que nuestras acciones tienen un impacto más allá de nosotros mismos. Afortunadamente, las tecnologías de mapeo Web ayudan a llevar esta poderosa información a todos nosotros para nuestro bien futuro.

2 Antecedentes

El software libre y de código abierto ha sido un tema de investigación en varios trabajos monográficos, sin embargo, no existe ningún trabajo monográfico que abarque los Sistemas de Información Geográfica en general y particularmente los Sistemas de Información Geográfica libres y/o de código abierto.

En el Departamento de Computación de la Facultad de Ciencias de la UNAN-León podemos encontrar trabajos monográficos que tratan la configuración del Servidor Web Apache, la configuración del Servidor de Base de Datos PostgreSQL, se han desarrollado aplicaciones que combinan el uso de ambos servicios con el lenguaje PHP, pero nunca se había realizado un trabajo que combine todos estos componentes para configurar un servidor de Web Mapping.

3 Justificación

En el transcurrir de los años los Sistemas de Información Geográfica se han ido adaptando con éxito a los cambios tecnológicos, aunque siempre han mantenido su forma basada en la interacción entre imágenes y una base de datos.

Gracias a que las nuevas herramientas para el desarrollo de estos sistemas también han ido creciendo en calidad y robustez, lo que ha permitido desarrollar Sistemas de Información Geográfica cada vez más completos y sofisticados.

En la actualidad, la realidad indica que el mayor problema, más que la falta de información es la disponibilidad de ella, pudiéndose observar la falta de conocimiento de las personas en general sobre la ubicación de la información necesaria para realizar su trabajo y, por otro lado, una gran cantidad de información que se encuentra dormida en los estantes del anonimato, lo cuál se traduce en la pérdida del esfuerzo realizado por los gobiernos e instituciones por informar a la ciudadanía sobre temas de diversa índole.

Considerando que Internet ofrece muchas posibilidades para un acceso mejorado de los datos en tiempo real y un ambiente de trabajo común, que permite a usuarios compartir herramientas de imágenes, información y datos. Unido a la operatividad que tienen hoy en día los Sistemas de Información Geográfica, nace una nueva variante en la forma de presentar estos productos a los usuarios los "Servidores de Mapas". Los sistemas de información geográfica y los servidores de mapas de libre distribución son hasta el momento un campo inexplorado, desconociéndose sus potencialidades.

4 Objetivos

4.1 Objetivo General

Utilizar herramientas de libre distribución para instalar y configurar un servidor de Mapas a través de Internet.

4.2 Objetivos Específicos

- Instalar el servidor Web Apache con soporte a PHP y MapServer.
- Configurar MapServer para trabajar con PHP Mapscript.
- Instalar las librerías y utilerías PROJ 4 para proyecciones de mapas.
- Instalar las utilerías de línea de comandos OGR para soporte de datos tipo vector.
- Instalar la librería GDAL (Geospatial Data Abstraction Library) para soporte de imágenes raster.
- Instalar PostgreSQL y configurarlo para utilizar la extensión espacial PostGIS.
- Crear una aplicación de Web que utilice MapServer, Apache y PostGIS para presentar mapas de Nicaragua a través de la Web.

5 Marco Teórico

5.1 Los Sistemas de Información Geográfica (SIG)

Un SIG se define como un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos. Los SIG son una tecnología que permite gestionar y analizar la información espacial y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato.

Existen muchas otras definiciones de SIG, algunas de ellas acentúan su componente de base de datos, otras sus funcionalidades y otras enfatizan el hecho de ser una herramienta de apoyo a la toma de decisiones, pero todas coinciden en referirse a un SIG como un sistema integrado para trabajar con la infraestructura de un municipio, región o incluso a nivel de país.

A continuación se presentan un par de definiciones más académicas sobre lo que es un SIG:

“Un sistema de hardware, software y procedimientos diseñados para facilitar la obtención, gestión, manipulación, análisis, modelación y salida de datos espaciales referenciados, para resolver problemas complejos de planificación y gestión”, [National Center for Geographic Information and Análisis, N.C.G.I.A.]

“Un sistema de cómputo para obtener, almacenar, integrar, manipular, analizar y representar datos relativos a la superficie terrestre”, [Association for Geographic Information, A.G.I.]

De las definiciones anteriores se puede extraer que la importancia de los SIG radica en que las soluciones para muchos problemas frecuentemente requieren acceso a varios tipos de información que sólo pueden ser relacionadas por geografía o distribución espacial. Sólo la tecnología SIG permite almacenar y manipular información usando geografía para analizar patrones, relaciones y tendencias en la información, todo tendiente a contribuir a tomar mejores decisiones.

Toda la generación de nueva información que puede proveer un SIG depende significativamente de la información que posee la base de datos disponible. La calidad de esta base de datos y sus contenidos determina la cantidad y calidad de los resultados obtenidos del SIG.

5.1.1 Aplicaciones de los Sistemas de Información Geográfica

En la mayoría de los sectores los SIG pueden ser utilizados como una herramienta de ayuda a la gestión y toma de decisiones, a continuación se describen brevemente algunas de sus aplicaciones principales:

- *Cartografía automatizada:* Las entidades públicas han implementado este componente de los SIG en la construcción y mantenimiento de planos digitales de cartografía. Dichos planos son puestos a disposición de las empresas a las que puedan resultar de utilidad estos productos con la condición de que estas entidades se encargan posteriormente de proveer versiones actualizadas de manera periódica.
- *Infraestructura:* Algunos de los primeros sistemas SIG fueron utilizados por las empresas encargadas del desarrollo, mantenimiento y administración de redes de electricidad, gas, agua, teléfono, alcantarillado, etc., en este caso, los sistemas SIG almacenan información alfanumérica de servicios relacionados con las distintas representaciones gráficas de los mismos. Estos sistemas almacenan información relativa a la conectividad de los elementos representados gráficamente, con el fin de

realizar un análisis de redes.

- Cartografía digital 3D: Este tipo de información tridimensional de construcciones civiles, es requerida para realizar, por ejemplo, la planeación de la cobertura de las ondas de radio en una población ubicando los rebotes de ondas radiales entre antenas, optimización de redes, ubicación de antenas, interferencias de radio frecuencia, tendido de líneas de transmisión en 3D; o en el caso de la planeación de un aeropuerto este modelado tridimensional permitiría realizar el estudio de los espacios aéreos que intervienen en el proceso de diseño referenciado, en su caso, la viabilidad técnica de su construcción.

5.1.2 ¿Qué es un Servidor de Mapas?

Los servidores de mapas permiten a los usuarios la máxima interacción con la información geográfica. Por un lado el usuario o cliente accede a la información en su formato original, de manera que es posible realizar consultas tan complejas como las que haría un SIG. Un servidor de mapas funciona enviando, a petición del cliente, desde su navegador de Internet, una serie de páginas HTML (normalmente de contenido dinámico DHTML), con una cartografía asociada en un formato de imagen (por ejemplo, una imagen GIF o JPEG). Un servidor de mapas es, de hecho, un SIG a través de Internet.

Las primeras versiones de servidores de mapas sólo permitían realizar funciones básicas de visualización y consultas alfanuméricas simples. En las versiones más recientes es posible realizar funciones mucho más avanzadas. El tiempo dirá si los servidores de mapas tendrán toda la funcionalidad de los SIG.

El servidor de mapas es personalizable, es decir, se pueden preparar o programar las herramientas (los iconos de la aplicación) de manera que sean intuitivas para el usuario no experto en SIG.

5.2 Introducción a las Proyecciones de Mapas

La Tierra es redonda, o al menos así pensamos. Juegos de vídeo, globos y arte gráfico representan a la Tierra como una esfera perfecta, pero realmente la Tierra es un poco achatada. Por ello, llamamos a la Tierra Esferoide, en lugar de esfera. Tiene forma de esfera pero algo elíptica.

Para ir más allá, todos sabemos que la superficie de la Tierra no es perfectamente uniforme. Existen montañas y valles. Geoide es el término utilizado para referirse a un modelo más detallado de la forma de la Tierra. En cualquier punto del globo, el geoide puede ser más alto o más bajo que el esferoide.

Como podemos ver, cuando hablamos de la forma de la Tierra, es muy importante saber a cual forma nos estamos refiriendo. La forma de la Tierra es un factor clave cuando estamos produciendo mapas por que usualmente queremos referirnos a la posición más exactamente posible.

5.3 Sistema de Coordenadas Geográficas

El creciente uso de receptores Sistemas de Posicionamiento Global portátiles está ayudando al público en general a considerar el Sistema de Coordenadas Geográficas. Con un receptor GPS podemos obtener direcciones de navegación hacia una ubicación en particular utilizando un par simple de números llamados coordenadas. Algunas veces puede proveerse una elevación, proporcionando una descripción perfecta en tres dimensiones de la ubicación.

El Sistema de Coordenadas Geográficas está basado en un método para describir las ubicaciones utilizando una medida en grados de longitud y latitud. Estas describen una distancia específica del ecuador (0° norte / sur) y del meridiano de Greenwich (0° este / oeste).

Desde este punto de partida, la Tierra se divide en 360 trozos, desde el polo Norte hasta el polo Sur, conocidos como grados y representados por el símbolo "°". La mitad de estos trozos están al Este de 0° y la otra mitad está al Oeste. Nos referimos a ellos como Longitudes o Meridianos. Por lo tanto, los máximos valores son 180° de longitud oeste y 180° de longitud este.

Una variante del Sistema Básico de Coordenadas para las longitudes es útil en algunas partes del mundo. Por ejemplo, 180° de longitud oeste corren a través de las islas del pacífico sur. Los mapas de esta área pueden utilizar un sistema donde las longitudes inician el 0° y se incrementan hacia el este solamente, finalizando en la misma ubicación que también es conocida como 360°.

La Tierra también está dividida en Latitudes. Podemos imaginar estos como 180 trozos que atraviesan horizontalmente el globo. La mitad de estos trozos se encuentran al norte del ecuador y la otra mitad se encuentra al sur del ecuador. También nos referimos a ellos como Paralelos. Existen valores como 90° de latitud sur (en el polo sur) y 90° de latitud norte (en el polo norte).

5.3.1 Grados decimales versus grados minutos segundos

Las coordenadas globales pueden ser representadas utilizando notaciones algo diferentes entre sí. Es difícil decir si alguna es más común que la otra, pero algunas son ciertamente más intuitivas que otras. Una de las notaciones más tradicionales es conocida como Grados Minutos Segundos (DMS, por sus siglas en inglés). Las coordenadas son escritas como tres números separados, cada uno representa un grado, un minuto y un segundo respectivamente.

Por ejemplo, 120°W se lee como 122 grados Oeste. Raras veces los grados son lo suficientemente precisos para una ubicación; minutos y segundos son subdivisiones de un grado y proveen una mayor precisión. Cada minuto es un sexto de un grado. De la misma

manera, un segundo es un sexto de un minuto. Un ejemplo más preciso sería $122^{\circ} 30' 15'' W$, leído como 122 grados, 30 minutos y 15 segundos Oeste.

Del otro lado del espectro se encuentran la notación en grados decimales (DD, por sus siglas en inglés). Esta notación utiliza un espacio decimal en lugar de los minutos y segundos. Por ejemplo, $122.525^{\circ}W$ se lee como 122.525 grados Oeste. La posición decimal .525 representa a penas un poco más de la mitad de un grado.

Entre DMS y DD existen otras permutaciones. Una de las más comunes es tener un grado y un minuto, pero los segundos son un decimal de los minutos; por ejemplo, $122^{\circ} 30.25'$. Esta notación es difícil de usar. No puede llamarse DD o DMS, haciéndola difícil de explicar.

También es común ver diferentes signos utilizados para distinguir un valor de Este u Oeste. Por ejemplo, $122^{\circ}W$ puede también escribirse como -122° ; es negativo por que es menor que o está al Oeste del meridiano de Greenwich. La versión opuesta de él es 122° , se asume positivo por estar al Este de los cero grados.

5.4 Proyecciones de Mapas: Aplanando el Esferoide

El Sistema de Coordenadas Geográficas está designado para describir ubicaciones precisas en la Tierra con forma de esferoide. Los mapas impresos o los que se muestran en una pantalla, no tienen del todo forma esferoidal; por lo tanto presentan problemas para realizar descripciones útiles del mundo real. Aquí es donde las proyecciones de mapas entran en juego. Tratar de describir una Tierra redonda en una superficie plana no es algo nuevo, pero la tecnología para hacerlo ha mejorado sustancialmente con el advenimiento de la creación de mapas utilizando computadoras.

La ciencia de la proyección de mapas involucra tomar una ubicación en el Sistema de Coordenadas Geográficas y proyectarlo en una superficie plana.

Existen muchas clases diferentes de proyecciones de mapas. Algunas de las más simples son las siguientes:

- **Proyecciones Cilíndricas.** Estas implican envolver el globo con un plano en forma de cilindro. La parte superior e inferior del mapa se muestran distorsionadas.
- **Proyecciones Cónicas.** Estas lucen como un plano enrollado en forma de cono y ubicado como un sombrero en el globo. Si se sitúan en la parte superior del globo, el cono descansará en una latitud común alrededor del polo Norte. La parte más profunda del cono captura los detalles del polo, pero son más precisos en la latitud común. Cuando el papel se muestra plano, es equivalente a quitarle la piel a una naranja y mostrarla perfectamente plana.
- **Proyecciones Ortográficas.** Lucen como un mapa dibujado por un astronauta en órbita. Son difíciles de utilizar para mapa de escala global porque solo muestran una porción del globo. Sin embargo, si ignoramos la parte no visible, el mapa luce como un facsímil de la Tierra. Con el poder de procesamiento actual de las computadoras, dibujar mapas con forma de esfera es posible, pero esta perspectiva realista no siempre es útil.

Cada proyección tiene sus puntos fuertes y sus puntos débiles. Diferentes tipos de proyecciones se utilizan para diferentes aplicaciones. Por ejemplo, una proyección cilíndrica nos muestra un mapamundi comprensible pero no captura las características de los polos. De la misma manera una proyección cónica es ideal para el mapeo de los polos, pero imposible para mapas globales.

Aunque existen pocas clases principales de proyecciones, se han diseñado cientos de proyecciones. Los detalles de cómo se han implementado cada una varían grandemente. En algunos casos, solo necesitamos saber que proyección queremos. En otras ocasiones, necesitamos saber detalles como, dónde toca el plano el globo o en que punto está centrado el mapa.

5.5 Sistema de Coordenadas Planas o Proyectadas

Plano y Proyectado son términos que describen el Sistema de Coordenadas diseñado para una superficie plana. El Sistema de Coordenadas Geográficas utiliza medidas basadas en mundo esférico, o grados. Un Sistema de Coordenadas Proyectadas, está diseñado para mapas y utiliza un Plano Cartesiano para ubicar las coordenadas.

El Plano Cartesiano es un conjunto de dos (o más) ejes y ambos ejes se interceptan en un punto central. El punto central son las coordenadas 0,0. Los ejes utilizan coordenadas comunes y pueden variar de una proyección a otra. Por ejemplo, las unidades pueden estar en metros: las coordenadas (100,5) significan 100 metros al este y 5 metros al norte. Las coordenadas (-100,-5) significan 100 metros al oeste y 5 metros al sur.

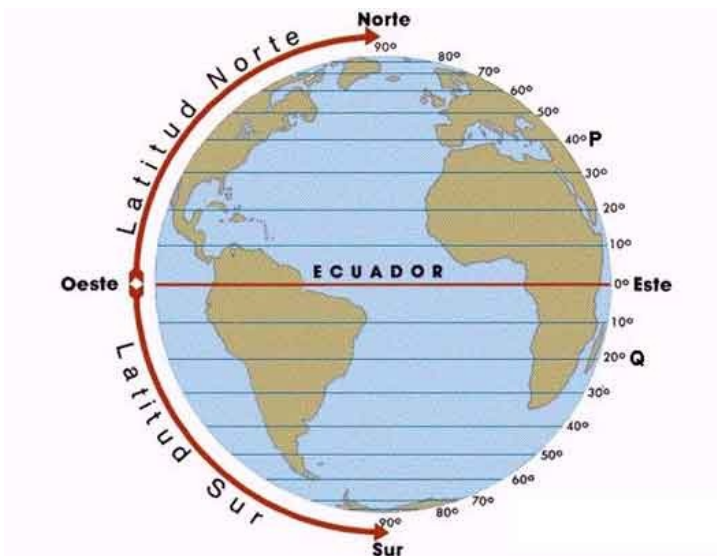


Ilustración 1: Diagrama que muestra las Latitudes

El eje Y hace referencia meridiano central. El meridiano central puede estar en cualquier lugar y depende de la proyección. Si la proyección está ubicada para Canadá, el meridiano central podría ser 100° oeste. Toda la coordenada proyectada este/oeste será relativa a este meridiano. Al contrario del Sistema de Coordenadas Geográficas donde el meridiano central

es el Meridiano de Greenwich, y donde todas las coordenadas este/oeste siempre son relativas a la misma longitud.

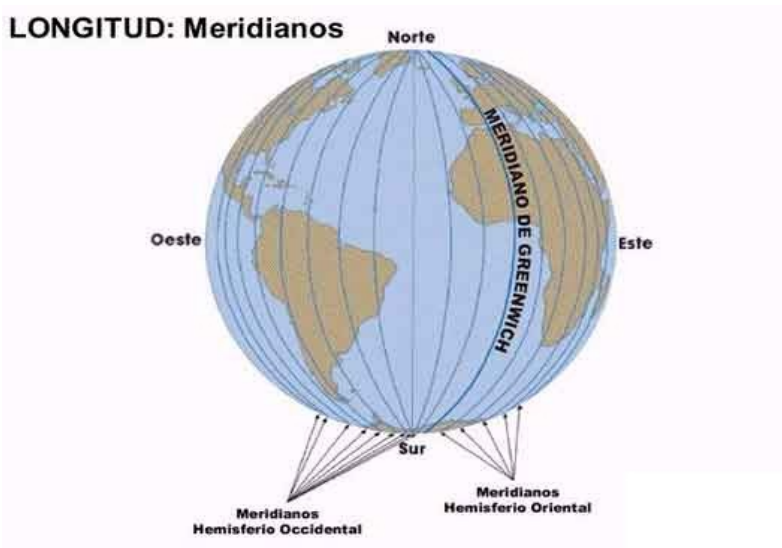


Ilustración 2: Diagrama que muestra las Longitudes

El eje X es perpendicular al eje Y, uniéndose en un ángulo recto en el punto 0,0. Este eje hace referencia a la latitud de origen. En el Sistema de Coordenadas Proyectadas, puede ser cualquier latitud. Siguiendo el ejemplo de Canadá, la latitud podría ser 45° norte. Toda coordenada proyectada norte/sur será relativa a esta latitud. En el Sistema de Coordenadas Geográficas, esta latitud es el ecuador, donde todas las coordenadas norte/sur siempre son relativas a el.

5.6 Introducción a los Mapas Digitales

No hace mucho tiempo atrás, la gente dibujaba y coloreaba sus mapas a mano. Analizar datos y crear los mapas resultantes era una tarea laboriosa y lenta. Los mapas digitales, gracias al siempre decreciente coste de poder de procesamiento y almacenamiento, han abierto un nuevo rango de posibilidades. Con el click del ratón o unas pocas líneas de código, nuestras computadoras analizan, dibujan y colorean mapas.

Por supuesto, aprender a producir mapas digitales requiere de cierto esfuerzo. Los datos de los mapas pueden ser utilizados de manera incorrecta, produciendo mapas incorrectos. Los mapas digitales no garantizan calidad o ética, al igual que los mapas convencionales.

5.6.1 El Poder de los Mapas Digitales

Cuando contrastamos los métodos para la creación de mapas convencionales y mapas digitales, el poder de los mapas digitales resulta evidente. El proceso para crear mapas convencionales incluye dibujos a mano de observaciones del mundo real transpuestas en papel. Si un feature cambia, se mueve, o es dibujado incorrectamente, se tiene que crear un nuevo mapa para reflejar ese cambio. Así mismo, si un mapa muestra las extensiones de una ciudad y éstas cambian, la extensión del mapa tendrá que cambiarse y el mapa tendrá que volver a hacerse en su totalidad.

Estos problemas se reducen con los mapas digitales. Gracias a que los features se almacenan como capas distintas en un archivo de computadora, podemos modificar un mapa sin tener que iniciar de cero. Una vez que un feature es modificado, el mapa basado en computadora refleja el cambio la siguiente vez que el feature es visualizado. Los mapas interactivos permiten a los usuarios ver precisamente el área que les interesa, en lugar de estar limitados por la dimensiones de una hoja de papel. El usuario también puede elegir el contenido a visualizar. El encargado de hacer los mapas no tiene que adivinar que información quiere el usuario pero puede hacer posible que el usuario pueda elegirla.

En lugar de enfocarse en los detalles de un área del mundo en particular, el creador de mapas digitales puede enfocarse en cual es la mejor manera de presentar información. Esto es similar a la diferencia entre un autor y un diseñador de páginas Web. Cuando nos movemos al plano digital, nos enfocamos en ayudar a los demás a encontrar información en lugar de hacer representaciones estáticas de información, como en una hoja impresa. Los

creadores de mapa, hoy día, son usualmente desarrolladores de sitios Web, programadores o algún tipo de analista de información geográfica. Su enfoque está en administrar y presentar información a una audiencia específica, ya sea finanzas, forestal o defensa nacional, por ejemplo.

5.6.2 Las Dificultades de Hacer Mapas

Usualmente es más sencillo describir un recorrido que dibujar un mapa. Quizás tengamos una percepción de cómo debe lucir un mapa y por ello nos atemoriza crear uno propio, pensarlo, en comparación, puede resultar muy sencillo. Aun así, un mapa dibujado por un amigo en una servilleta puede ser de muchísima más ayuda que un mapa profesional de una ciudad.

5.6.2.1 Mapas Personales

El elemento de conocimiento personal, en lugar del conocimiento general es lo que puede hacer que un mapa dudosamente útil sea una herramienta poderosa. Cuando las palabras no pueden describir la ubicación de algo que no es de conocimiento general, un mapa puede aclararnos el panorama. Un mapa puede utilizarse como un suplemento a la descripción verbal, pero debido a que crear un mapa conlleva dibujar una perspectiva que solo se encuentra en nuestra cabeza, puede ser una tarea muy intimidante. Esa intimidación y la carencia de apropiación sobre los mapas ha creado un dilema interesante. En nuestras mentes, los mapas son algo que los profesionales crean, no las personas promedio. Aun así, un mapa hecho en una hoja de papel puede tener mucho mayor significado para una persona que una mapa profesional de la misma zona. Entonces ¿De qué carecen los mapas profesionales? Generalmente muestran información común y usualmente carecen de información personal que podría hacer al mapa más útil e interesante para nosotros.

5.6.2.2 Barreras Tecnológicas

Los mapas digitales no son un tópico reciente. Desde que las computadoras pueden crear representaciones de la Tierra, la gente ha creado mapas con ellas. En los primeros tiempos de la computación, la gente solía dibujar mapas basados en textos ASCII. Sin embargo, diseñar gráficos con símbolos ASCII no era elegante. Afortunadamente, las tecnologías de gráficos más sofisticadas nos permiten crear nuestros propios mapas de alta calidad.

Es posible que actualmente hayamos creado mapas pero no estemos satisfechos con las herramientas utilizadas. Para algunos, el coste de herramientas comerciales puede ser prohibitivo. El software Open Source puede aliviar la necesidad de una inversión o pago inmediato.

Para otros, el coste no puede ser un problema pero las capacidades si. Al igual que el software propietarios, los productos open source para crear mapas varían en sus características. Características mejoradas pueden incluir facilidad de uso o calidad de salida . Una área con diferencias marcadas es la manera en que los productos se comunican con otros productos. Esto se llama interoperatividad y se refiere a la capacidad de un programa para compartir datos o funciones con otros programas. Estos habitualmente de adhieren a estándares abiertos - protocolos para comunicación entre aplicaciones. La idea básica es definir estándares que no dependan de un paquete de software en particular; los estándares deberán depender de los procesos de comunicación que los desarrolladores decidan implementar. Un ejemplo de éstos estándares en acción es la capacidad de un programa para solicitar mapas de otras aplicaciones de mapeo a través de internet. El verdadero poder de los estándares abiertos es evidente cuando nuestro programa se puede comunicar con un programa desarrollado por un grupo o vendedor diferente. Este es un problema crucial para muchas organizaciones grandes, especialmente agencias gubernamentales, donde compartir datos a través de sus departamentos puede hacer o quebrantar la eficiencia en la organización. Productos que utilicen estándares abiertos ayudarán a garantizar la disponibilidad a largo plazo de las aplicaciones que construyamos.

Otra barrera puede ser la falta de conocimiento técnico requerido para crear mapas digitales. Las técnicas para crear mapas digitales pueden ser prohibitivas para aquellos usuarios que no cuentan con los conocimientos técnicos necesarios. Esto se debe a que instalar y personalizar software está más allá del alcance de muchos usuarios de computadoras.

Las barreras tecnológicas existen, pero para aquellos que pueden utilizar computadoras y quieren hacer mapas con ellas, las posibilidades son infinitas. Las herramientas descritas no son necesariamente fáciles de utilizar: requieren cierto grado de habilidades técnicas. Programas para presentar mapas en el Web son más complicados que el software tradicional para escritorio. Existen usualmente procesos automáticos de instalación nada sencillos, y alguna configuración personalizada es requerida. Pero en general, una vez configuradas, las herramientas requieren una intervención mínima.

5.6.3 Diferentes tipos de Web Mapping

Una manera muy efectiva de tener disponible la información de un mapa para usuarios no tecnológicos es tenerla disponible a través de una página Web. Sitios de Web Mapping se han vuelto incrementalmente populares. Existen dos tipos de aplicaciones para Web Mapping: **estáticas** e **interactivas**.

Mapas estáticos mostrados como imágenes en una página Web son muy comunes. Si actualmente contamos con un mapa digital (por ejemplo: un mapa escaneado), podemos ponerlo a disposición fácilmente en nuestra página Web. Habilidades básicas de diseño Web es todo lo que necesitamos para esto pues solo estamos mostrando una imagen en una página.

Los mapas interactivos no son tan comunes debido a que se necesitan habilidades especializadas para mantener ese tipo de sitios activos. El término interactivo implica que la

persona que ve el mapa puede de alguna manera interactuar con el mapa. Esto puede significar seleccionar diferentes capas de datos del mapa para visualizar o realizar acercamientos a una zona en particular del mapa en la cual estemos interesados. Todo esto ocurre mientras interactuamos con una página Web y la imagen de un mapa que es actualizada repetidamente.

Los mapas interactivos que son accedidos a través de páginas Web usualmente son llamados mapas basados en Web o simplemente mapas Web. Estos mapas pueden ser muy poderosos, pero como hemos mencionado, también pueden ser difíciles de configurar debido a los conocimientos técnicos requeridos para administrar un servidor Web, un programa servidor de mapas y administrar la información subyacentes de los mapas. Como podemos ver, estos tipos de mapas son fundamentalmente diferentes a los mapas estáticos debido a que los mapas interactivos son realmente un tipo de aplicación basada en Web.

5.6.3.1 Usuarios de Mapas Web

Generalmente existen dos tipos de personas que utilizan mapas web: proveedores de servicios y usuarios finales.

Un usuario final puede ser una persona interesada en cierta información provista por el mapa web. Por ejemplo, puede ser una persona interesada en saber que tan cerca puede pasar un huracán por la casa de un amigo o puede ser alguien que necesita tener una idea de cuales clientes pueden ser afectados por un huracán.

Las necesidades de los usuarios finales pueden variar ampliamente. Podríamos buscar un sitio de Web mapping que provea direcciones para un lugar en particular. Alguien más podría querer ver una foto aérea y mapas topográficos para un futuro viaje a escalar. Algunos usuarios finales han creado un sitio de web mapping para llenar sus necesidades específicas, mientras otros solo buscan en Internet un sitio que tenga las capacidades que le interesan.

Los proveedores de servicios tiene propósitos complemente distintos para proporcionar un mapa a través del Web. Implementar un sitio de Web Mapping puede ser una excelente manera de hacer disponible cierta información que antes no lo estaba. Si una organización no está preparada para capacitar a un equipo de trabajo en aplicaciones SIG más tradicionales (que pueden tener una curva de aprendizaje muy grande), tener a un experto manteniendo un sitio de Web Mapping es un servicio muy valorado.

Otra razón por la cual un proveedor de servicios pueda crear un sitio de Web Mapping es para diseminar datos sin tener que transferir los datos a los clientes. Un buen ejemplo de esto es el gobierno provincial de Columbia Británica en Canadá. Actualmente cuentan con una excelente galería de fotos aéreas y mapas bases bien detallados, pero si deseamos los datos digitales, debemos negociar un acuerdo de intercambio de datos o comprarles los datos. La otra opción es usar uno de sus sitios de Web Mapping.

5.6.3.2 Tras la Página Web

Para algunas personas, los sitios de Web Mapping pueden parecer muy simples, mientras que para otras, estos sitios parecen casi mágicos. El funcionamiento interno de algunos sitios puede variar en dependencia del software utilizado, pero existe una serie de conceptos generales comunes:

- El servidor Web se encarga de las solicitudes de páginas Web y provee páginas con imágenes incluidas, etc. de regreso a los solicitantes.
- El servidor de Web Mapping acepta solicitudes transmitidas desde el servidor Web. La solicitud pregunta por un mapa con cierto contenido y para un área geográfica determinada. También puede hacer solicitudes para analizar o consultar resultados en forma tabular. El programa servidor de Web Mapping crea entonces las imágenes de los mapas solicitados (o los datos tabulados) y los envía de regreso al servidor Web para ser transmitidos al usuario final.

El servidor de Web Mapping necesita tener acceso a las fuentes de datos requeridos por la solicitud. Las fuentes de datos incluyen archivos ubicados en el mismo servidor o a través de una red interna. Si se utilizan estándares de Web Mapping, los datos también pueden venir de otro servidor de Web Mapping a través de una consulta en vivo.

5.6.3.3 Haciendo nuestro propio sitio de Web Mapping

Para darnos una idea general de los diferentes tipos de tecnologías involucradas para crear un sitio de Web Mapping, mostramos algunos de los requisitos básicos de un sitio de Web Mapping.

- **Una computadora.** Puede ser cualquier computadora, pero entre más compleja sea la aplicación de Web Mapping que deseamos almacenar, más poderosa debe ser la computadora. Los mapas más grandes y complejos tardan más tiempo en ser procesados; un procesador más rápido completa las solicitudes más rápidamente. Las opciones de hosting en Internet generalmente son demasiado simples para manejar sitios de Web Mapping dada la necesidad de acceso continuo al sistema operativo subyacente y al servidor Web. Servicios de hosting específicos para Web Mapping también están disponibles. El sistema operativo de la computadora puede ser una barrera para ejecutar algunas aplicaciones. En general, los sistemas operativos Windows y Linux tienen un mejor soporte, y Mac OS X u otros sistemas basados en Unix lo son en menor grado.
- **Una Conexión a Internet.** Es posible que tengamos un sitio de Web Mapping ejecutándose solo para nosotros o para una red interna, pero si queremos compartirlo públicamente, necesitamos una conexión de red de acceso público. Algunas cuentas de Internet personales limitan nuestra capacidad de ejecutar estos tipos de servicios, lo que nos hace necesitar cuentas de tipo comercial que pueden ser más costosas. El desempeño de un sitio de Web Mapping depende del ancho de banda de la conexión

a Internet.

- **Un Servidor Web.** Es necesario para manejar las comunicaciones de alto nivel entre los usuarios finales y los servicios subyacentes para la creación de mapas que se encuentran en nuestra computadora. El servidor Web presenta una página que contiene mapas y herramientas relacionadas con los mapas a los usuarios finales. Dicho servidor puede ser el Servidor HTTP [<http://httpd.apache.org>] o Microsoft Internet Information Server [<http://www.microsoft.com/WindowsServer2003iis/default.msp>]. Si utilizamos un Proveedor de Servicios de Internet para almacenar nuestro servidor Web, es posible que no tengamos acceso a las respectivas configuraciones de software subyacentes.
- **Un Servidor de Web Mapping.** Es un motor detrás del mapa que se muestra en la Web. El servidor de mapas o servidor de Web Mapping necesita configurarse para comunicarse con el servidor Web y ensamblar capas de datos en una imagen apropiada.
- **Mapping Data.** Un mapa es imposible de crear sin cierta información para mostrar. Dicha información pueden ser imágenes satelitales, conexiones a bases de datos, archivos de programas SIG, archivos de texto con listas de coordenadas de mapas u otros servidores de web mapping que se encuentren en Internet. Los datos de los mapas generalmente es conocida como datos espaciales o datos geoespaciales y pueden ser usados en cualquier tipo de programas de Mapping para escritorio o en servidores de Web Mapping.

5.6.4 Identificando los Tipos de Tareas para un Proyecto

Los creadores de mapas digitales tienen una variedad de herramientas para escoger y cada una de ellas está diseñada para cierto propósito. Muchas herramientas pueden hacer una o

dos tareas bien, y otras tareas moderadamente bien o no las realizan del todo. Existe cinco tipos de diferentes de herramientas utilizadas para crear mapas digitales y disciplinas relacionadas. Estas son categorías generales que usualmente se entrelazan.

5.6.4.1 Visualización y Creación de Mapas

Crear y visualizar mapas no son necesariamente la misma cosa. Algunas aplicaciones están hechas con la única intención de visualizar datos, mientras que otras apuntan a la producción de mapas. La producción de mapas está más orientada a productos de alta calidad visual con la intención de imprimirlos. Para el caso de Web Mapping, las herramientas de visualización son utilizadas para recopilar información de forma visual sobre los datos del mapa - cómo está dispuesta la información, al área geográfica que cubre los datos, compararla con otros datos, etc.

Las herramientas de creación de mapas son utilizadas para publicar datos en Internet a través de una aplicación de Web Mapping o servicios Web. También pueden ser utilizadas para imprimir un mapa en papel. Los conceptos de visualización y creación de mapas pueden agruparse ya que ambos implican un producto final gráfico. Tienden a ser el producto final después de la culminación de las actividades de las siguientes categorías.

5.6.4.2 Análisis

Solo visualizar mapas o imágenes no es usualmente el objetivo principal de un proyecto. Ciertos tipos de análisis son requeridos para hacer la visualización de datos más entendible y presentable. Esto incluye clasificación de datos (donde se agrupan features similares en categorías), cálculos de proximidad espacial (features a cierta distancia de otras), y sumariación estadística (agrupación de datos utilizando funciones como promedio o suma). El análisis tiende a recopilar información temporalmente, donde la manipulación de datos puede cambiar o crear nuevos datos.

5.6.4.3 Creación y Manipulación

Esta categoría puede incluir la creación de figuras , y usualmente utiliza un proceso llamado digitalización. Estas figuras pueden crearse como resultado de algún tipo de análisis.

Podemos manipular datos con una amplia variedad de herramientas, desde programas en línea de órdenes hasta manipulación gráfica utilizando drag and drop. Muchas aplicaciones de visualización no pueden editar figuras. Aquellas que pueden editar datos usualmente crean nuevos datos simplemente dibujando en la pantalla (digitalizando) o moviendo figuras. Algunos productos tienen la habilidad de hacer más, como análisis de traslapado o agrupación de figuras en menos piezas pero más grandes. Debido a que estas características son comunes en muchos productos comerciales, productos SIG opensource apenas están empezando a aparecer.

5.6.4.4 Conversión

Ciertas aplicaciones requieren que los datos estén en cierto formato de fichero o base de datos. Este es particularmente el caso en el mundo comercial, donde casi todos los vendedores soportan sus propios formatos propietarios marginando el soporte para otros formatos. El uso de estos formatos propietarios de datos ha llevado a una histórica dependencia a un producto de cierto fabricante. Afortunadamente, avances recientes en la industria del software geomático han llevado al soporte entre aplicaciones para más formatos competidores. Este giro conlleva a la interoperatividad entre estándares neutrales de fabricantes a través de organizaciones como el Open Geospatial Consortium (OGC).

Los datos de origen no siempre están en el formato requerido por las aplicaciones de visualización o manipulación. Si recibimos datos de alguien que utiliza un sistema de creación de mapas distinto, es casi seguro que será necesario convertir los datos. Los datos

de salida que podrían ser generados por los procesos de manipulación no siempre están en el formato que el usuario final o el cliente requieren. Aquí entra en juego el rol de las herramientas que convierten de un formato a otro.

Los programas de conversión de datos ayudan a hacer que los datos se encuentre disponibles en varios formatos. Existen disponibles algunas herramientas excelentes, y también existen librerías de soporte para aplicaciones, haciendo innecesaria la conversión de datos. Las librerías de acceso a datos permiten a las aplicaciones acceder directamente a los datos en lugar de convertirlos antes de utilizarlos en la aplicación.

5.6.4.5 Compartir

Existe cierto placer al crear y publicar un mapa de nuestra propia creación. Gracias a la variedad de herramientas libres que ahora disponibles, esto ya no es más que solo un sueño.

5.7 MapServer

Ya sea que estemos preparando imágenes de mapas estáticos para una página Web o publicar un sitio interactivo, MapServer puede hacer este trabajo. Los mapas interactivos permiten a los usuarios realizar acercamientos en áreas particulares y mostrar u ocultar capas del mapa. Se pueden crear aplicaciones que permitan al usuario seleccionar un feature en el mapa y vincularlo con otros datos. Las posibilidades son infinitas, pero el producto final usualmente será un proceso de apuntar, seleccionar y visualizar.

Podemos utilizar MapServer desde la línea de comandos o para usuarios interesados, a través de un API. MapServer también puede ser utilizado como una aplicación CGI (Common Gateway Interface) o usando scripts en lenguajes de programación Web comunes tales como PHP, Perl, Python y Java. Ya sea que usemos la versión CGI o creando nuestros propios

scripts, los archivos de configuración en tiempo de ejecución de MapServer controlan que capas se van a mostrar y cómo se van a dibujar. Datos para un mapa pueden añadirse a una aplicación existente de forma muy fácil editando el archivo de configuración.

5.7.1 Cómo Operan las Aplicaciones MapServer

MapServer generalmente trabaja detrás de un servidor Web. El servidor Web recibe las solicitudes de mapas y se las pasa a MapServer. MapServer genera la imagen del mapa solicitado y se la envía al servidor Web, que la transmite de regreso al usuario.

La función principal de MapServer es leer datos de distintas fuentes y unir estas capas de datos en un archivo gráfico, también conocido como imagen del mapa. Una capa puede ser una imagen satelital, otra puede ser un delineado de los límites de nuestro país o puntos que muestran las principales ciudades. Cada capa es superpuesta o dibujada encima de las otras y luego es imprimida en un gráfico amigable al Web para que los usuarios la vean.

El proceso de dibujo (renderizado) ocurre cada vez que se realiza una solicitud por un nuevo mapa a MapServer, por ejemplo, cuando un usuario realiza un acercamiento en un mapa para una vista más cercana. Este proceso también ocurre cuando un usuario solicita manualmente el redibujado, tal como cuando el contenido de una capa de datos ha cambiando, y el usuario quiere que dicho cambios se visualicen.

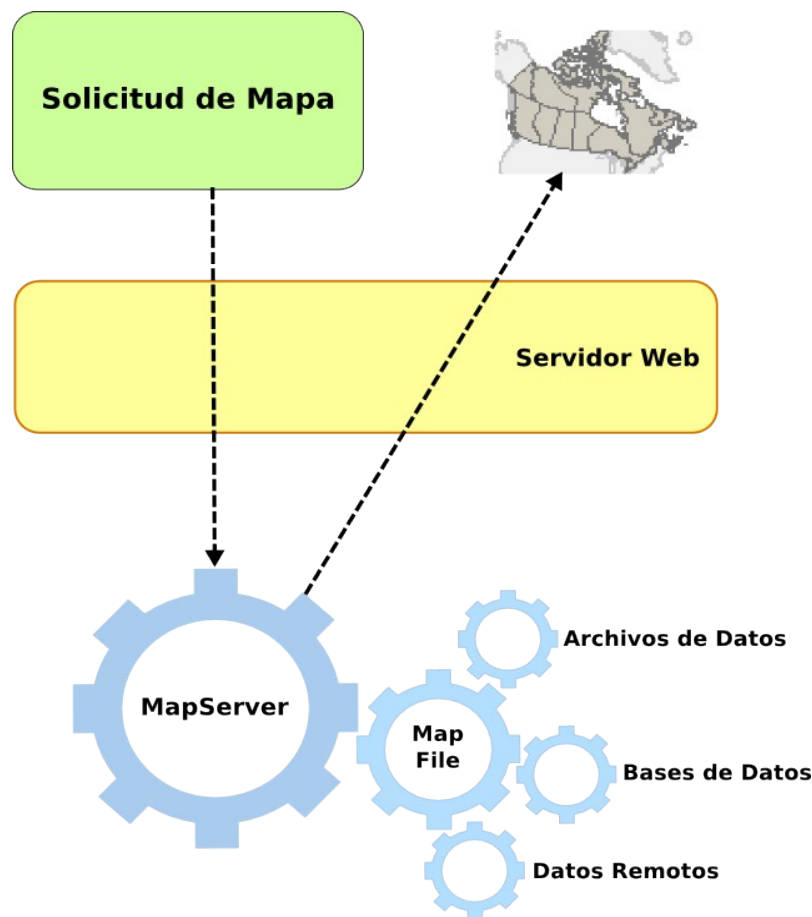


Ilustración 3: Operación básica de una aplicación MapServer

5.7.2 Componentes Principales

MapServer produce archivos gráficos basado en la solicitud realizada por el usuario y cómo esté definido el mapa. Los componentes principales incluyen el programa ejecutable de MapServer o programa CGI, un archivo de mapas (map file), las fuentes de datos y las imágenes generadas. Después que un usuario hace una solicitud, el programa CGI de MapServer accede al archivo de mapa, dibuja la información desde las fuentes de datos, y devuelve una imagen del mapa.

5.7.2.1 Ejecutable de MapServer

MapServer se ejecuta en su versión más simple, como una aplicación CGI ejecutable en un servidor Web. Técnicamente, MapServer es considerado un proceso sin estados basado en HTTP. Sin estados significa que procesa una solicitud y luego detiene su ejecución. Una aplicación CGI recibe solicitudes de un servidor Web, las procesa y luego regresa una respuesta o datos al servidor Web. CGI es por mucho el más popular debido a su simplicidad: no se requiere programación alguna para echarlo a andar.

El ejecutable CGI de MapServer actúa como intermediario entre los archivos de datos del mapa y el servidor Web que solicita el mapa. Las solicitudes se pasan como parámetros CGI del servidor Web a MapServer. Las imágenes creadas por MapServer son retornadas al servidor Web y finalmente al navegador Web del usuario.

5.7.2.2 Archivos de Mapas de MapServer

MapServer es como un motor que requiere de combustible para poder andar, así como de un sistema de distribución de combustible que lleve el combustible al motor. El ejecutable de MapServer necesita saber que capas del mapa debe dibujar, cómo debe dibujarlas y donde se encuentran ubicadas las fuentes de datos. Los datos son el combustible, y el archivo de mapas - también conocido como archivo .map - funciona como el sistema de distribución. El archivo de mapas es un archivo de configuración en texto plano que lista las configuraciones para dibujar en interactuar con el mapa. Incluye información sobre las capas de datos a dibujar, donde se encuentra el foco geográfico del mapa, que sistema de proyección se está utilizando y que formato de imagen de salida usar, y establece la manera en que se dibujarán la leyenda y la barra de escala.

Cuando llega una solicitud a una aplicación MapServer, la solicitud debe especificar que

archivo de mapas usar. Entonces MapServer crea el mapa basándose en las configuraciones que se encuentran en el archivo de mapas. Esto hace que el archivo de mapas sea la pieza central de cualquier aplicación MapServer.

5.7.2.3 Fuentes de Datos

MapServer puede utilizar un conjunto amplio de fuentes de datos para crear mapas. El soporte out-of-the-box cubre los formatos más comunes. Agregados de acceso a datos opcionales proporcionan acceso a docenas de formatos tipo *vector* y *raster* (formatos soportados por las librerías GDAL y OGR). Las fuentes de datos pueden ser archivos de datos SIG, conexiones a bases de datos e incluso archivos de texto separados por coma que usan las capacidades de la librería OGR para el formato de Datos Espaciales Virtuales.

El modelo de datos raster

El modelo raster centra su interés más en las propiedades del espacio que en la presentación precisa de los elementos que lo conforman. Para ello estructura el espacio en una serie de elementos discretos por medio de una retícula regular, generalmente compuesta por celdas cuadradas, también llamadas 'pixels'. Cada una de esas celdas se considera como indivisible y es identificable por su número de fila y columna.

El modelo vectorial

El modelo vectorial representa cada objeto geográfico de forma independiente mediante las primitivas gráficas (puntos, líneas y polígonos), codificando explícitamente el límite que los separa del entorno. Las primitivas están caracterizadas de la siguiente forma:

- Los elementos puntuales se representan mediante un par de coordenadas 'x,y' que definen la posición del punto.

- Los elementos lineales están formados por uno o más segmentos lineales que se unen en vértices mediante coordenadas 'x,y'.
- Los elementos superficiales se presentan mediante las coordenadas 'x,y' de ellos vértices de las líneas que forman un perímetro.

Ventajas e inconvenientes de ambos modelos

Ventajas del modelo vectorial:

- ◆ Buena presentación y resolución.
- ◆ Menor ocupación, mayor velocidad de proceso.
- ◆ Facilidad de descripción y aplicación topológica.

Inconvenientes del modelo vectorial:

- ◆ Estructura de datos compleja.
- ◆ Programas de tratamiento más complejo.

Ventajas del modelo raster:

- ◆ Captura de información más rápida.
- ◆ Estructura de datos más simple.
- ◆ Facilidad de análisis espacial y simulación.
- ◆ Tecnología más económica.

Inconvenientes del modelo raster:

- ◆ Gran volumen de datos.
- ◆ Menos resolución y dificultad de estructura en capas de información.

MapServer también puede utilizar las especificaciones Web OGC para acceder y compartir datos a través de Internet. Capas de datos puede ser solicitadas a servidores remotos que también utilicen las especificaciones OGC.

5.7.2.4 Imagen del Mapa de Salida

Mientras el archivo de mapa es la pieza central de cualquier aplicación MapServer, la imagen del mapa generada es usualmente lo que el usuario busca. Después que todas las capas son procesadas y escritas en un archivo de gráficos amigables para la Web, el navegador Web del usuarios es orientado para cargar la imagen en una página Web para su visualización.

El mapa no es la única imagen que se puede crear. Barras de escala, leyendas gráficas y mapas de referencia también pueden formar parte de una aplicación MapServer. Estos se manejan en forma similar a una imagen de mapa.

5.8 Administración de Bases de Datos Espaciales

Una base de datos es una herramienta para almacenar y acceder tablas de información. Tradicionalmente las bases de datos almacenan información en campos y registros (columnas y filas o atributos y valores). El tipo de datos que puede almacenar un campo varía según los diferentes tipos de bases de datos, pero generalmente hablando, almacenan datos numéricos y de texto. La principal característica de una base de datos son las consultas, a través de ellas podemos obtener la información que cumpla con los criterios especificados. Las bases de datos relacionales nos permiten unir (join) información almacenada en diferentes tablas utilizando una pieza común de información que se encuentra disponible en ambas tablas.

Una base de datos espacial es una base de datos relacional capaz de almacenar datos geográficos. Distintas bases de datos y productos SIG utilizan el término "Base de datos Espacial" para definir términos ligeramente distintos. Por ejemplo, el motor de base de datos espaciales (SDE, por sus siglas en inglés) de ESRI no es una base de datos espacial, pero si es una interfaz entre un software cliente y una base de datos normal. Permite almacenar

datos espaciales en formatos SDE en la base de datos. Para cargar y manipular los datos espaciales, se requiere tener un producto ESRI o acceder a un servicio de ESRI.

5.8.1 Introducción a PostGIS

PostGIS es una extensión espacial (o geográfica) para la popular base de datos relacional de fuente abierta PostgreSQL.

PostGIS tiene muchas características que la mayoría de bases de datos comerciales no tienen, muchas de las cuales hicieron que la gente considerara a PostGIS como un sistema empresarial de administración de base de datos espaciales.

PostGIS puede considerarse una base de datos espaciales avanzada por que tiene la habilidad tanto de almacenar como de manipular datos espaciales. PostGIS no es un simple repositorio de almacenamiento, también es un ambiente para interactuar con datos espaciales. La OGC ha creado una especificación para el almacenamiento y consulta de datos espaciales en una base de datos SQL llamada Simple Features Specification for SQL (SFSQL). Las especificaciones de la OGC se están convirtiendo en un requisito integral para la interoperatividad entre datos geoespaciales.

PostGIS tiene una de las más robustas implementaciones de las especificaciones SFSQL, según el sitio oficial de la [<http://www.opengeospatial.org/resources/?page=products>] Dado que PostGIS implementa todas las especificaciones SFSQL, es posibles acceder a funciones estandarizas sin giros propietarios que puedan causar una pérdida de tiempo en ciclo de desarrollo de nuestros proyectos.

La manera en que la información geográfica es presentada en PostGIS es muy intuitiva. Es posible acceder a coordenadas de características espaciales a través de una variedad de

métodos, en dependencia de nuestras necesidades. Podemos realizar consultas a través de una herramienta de línea de comandos o visualizarla gráficamente con un software de mapeo.

5.8.2 SIG basados en Servidores

PostgreSQL es un servidor de bases de datos relacionales. Cuando se realizan consultas a la base de datos, el servidor procesa las consultas, prepara los datos, y devuelve los resultados a nuestra aplicación. Todo el trabajo pesado es realizado por el servidor y solo los resultados nos son devueltos. Aun con los caballos de fuerza que tienen las computadoras modernas, la mayoría de las PC's no está diseñadas para manejar el trabajo pesado de procesar consultas a bases de datos. Si todos los datos tuvieran que enviarse a través de una red para ser procesados por nuestra aplicación en el cliente, la red y la aplicación cliente formarían un cuello de botella en el rendimiento.

El mismo problema existe con los Sistemas de Información Geográfica y la administración de datos espaciales. Muchas, si no todas, aplicaciones SIG para escritorio tienen una fuerte repercusión en las computadoras de los usuarios. Esto puede estar bien para procesos normales de mapeo (aunque algunas características complejas puede ser muy lentas), pero cuando requerimos realizar análisis espaciales avanzados, pueden aparecer problemas.

Otro problema es que el programa SIG de escritorio puede no tener la capacidad para realizar el análisis que necesitamos. En este caso, tenemos la opción de comprar un componente adicional que se integre a nuestro programa SIG o utilizar otra aplicación para procesar los datos. Igualmente, existe una sobrecarga de trabajo al copiar y procesar los datos en una PC, y usualmente nos vemos obligados a convertir los datos a otros formatos. PostGIS aprovecha las ventajas de las bases de datos del lado del servidor, poniendo a disposición del servidor un conjunto extenso de funciones SIG. PostGIS incluye un almacén

de datos espaciales y también la capacidad de manipulación de datos espaciales usualmente encontrados solo en aplicaciones de SIG de escritorio. Esta característica clave de PostGIS reduce significativamente los requisitos de las aplicaciones clientes al tomar ventaja de las capacidades de los servidores.

PostGIS es una extensión para el servidor de base de datos PostgreSQL, de manera que una instalación estándar de PostgreSQL es la base para la instalación de PostGIS. No se requiere una versión de PostgreSQL compilada a la medida para utilizar PostGIS. PostGIS consta de tres componentes:

- **Librerías de PostGIS:** la librería principal es libpostgis.so. Esta librería es la interfaz entre las capacidades de PostgreSQL y las habilidades espaciales de PostGIS.
- **Script de PostGIS para funciones y tipos:** existe un script principal que carga los cientos de funciones y tipos de PostGIS: `postgis.sql`. Versiones recientes de PostGIS no tiene un archivo `postgis.sql`; en su lugar utilizan un archivo llamado: `lwpostgis.sql`.
- **Script opcional de soporte a proyecciones:** usualmente se carga un script opcional llamado `spatial_ref_sys.sql` que nos permite utilizar sistemas de referencia espaciales o proyecciones con datos de PostGIS.

Los scripts son independientes de la plataforma, de manera que la tarea difícil es obtener las librerías necesarias. Posiblemente queramos compilar nuestra propia versión de PostGIS para incluir otras características, pero con las nuevas versiones esto se está volviendo innecesario.

5.9 MapScript: El API de MapServer

MapServer usualmente es utilizado como una aplicación CGI que se coordina con un servidor Web. Para obtener mayor poder y flexibilidad podemos utilizar el API de MapServer con uno de varios lenguajes de programación, incluyendo PHP, Perl, Python, Ruby, Java, C# y más. MapScript provee métodos para interactuar con datos geoespaciales, estilos cartográficos de los mapas generados y la creación de las imágenes finales del mapa.

6 Diseño Metodológico

6.1 Instalando MapServer

MapServer se ejecuta en múltiples plataformas y los detalles de su instalación varían en dependencia de donde queramos ejecutarlo y cómo queremos integrarlo con el resto de nuestro sistema.

6.1.1 Plataformas

Muchos sistemas operativos y procesadores pueden ejecutar MapServer satisfactoriamente. Dentro de los sistemas operativos que ejecutan MapServer se encuentran los siguientes:

- Windows 2003, 2000, XP, 95.
- RedHat / SuSE / Debian / Mandrake / Ubuntu Linux.
- Solaris.
- Mac OS X Panther.
- VMWare ejecutando Windows y Linux.
- FreeBSD.
- SCO Open Server.
- SGI IRIX.

La velocidad mínima de procesador es tan baja como 120 MHz. y 64MB de memoria. Con la diversidad y flexibilidad requerida para cumplir con estos requisitos multiplataforma, muchos desarrolladores han encontrado en MapServer la única opción para alojar mapas basados en Web.

6.1.2 Adquirir los Binarios de MapServer

Muchos documentos y how-tos incluyen información sobre compilar MapServer, lo cual puede llevarnos a asumir que es necesario compilar MapServer desde el código fuente. Para la mayoría de los usuarios, esto no es requerido. Adquirir los binarios se refiere al proceso de descargar los archivos ejecutables y librerías listos para ejecutarse en el sistema operativo de nuestra elección, sin compilar el código fuente.

6.1.2.1 Binarios para Linux

Un proyecto llamado Free Open Source Software GIS Suite [<http://maptools.org/fgs/>] ofrece un mecanismo de paquetería para todos los binarios, librerías y otros archivos necesarios para ejecutar aplicaciones FGS. La versión inicial instala un ambiente mínimo para Apache, PHP y MapServer.

6.1.2.2 Distribuciones SIG de Linux

Existen por lo menos cuatro distribuciones de Linux que incluyen MapServer y aplicaciones relacionadas.

- DebianGIS Esta distribución basada en Debian incluye varias aplicaciones SIG. [<http://pkg-grass.alioth.debian.org/cgi-bin/wiki.pl>].
- HostGIS Esta distribución incluye MapServer, PostGIS, PHP, Python, Perl y muchas más aplicaciones GNU/Linux estándar. Está basada en Slackware Linux. [<http://hostgis.com/linux.html>].

- STARCD Esta distribución de CD "en Vivo" se ejecuta desde un CD-ROM arrancable. Incluye MapServer, QGIS y otras aplicaciones SIG incluido GRASS GIS. Esta basada en Mandrake Linux. [<http://rslultra.star.ait.ac.th/~yann/starcd/>].
- GIS-Knoppix Esta distribución de CD arrancable incluye un gran número de paquetes importantes como MapServer, PostGIS, Terraview, JUMP, QGIS, Thuban, Maplab, etc. [<http://www.sourcepole.com/sources/software/gis-knoppix/>].

6.1.2.3 Versiones de Debian Linux

Existen varios repositorios de paquetes para Debian Linux con MapServer disponible. Varían en dependencias y características. Un repositorio se encuentra en [<http://agrogeomatic.educagri.fr/debian/>].

6.1.3 Instalando MapServer en Ubuntu

El método típico de instalación de MapServer es copiar el fichero mapserv en el directorio cgi-bin del servidor Web. Si el fichero se encuentra en un directorio desde el cual el servidor Web pueda ejecutarlo, será suficiente para instalar la versión CGI de MapServer.

Los paquetes DEB de MapServer incluidos en los repositorios de Ubuntu, nos facilitan el proceso de instalación. Para instalar MapServer con soporte a PHP MapScript ejecute la siguiente orden:

```
$ sudo apt-get install cgi-mapserver mapserver-bin mapserver-doc php5-mapscript
```

Una vez finalizada la instalación el fichero ejecutable de MapServer se encontrará en el directorio `/usr/lib/cgi-bin/` si estamos ejecutando Apache2.

Una prueba básica de MapServer es ejecutar el programa `mapserv` desde la línea de órdenes. Abrimos una terminal y ejecutamos el siguiente comando:

```
$ /usr/lib/cgi-bin/mapserv
```

Debemos obtener una salida parecida a esta:

```
This script can only be used to decode form results and should be
initiated as a CGI process via a httpd server.
```

6.2 Publicando Mapas Interactivos en la Web

6.2.1 MapServer en Linux con Apache

Configurar Apache para hacer uso de MapServer no es una tarea complicada. Basta con copiar el programa `mapserv` en la ubicación `cgi-bin` predeterminada del servidor Web. Hay otras configuraciones que debemos realizar para poder acceder a los mapas producidos por MapServer. Cuando MapServer se ejecuta, necesita de un área temporal para crear las imágenes de los mapas, leyendas, barras de escala, etc. Este directorio temporal debe también ser accesible a través de la Web, de tal manera que los usuarios puedan ver la imágenes creadas por MapServer.

6.2.1.1 Ubicación del Directorio Temporal para Imágenes

Podemos configurar nuestro directorio temporal en cualquier parte de nuestro sistema de ficheros. Para mantener una configuración simple, vamos a crearlo directamente dentro del directorio raíz htdocs. Lo hacemos así para no crear un alias para Apache, ya que el directorio es un directorio hijo del directorio raíz del servidor Web.

Por ejemplo, si el directorio raíz del servidor Web se encuentra ubicado en el directorio `/var/www`. Podemos crear nuestro directorio temporal de imágenes en el directorio `/var/www/ms_tmp` de manera que MapServer será capaz de encontrar dicho directorio cuando se le configure para ubicarlo en `http://localhost/ms_tmp`.

MapServer deberá tener los permisos apropiados sobre el directorio `ms_tmp` para poder grabar las imágenes. Para ello, debemos asignar al usuario y grupo del servidor Web como dueño del directorio. Podemos ejecutar el siguiente comando para realizar esta tarea:

```
$ sudo chown www-data /var/www/ms_tmp
```

Esto hará que el directorio sea accesible al servidor Web y por ende a MapServer. Este proceso puede variar en otras plataformas, dependiendo de cómo estemos usando nuestro servidor web, pero los conceptos son los mismos.



Ilustración 4: Prueba de MapServer con Apache y Linux

6.3 Creando la Aplicación de Web Mapping

Para demostrar las capacidades de presentación de mapas a través de la Web, que posee MapServer, creamos una aplicación web que hace uso de Apache y MapServer para mostrar un mapa de Nicaragua en una página Web. Para ello, creamos tres archivos que conforman la aplicación Web:

- El archivo de mapa o mapfile (nicaragua.map): almacena todas las configuraciones necesarias para dibujar el mapa.
- Página inicial (index.html): crea las configuraciones iniciales y lanza el primer mapa.
- Plantilla HTML (global.html): una plantilla que MapServer llena para mostrar el mapa actual, leyenda, etc.

6.3.1 Creando el MapFile

Las configuraciones principales de MapServer se realizan a través de un archivo de texto que se lee en tiempo de ejecución. Este archivo es llamado mapfile, y es el corazón de la mayoría de las aplicaciones basadas en MapServer. El programa CGI o la aplicación MapScript leen esta información de configuración, acceden a los datos, dibujan el mapa y retornan un gráfico listo para ser visualizado en línea. El mapfile tiene una estructura jerárquica simple de

objetos, con objetos heredando configuraciones de sus padres. En el anexo A se encuentra una referencia de los objetos del Mapfile.

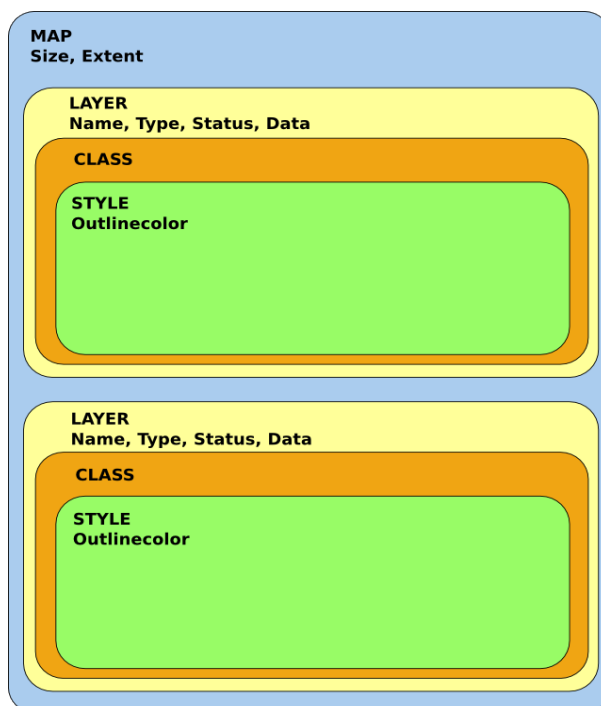


Ilustración 5: Estructura de un MapFile con dos layers.

Para nuestra aplicación creamos un mapfile que muestra una leyenda para el mapa, una barra de escala y que muestra cuatro layers: el primer layer es de tipo polígono, siempre está visible y muestra la división política administrativa de Nicaragua, el segundo layer es de tipo punto, puede mostrarse y ocultarse y muestra la ubicación de las ciudades de Nicaragua, el tercer layer es de tipo línea, puede mostrarse y ocultarse y muestra la red vial de Nicaragua, y el cuarto layer es de tipo línea, puede mostrarse y ocultarse y muestra los ríos de nuestro país. Las fuentes de datos para nuestra aplicación serán archivo en formato Shape de ESRI. Las configuraciones generales del mapa se llevan a cabo agregando las siguiente líneas a nuestro mapfile:

```
SIZE 600 400  
EXTENT -89 10 -83 16
```

```
IMAGECOLOR -1 -1 -1
IMAGETYPE JPEG
SHAPEPATH "shapes"
FONTSET "fonts/fonts.list"
UNITS DD

PROJECTION
    "proj=utm"
    "zone=16N"
END
```

El atributo SIZE define el tamaño en pixels de la imagen del mapa que se va a generar. El atributo EXTENT establece la extensión inicial del mapa y está dada en la unidades establecidas en el atributo UNITS, que en nuestro caso son Grados Decimales (DD por su siglas en inglés). Los atributos IMAGECOLOR e IMAGETYPE definen el color de fondo de la imagen generada y el formato de salida de dicha imagen, en nuestra aplicación definimos el color de fondo como transparente. El atributo SHAPEPATH indica el directorio en el que se encuentran los archivos en formato Shape que utilizamos como orígenes de datos para nuestro mapa. El atributo FONTSET establece la ubicación de la lista de fuentes TrueType disponibles para ser utilizadas en nuestro mapa.

La sección PROJECTION la utilizamos para establecer el tipo de proyección que utilizaremos para mostrar nuestro mapa, para esta aplicación elegimos el sistema de proyecciones UTM que divide el globo terráqueo en zonas e indicamos que vamos a proyectar nuestro mapa en la zona 16N, que es la zona correspondiente a nuestro país.

En nuestro mapfile también debemos incluir información sobre los elementos Web que van a interactuar con MapServer para generar y presentar el mapa. Para ellos hacemos uso del objeto Web.

```
WEB
    TEMPLATE global.html
    IMAGEPATH "/var/www/ms_tmp/"
    IMAGEURL "/ms_tmp/"
END
```

El atributo TEMPLATE le indica a MapServer que la plantilla Web a utilizar para mostrar el mapa es “global.html”. IMAGEPATH indica la dirección absoluta en donde MapServer debe guardar las imágenes generadas y el atributo IMAGEURL indica el URL que se va a utilizar en la plantilla Web para encontrar la ubicación de las imágenes.

Para establecer las configuraciones de los diferentes layers que vamos a mostrar en nuestro mapa, debemos hacer uso del objeto LAYER. Cada layer debe contener información como el nombre del layer, el tipo, estado y fuente de datos, de igual manera de contener como mínimo un objeto CLASS.

```
NAME Nicaragua
TYPE POLYGON
STATUS DEFAULT
DATA nic_admin

CLASSITEM 'DEPTO'
CLASS
    NAME 'Boaco'
    EXPRESSION 'Boaco'
    STYLE
        OUTLINECOLOR 100 100 100
        COLOR 180 180 180
    END
END
...
```

Como podemos observar, los primero cuatro atributos definen las características de del layer. El atributo NAME define el nombre de layer. El atributo TYPE indica del tipo de datos que muestra ese layer. El atributo STATUS indica el estado del Layer, puede ser DEFAULT, ON y OFF, DEFAULT indica siempre activo y el atributo DATA indica la fuente datos para ese layer, en este caso particular indica que la fuente de datos es un archivo Shape de ESRI que se llama nic_admin.

El atributo CLASSITEM indica que se utilizará la columna de la fuente de datos indicada. El objeto CLASS nos permite clasificar los elemento de la columna CLASSITEM a través de la expresiones indicadas en el atributo EXPRESSION. Siempre que definimos un nuevo objeto

CLASS aplicamos un estilo diferente a él. En nuestra aplicación, creamos un objeto CLASS para cada departamento del país y establecemos un color de fondo distinto para cada uno de ellos. Para ver el contenido completo del mapfile, lea el anexo B.

6.3.2 Creando la Página de Inicialización

La página de inicialización contiene aspectos relevantes a la información entregada por la aplicación; su función principal es entregar internamente un conjunto de parámetros de inicialización a MapServer, los cuales se pueden ver a continuación:

```
<form method=GET action="/cgi-bin/mapserv">
  <input type="hidden" name="map"
        value="/var/www/nicaragua/nicaragua.map">
  <input type="hidden" name="zoomdir" value=0>
  <input type="hidden" name="zoomsize" value=2>

  <input type="hidden" name="program"
        value="/cgi-bin/mapserv">

  <input type="submit" value="Mostrar Mapa">
</form>
```

Estos parámetros son fijos y muy comunes, se utilizan en la mayoría de las aplicaciones desarrolladas con MapServer, lo único que cambia según la aplicación son los nombres de los archivos y las rutas de acceso.

Lo que hacen estos parámetros es: por medio del CGI de MapServer llamar el archivo de configuración nicaragua.map e inicializa el tamaño del zoom en 2. Luego le dice a MapServer en que directorio debe ir a buscar las imágenes que se generan después de cada petición Web.

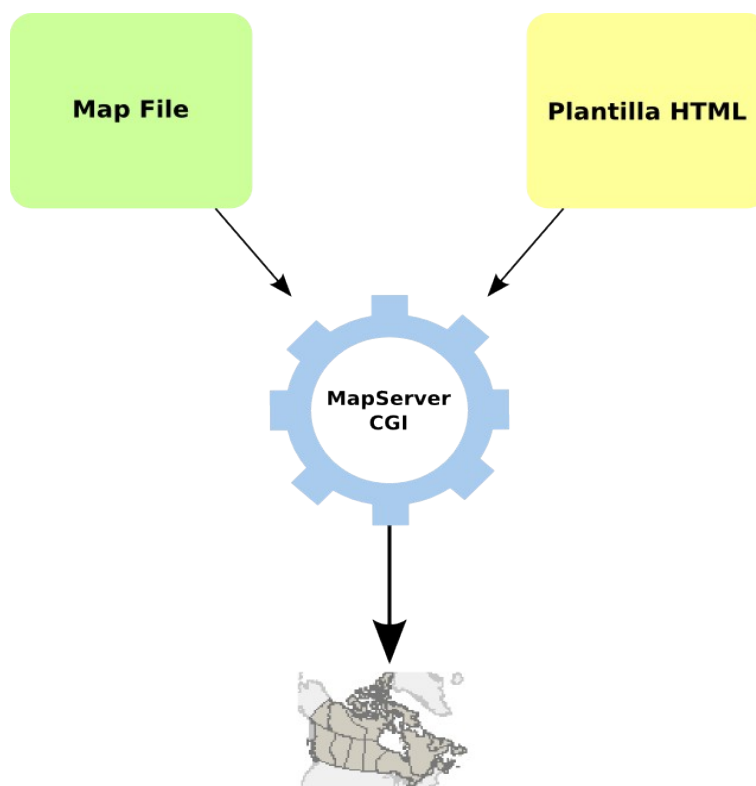


Ilustración 6: Función de la Página de Inicialización

6.4 Instalación y Uso de PostGIS

La distribución Ubuntu 7.04 Feisty Fawn contiene en sus repositorios paquetes para la instalación de las extensiones PostGIS, por lo que la instalación del servidor de bases de datos PostgreSQL junto con las extensiones PostGIS es bastante sencilla.

Para instalar las extensiones PostGIS debemos tener activado el repositorio Universe, para ello abrimos el gestor de paquetes Synaptic que se encuentra en el submenú Administración del Menú Sistema. En el menú Configuración seleccionamos Repositorios.

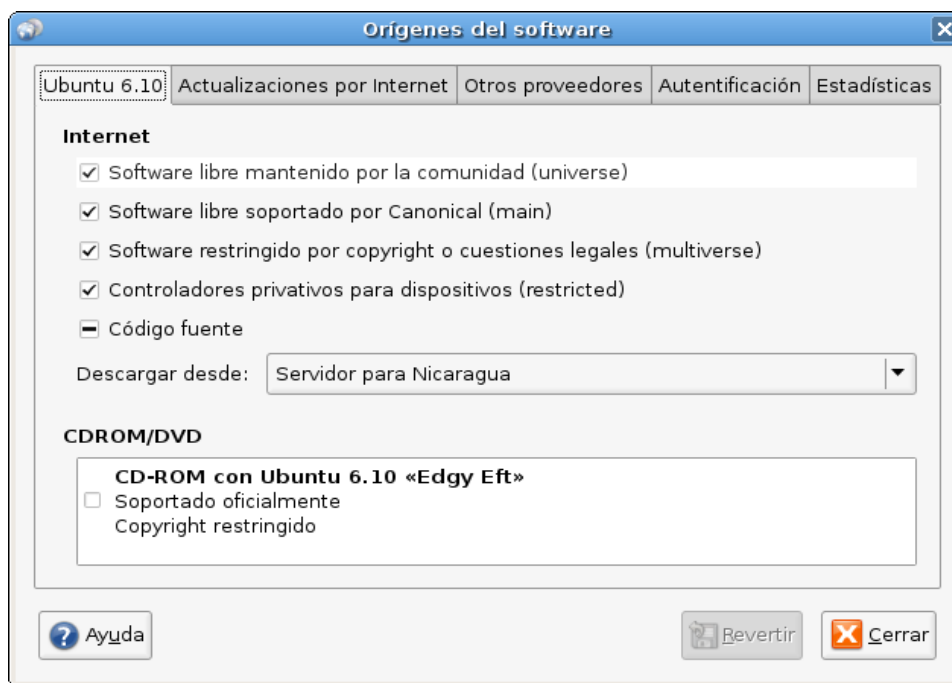


Ilustración 7: Activar el Repositorio Universe desde Synaptic

Activamos la casilla de verificación que está junto al texto "Software Libre mantenido por la comunidad (Universe)" y pulsamos el botón cerrar. Pulsamos el botón recargar en la barra de herramientas de Synaptic, para que el sistema obtenga la lista de paquetes del nuevo repositorio. Una vez activado el repositorio Universe, instalamos Postgresql con las extensiones PostGIS con el siguiente comando desde la consola.

```
$ sudo apt-get install libgeos2 postgresql-8.1 postgresql-server-dev-8.1 libgeos-dev postgresql-dev postgresql-8.1-postgis proj
```

Si no ha habido errores, ya tenemos instalado en nuestro sistema Postgresql con las extensiones PostGIS. A continuación, agregamos una cuenta de usuario al sistema que pertenezca al grupo postgres. Para ello, seleccionamos Usuarios y Grupos en el submenú Administración del menú Sistema. Pulsamos el botón Añadir Usuario y se nos muestra una ventana de diálogo como la siguiente:

Propiedades de la cuenta «gis»

Cuenta Privilegios del usuario Avanzado

Configuración básica

Nombre de usuario: gis

Nombre real: GIS

Información de contacto

Ubicación en la oficina:

Teléfono del trabajo:

Teléfono del domicilio:

Contraseña

Establecer la contraseña a mano

Contraseña del usuario:

Confirmación:

Generar una contraseña aleatoria

Contraseña establecida a:

Ilustración 8: Datos del Nuevo Usuario

En el campo Nombre de usuario escribimos *gis*, en el campo Nombre real escribimos *GIS* y Establecemos una contraseña a mano, en nuestro caso la contraseña es *gis*. Seleccionamos la pestaña Avanzado y en el combo Grupo principal seleccionamos *postgres* y pulsamos el botón Aceptar.



Ilustración 9: Agregando el nuevo usuario al grupo postgres

Este nuevo usuario nos servirá para autenticarnos cuando vayamos a acceder a PostgreSQL desde máquinas remotas.

6.4.1 Habilitando el acceso a PostgreSQL desde la red

PostgreSQL viene preconfigurado para atender solo conexiones locales al puerto 5432, para adicionar el acceso a otras máquinas en la red a este servicio se requiere modificar la configuración predeterminada de PostgreSQL.

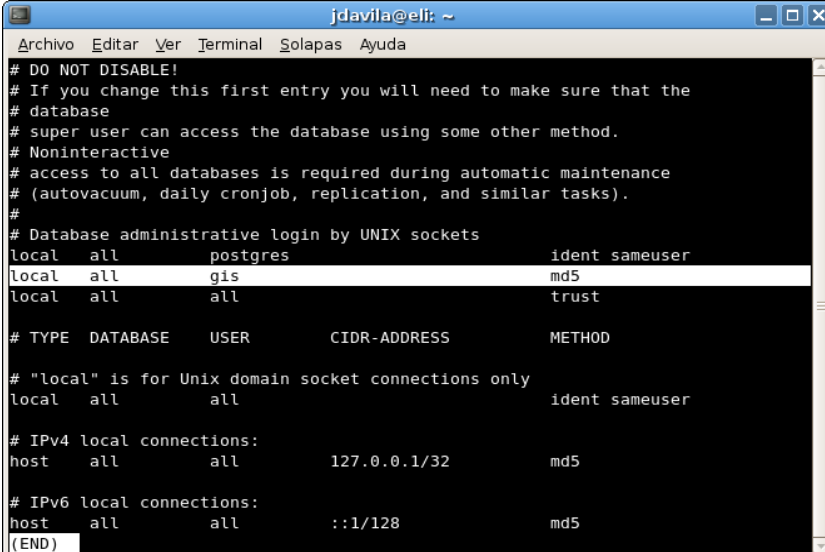
Los archivos de configuración se encuentran en la carpeta `/etc/postgresql/8.1/main/` y son:

- `pg_hba.conf`. Controla cuales hosts están autorizados para conectar, cómo los clientes son autenticados, por medio de cuales usuarios se puede acceder y a cuales bases de datos.
- `pg_ident.conf`. Registra el mapeo de usuarios Unix con los usuarios PostgreSQL.
- `postgresql.conf`. Este archivo contiene los parámetros de configuración general, tales como el puerto en donde atiende el servicio, la cantidad de conexiones entre otros.

En este punto es necesario configurar el archivo `pg_hba.conf` para habilitar el acceso de otras máquinas de la red al servicio PostgreSQL. La modificación consiste en agregar las siguientes líneas dentro del conjunto de máquinas permitidas:

```
local  all          gis          md5
local  all          all          trust
```

Una vez realizada la modificación, nuestro archivo de configuración quedó a como se muestra en siguiente ilustración:



```

jdavila@eli: ~
Archivo Editar Ver Terminal Solapas Ayuda
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database
# super user can access the database using some other method.
# Noninteractive
# access to all databases is required during automatic maintenance
# (autovacuum, daily cronjob, replication, and similar tasks).
#
# Database administrative login by UNIX sockets
local  all          postgres      ident sameuser
local  all          gis          md5
local  all          all          trust

# TYPE DATABASE  USER        CIDR-ADDRESS  METHOD
# "local" is for Unix domain socket connections only
local  all          all          ident sameuser

# IPv4 local connections:
host   all          all          127.0.0.1/32  md5

# IPv6 local connections:
host   all          all          ::1/128       md5
(END)

```

Ilustración 10: Archivo `pg_hba.conf` modificado

6.4.2 Creando la base de datos geoespacial

Vamos a empezar agregando un usuario llamado 'gis' (los usuarios de PostgreSQL son independientes de los usuarios del sistema). En un entorno de producción, se le otorgarían los privilegios justos para introducir, buscar y actualizar datos. Por lo pronto, lo crearemos como un superusuario.

```
$ sudo su postgres -c psql
#CREATE ROLE gis LOGIN CREATEDB CREATEROLE PASSWORD 'gis'
#\q
```

Hemos creado un usuario gis con privilegios de superusuario y con la contraseña gis. Utilizaremos este usuario para realizar todas las tareas relacionadas con nuestra base de datos geoespacial. Para crear nuestra base de datos, iniciamos sesión como usuario gis.

```
$ sudo su gis
$ createdb -U gis --encoding SQL_ASCII db_gis
```

Ahora que hemos creado nuestra base de datos, vamos a agregarle las extensiones de PostGIS para almacenar y manipular datos geoespaciales.

```
$ createlang -U gis plpgsql db_gis
$ psql -U gis -d db_gis -f /usr/share/postgresql-8.1-
postgis/lwpostgis.sql
$ psql -U gis -d db_gis -f /usr/share/postgresql-8.1-
postgis/spatial_ref_sys.sql
```

Podemos proceder a importar los archivos shape (*.shp). Para ello hacemos uso de la utilidad shp2pgsql que viene con PostGIS:

```
$ shp2pgsql -D nic_admin.shp nic_admin db_gis | psql -U gis -d
db_gis
$ shp2pgsql -D nic_cabmuni.shp nic_cabmuni db_gis | psql -U gis
-d db_gis
$ shp2pgsql -D nic_redvial.shp nic_redvial db_gis | psql -U gis
-d db_gis
```

```
$ shp2pgsql -D nic_rios.shp nic_rios db_gis | psql -U gis -d db_gis
```

Los parámetros `nic_rios` y `db_gis` indican el nombre de la tabla en la que se guardará la información geoespacial y la base de datos en la que se creará la tabla, respectivamente. El resultado devuelto por `shp2pgsql` se escribe en la salida estándar, por eso utilizamos una tubería para agregarlo a la base de datos de PostgreSQL que habilitamos para manejar datos geoespaciales. La utilidad `shp2pgsql` agrega una columna llamada “`the_geom`” en cada tabla creada, para almacenar la información geoespacial de cada feature.

6.4.3 Modificaciones en el mapfile

Para mostrar los datos almacenados en la base de datos de PostgreSQL que creamos en el apartado anterior, debemos de modificar las fuentes de datos en nuestro mapfile de manera que la información geoespacial de cada layer se obtenga desde la base de datos y no desde un archivo shape.

Para ello, abrimos nuestro mapfile (`nicaragua.map`), ubicamos el objeto DATA en cada uno de los layers y los sustituimos por las siguientes líneas:

```
#Layer de Fondo
CONNECTIONTYPE POSTGIS
CONNECTION "host=localhost dbname=db_gis user=gis password=gis
port=5432"
DATA "the_geom from nic_admin"
```

```
#Layer de Ciudades
CONNECTIONTYPE POSTGIS
CONNECTION "host=localhost dbname=db_gis user=gis password=gis
port=5432"
```

```
DATA "the_geom from nic_cabmuni"  
  
#Layer de Carreteras  
CONNECTIONTYPE POSTGIS  
CONNECTION "host=localhost dbname=db_gis user=gis password=gis  
port=5432"  
DATA "the_geom from nic_redvial"  
  
#Layer de Ríos  
CONNECTIONTYPE POSTGIS  
CONNECTION "host=localhost dbname=db_gis user=gis password=gis  
port=5432"  
DATA "the_geom from nic_rios"
```

La directiva CONNECTIONTYPE POSTGIS indica que nos vamos a conectar a una base de datos PostgreSQL que tiene habilitada la extensión PostGIS.

El objeto CONNECTION establece la cadena de conexión a la base de datos PostgreSQL. Indicar el puerto a través del cual nos conectamos a PostgreSQL es opcional en la cadena de conexión.

Como podemos notar el objeto DATA ahora tiene la siguiente sintaxis: "the_geom from nombre_tabla". Esto le indica a MapServer que el origen de datos geoespaciales es la columna "the_geom" de la tabla indicada.

6.5 Creando Mapas con PHP-MapScript

6.5.1 Obtener la extensión de MapScript para PHP

Antes que empecemos a utilizar PHP MapScript debemos estar seguro que instalamos y configuramos las herramientas de MapScript. En el apartado 6.1.3 de este documento

instalamos MapServer con la extensión MapScript para PHP 5. De igual manera en el sitio principal de PHP MapScript [http://maptools.org/php_mapscript/] podemos encontrar instrucciones para descargar la extensión así como un documento de referencia del API.

Para activar la extensión MapScript en PHP editamos el fichero de configuración de PHP que se encuentra en `/etc/php5/apache2/php.ini` y agregamos la siguiente línea:

```
extension=php_mapscript.so
```

Para verificar que la extensión se ha agregado correctamente creamos un fichero llamado `phpinfo.php` y lo guardamos en el DocumentRoot de nuestro servidor Web, y escribimos las siguientes líneas de código PHP en el fichero:

```
<?php
if (!extension_loaded("MapScript"))
    dl("php_mapscript.so");
phpinfo();
?>
```

Si accedemos al fichero recién creado a través de nuestro navegador Web preferido, se mostrará la información de configuración de PHP y aparecerá un apartado llamado MapScript si la extensión de MapScript está instalada y activada.

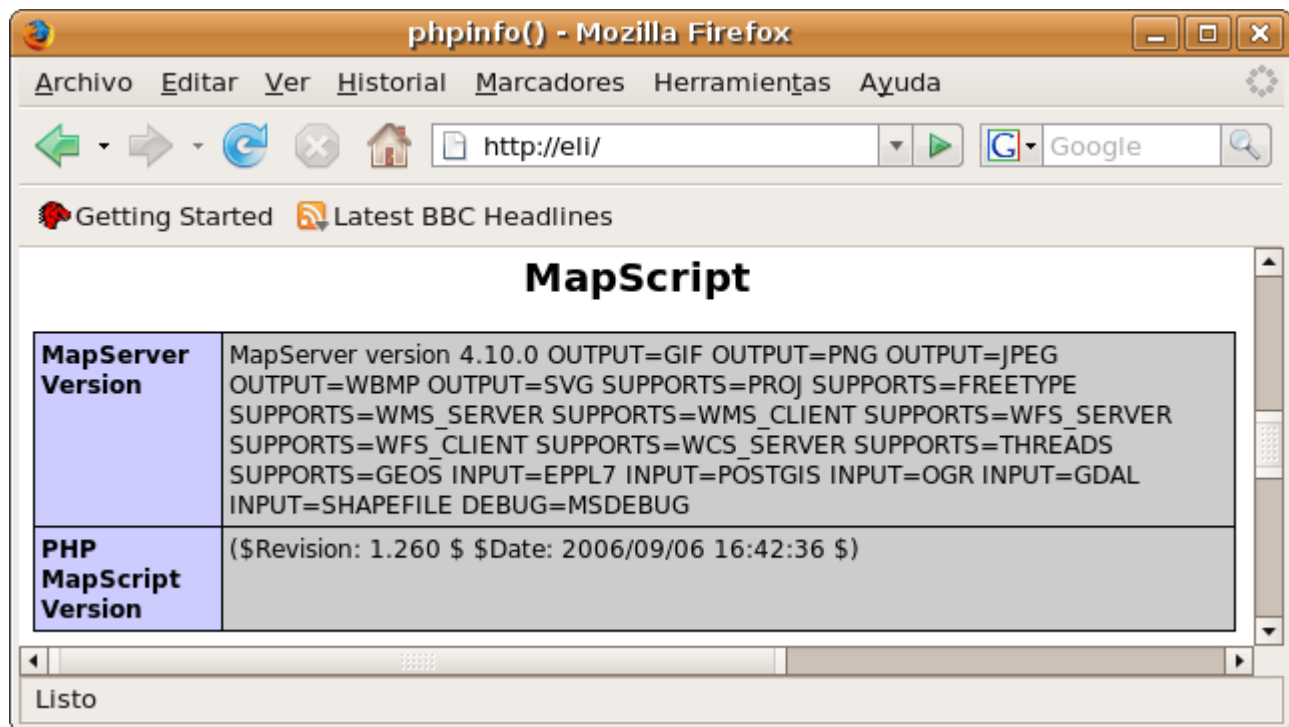


Ilustración 11: Extensión de MapScript cargada en PHP.

Ahora que hemos agregado la extensión MapScript a PHP podemos reutilizar el mapfile con el que hemos estado trabajando para presentar el mapa de Nicaragua con los distintos layers de información utilizando PHP MapScript. Creando un objeto map en nuestro script tendremos acceso a muchos métodos para trabajar con un mapfile específico.

```
$map_path = "/var/www/nicaraguaphp/";
$map_file = "nicaragua.map";
//Obtener un mapfile y crear un mapa desde el
$map = ms_newMapObj($map_path.$map_file);
```

Basta con tres líneas de código PHP para realizar las tareas básicas de la creación de mapas: abrir un mapfile, renderizar una imagen y guardar la imagen en un fichero. Las líneas de código PHP para renderizar y guardar la imagen son las siguientes:

```
$image=$map->draw();
```

```
$image->saveImage($img_path.$image_name);
```

Para mostrar y ocultar layers utilizamos las siguientes líneas de código PHP:

```
if ($_POST['layer'])
{
    $layer = join("",$_POST['layer']);
}
else
{
    $layer = "";
}
$this_layer = 0;

if (preg_match("/rios/", $layer))
{
    $rios = "CHECKED";
    $this_layer = $map->getLayerByName('rios');
    $this_layer->set('status', MS_ON);
}
else
{
    $rios = "";
    $this_layer = $map->getLayerByName('rios');
    $this_layer->set('status', MS_OFF);
}
if (preg_match("/vias/", $layer))
{
    $vias = "CHECKED";
    $this_layer = $map->getLayerByName('vias');
    $this_layer->set('status', MS_ON);
}
```

```
}
else
{
    $vias = "";
    $this_layer = $map->getLayerByName('vias');
    $this_layer->set('status', MS_OFF);
}
if (preg_match("/cabeceras/", $layer))
{
    $cabeceras = "CHECKED";
    $this_layer = $map->getLayerByName('cabeceras');
    $this_layer->set('status', MS_ON);
}
else
{
    $cabeceras = "";
    $this_layer = $map->getLayerByName('cabeceras');
    $this_layer->set('status', MS_OFF);
}
```

Como podemos observar, primero verificamos si ha cambiado la lista de layers seleccionados con la línea `if ($_POST['layer'])` si alguno de los layers ha cambiado de estado entonces almacenamos su nombre en una variable para luego verificar cual fue el layer que cambió de estado y establecer su estado a través del objeto map.

Para controlar las acciones de zoom y desplazamientos en el mapa primero identificamos la acción a realizar (zoom in/out, pan). Para identificar la acción a realizar hacemos el cálculo de un factor basándonos en los valores que establecimos en el formulario Web para cada una de esas acciones. Si el factor es mayor que cero, vamos a realizar zoom in, si el factor es menor que cero vamos a realizar zoom out y finalmente si el factor es igual a cero vamos

a realizar un desplazamiento en el mapa. Todo esto lo realizamos con el siguiente código PHP:

```
$zoom_factor = $_POST['zoom']*$_POST['zsize'];

// Establece la dirección del zoom
if ($zoom_factor == 0){
    $zoom_factor = 1;
    $pan = "CHECKED";
    $zoomout = "";
    $zoomin = "";
} elseif ($zoom_factor < 0){
    $pan = "";
    $zoomout = "CHECKED";
    $zoomin = "";
} else {
    $pan = "";
    $zoomout = "";
    $zoomin = "CHECKED";
}
}
```

Una vez realizado el cálculo del factor y establecida la acción a realizar sobre el mapa, obtenemos el tamaño del zoom y realizamos la acción con las siguientes líneas:

```
$zoomsize = abs( $_POST['zsize']);
$map->zoomPoint($zoom_factor,$clkpoint,$map->width,
    $map->height,$old_extent,$max_extent);
```

Nuestro script puede hacer mucho más si lo hacemos interactivo para el usuario vinculando código MapScript con aplicaciones Web. El código completo de una aplicación Web interactiva que utiliza MapScript se puede observar en el anexo E.

7 Conclusiones

La instalación y configuración de nuestro servidor de mapas utilizando herramientas libres y de código abierto resultó muy sencilla gracias a la excelente colección de paquetes que se encuentra en los repositorios de Ubuntu 7.04 Feisty Fawn. La elección de esta distribución de Linux fue muy importante para alcanzar todos nuestros objetivos ya que nos ahorró el tedioso trabajo de compilar una gran cantidad de dependencias. La configuración del servidor de mapas descrita en este trabajo monográfico se aplica sin ningún cambio a la versión 6.10 Edgy Eft de Ubuntu.

Una gran ventaja de MapServer además de su rapidez en la entrega de resultados, es que al trabajar con imágenes vectoriales y el hecho de generar una nueva imagen cada vez que se realiza una actualización de una visualización, permite hacer acercamientos a un punto de la imagen sin que se pixelée, lo que entrega un alto nivel de detalle.

La utilización de servidores de mapas de libre distribución puede abrir las puertas a pequeñas y medianas empresas o instituciones públicas y privadas hacia una gestión más abierta desde el punto de vista de la comunicación, al compartir información valiosa para muchas personas sin importar el lugar donde éstas se encuentren.

8 Recomendaciones

El uso del software libre y la demanda de profesionales con conocimientos avanzados de herramientas libres ha incrementado en nuestro país en los últimos años; por tales razones, recomendamos que se haga mayor uso de herramientas libres en las diferentes asignaturas de las carreras de Ingeniería en Sistemas e Ingeniería en Telemática. De igual manera recomendamos que se propongan temas monográficos o de investigación en el área del software libre.

El Departamento de Computación debería incluir en el pènsum de la carrera de Ingeniería en Sistemas un curso electivo donde se estudien los Sistemas de Información Geográfica para ampliar el ámbito de trabajo de los egresados de la carrera.

Para facilitar la creación y edición de los mapfiles recomendamos utilizar la herramienta MapLab que permite la creación y modificación de los mapfiles utilizando una interfaz Web muy intuitiva. Además incluye una herramienta llamada MapBrowser para visualizar los mapfiles creados utilizando siempre una interfaz Web. Esta herramienta puede ser de mucha utilidad para futuros trabajos monográficos o de investigación que abarquen el uso de MapServer.

9 Bibliografía

Gomez Dans, J. "Como instalar PostGIS en Kubuntu". Blog de jgomezdans. [<http://jgomezdans.googlepages.com>].

González Jaramillo Víctor H. "MapServer y su aplicación a SIG". Universidad Técnica Particular de Loja, Ecuador. 2005.

Kropla, Bill. "Beginning MapServer Open Source GIS Development", 2005. Apress.

Lantada Zarzosa, Nieves. Núñez Andrés, M. Amparo. "Sistemas de Información Geográfica". Alfaomega. 2006.

Mitchell, Tyler. "Web Mapping Illustrated. Using Open Source GIS Toolkits", 2005. O'Reilly.

Penroz Díaz, Alvaro Antonio. "Graphical User Interface (GUI) para el programa servidor de mapas MapServer 4.6.1". Universidad de la Frontera – Temuco, Chile. 2005.

Powell Rivera, Valeska Ivonne. "Sistema de Información sobre un Plan Regulador Municipal a través de Internet". Universidad de la Frontera - Temuco, Chile. 2003.

Sánchez Forero, Edilberto. "Curso de Desarrollo de Aplicaciones con .NET, Mono y GTK". 2005. Universidad de Colombia, Grupo EIDOS.

Anexo A - Referencia de Objetos MapServer

Objeto MAP (mapa)

extent [minx][miny][maxx][maxy]: Indica el fragmento espacial del mapa a ser creado en coordenadas terrestres.

fontset [filename]: Nombre completo del archivo fuente a ser usado. Estos son archivos de texto que contienen información acerca de las fuentes TrueType disponibles para Mapserver.

imagecolor [r][g][b]: Color con el cual se inicializa el mapa.

interlace [on|off]: Entrelaza las imágenes de salida. Valor predeterminado es on.

layer: Señala el inicio de un objeto cobertura.

legend: Señala el inicio de un objeto leyenda.

name [string]: Prefijo adjunto al mapa, usado por los archivos temporales que muestran la barra escalar y leyenda, son creados usando este archivo. Es recomendado sólo un par de caracteres.

projection: Señala el inicio de un objeto proyección.

querymap: Señala el inicio de un objeto consulta.

reference: Señala el inicio del objeto mapa de referencia.

scale [double]: Escala del mapa. Generalmente entregada por la aplicación.

scalebar: Señala el inicio del objeto barra escalar.

shapepath [path]: Ruta a la colección de archivos Shapefile.

size [cols][rows]: Tamaño de la imagen, en píxeles.

status [on|off]: Indica si un mapa esta activo. A veces sólo se desea visualizar la barra escalar y la leyenda.

title [string]: Nombre del directorio que contiene los archivos Shapefile.

transparent [on|off]: Indica si el color de fondo de los mapas es transparente o no. El valor por defecto es off.

units [feet|inches|kilometers|meters|miles|dd]: Unidades de medición de las coordenadas del mapa. Usado por la barra escalar para los cálculos de escala.

web: Señala el inicio de un objeto Web.

Objeto LABEL (Etiqueta)

angle [double]: ángulo, dado en grados.

backgroundcolor [r][g][b]: Color con el cual dibujar un cuadrado de fondo.

backgroundshadowcolor [r][g][b]: Color con el cual dibujar un cuadrado de fondo con sombra.

backgroundshadowsize [x][y]: Cuan lejos se debe compensar el rectángulo del fondo. El valor predeterminado es 1 píxel.

buffer [integer]: Valor de relleno alrededor de las etiquetas (en píxeles). Útil para mantener el

espacio alrededor del texto y reforzar la legibilidad. El valor predeterminado es 0.

color [r][g][b]: Color para el texto.

font [filename]: Alias de la fuente usado para etiquetar (definido en el fontset).

force [true|false]: Fuerza a las etiquetas de una clase particular a cambiar a on, sin tener en cuenta las colisiones. Valor predeterminado es false.

mindistance [integer]: Distancia mínima entre etiquetas duplicadas. Medido en píxeles.

minfeaturesize [integer|auto]: Tamaño mínimo al que un elemento de una cobertura es etiquetado. Medido en píxeles. Para los datos de línea se usa la longitud global de la línea desplegada, para las características de polígono se usa la dimensión más pequeña de la caja limitante. La palabra clave auto le dice a MapServer que etiquete sólo elementos que son más grandes que su etiqueta correspondiente.

maxsize [integer]: El tamaño de la fuente máximo para usar cuando el texto es escalar (en píxeles). Valor predeterminado es 256.

minsize [integer]: El tamaño de la fuente mínimo para usar cuando el texto es escalar (en píxeles). Valor predeterminado es 4.

offset [x][y]: Valor de desplazamiento relativo, para una etiqueta, a la esquina inferior izquierda de la etiqueta. Medido en píxeles. En el caso de texto girado, se deben especificar los valores como si todas las etiquetas estuviesen horizontales y cualquier rotación será compensada.

outlinecolor [r][g][b]: Color de la frontera de un texto.

position [ul|uc|ur|cl|cc|cr|ll|lc|lr|auto]: Posición de la etiqueta, relativa a la sección donde estas son ubicadas. La primera letra es la posición Y (arriba, centrado o abajo), la segunda es la posición X (izquierda, centrado o derecha). Auto le dice a Mapserver que calcule una posición de etiqueta que no interfiera con otras etiquetas. Si todas las posiciones causan el traslapo (sobreposición), entonces la etiqueta no es dibujada. Esta solo se aplica a etiquetas de coberturas (no a leyendas o barra escalar).

size [integer or tiny|small|medium|large|giant]: Tamaño del texto. Si se usa una fuente TrueType entonces el tamaño es especificado en píxeles por puntos Si se usan fuentes construidas con bitmap el tamaño es uno de los 5 especificados en la lista.

type [bitmap|truetype]: Selecciona el tipo de fuente a usar. Generalmente son usadas fuentes de bitmap, ya que son más rápidas de dibujar que las fuentes TrueType. De todas maneras, las fuentes TrueType son escalables y están disponibles en una variedad de faces.

wrap [character]: Carácter que representa una condición de fin de línea en etiquetas de texto, produciendo una etiqueta multilínea. Las etiquetas multilínea son ubicadas correctamente.

Objeto LAYER (Cobertura)

connection [string]: String de conexión de la base de datos para recuperar datos remotos. En la actualidad solo ESRI SDE es soportado. Un string de conexión SDE consiste en un nombre de host, instancia de nombre, nombre de base de datos usuario y password separado por comas.

connectiontype [local|sde]: Tipo de conexión. El valor por defecto es local.

data [filename]: Nombre completo de los datos espaciales a procesar No es necesaria la extensión de los archivos Shapefile.

description [string]: Una corta descripción de la cobertura.

feature: Señala el inicio de un objeto característica.

footer [filename]: La plantilla Web (sección de página Web) para usar después de que un juego de resultados de coberturas se haya enviado.

header [filename]: La plantilla Web (sección de página Web) para usar antes de que un juego de resultados de coberturas se haya enviado.

labelcache [on|off]: Especifica si deben dibujarse las etiquetas como elementos para esta capa o si ellos deben esconderse y deben dibujarse después que todas las coberturas han sido dibujadas. El valor predeterminado es on.

labelitem [string]. Nombre de artículo de la tabla de atributo usado por la clase annotation.

labelmaxscale [double]: Escala máxima a la cual esta cobertura esta etiquetada.

labelminscale [double]: Escala mínima a la cual esta cobertura esta etiquetada.

labelsizeitem [string]: Nombre de artículo de la tabla de atributo usado por la clase annotation sizes. El valor debe estar en píxeles.

maxfeatures [integer]: Determina el número de elementos que deben ser dibujadas para esta cobertura en la ventana actual.

maxscale [double]: Escala máxima en la cual la cobertura es dibujada.

minscale [double]: Escala mínima en la cual la cobertura es dibujada.

name [string]: Nombre corto para la cobertura. El límite es de 20 caracteres. Este nombre es el lazo entre el archivo MAP y la interfaz Web. El nombre debe ser único.

postlabelcache [true|false]: Le dice a MapServer que entregue esta cobertura después que todas las etiquetas del caché han sido dibujadas. El valor por defecto es false.

query: Señala el inicio de un objeto consulta.

queryitem [string]: Nombre del campo de la tabla de atributos usado por las consultas.

symbolscale [double]: La escala en la cual el símbolo y/o texto se presenta en tamaño completo. Esto permite escalas dinámicas de objetos basados en la escala del mapa. Si no se indica, esta cobertura se verá siempre del mismo tamaño.

status [on|off|default|queryonly]: Indica el estado actual de la cobertura que es modificado por MapServer constantemente. El valor predeterminado es on. Queryonly significa que la cobertura puede ser consultada pero no dibujada.

titleindex [filename]: Nombre de archivo completo para el índice o definición de un título para esta cobertura.

tolerance [double]: Grado de tolerancia para consultas en puntos (click del mouse en una coordenada del mapa). Medido en unidades de tolerancia con un valor predeterminado de 3 píxeles.

toleranceunits [pixels|feet|inches|kilometers|meters|miles|dd]: Unidad de medida del valor de tolerancia. Valor predeterminado es el pixel.

transform [true|false]: Le indica a Mapserver cuando una cobertura no necesita ser transformada desde un sistema de coordenadas a una imagen coordenada. Valor predeterminado es true. Esto permite que se creen archivos Shapefile en coordenadas de

image|graphics y, por lo tanto, que tenga características que sean visualizadas siempre en la misma localización en cada correspondencia. Ideal para ubicar logos o texto en mapas.

type [point|line|polyline|polygon|annotation|raster]: Especifica como deben ser dibujados los datos. No debe ser el mismo tipo del archivo Shapefile.

Objeto CLASS (Clase)

backgroundcolor [r][g][b]: Color para símbolos no transparentes.

color [r][g][b]: Color a usar para el dibujo.

expression [string]: Tres tipos de expresiones son ahora soportadas para definir un miembro de clase. Comparación de string, expresiones regulares y expresiones lógicas simples. Si no se da la expresión entonces todas las características pertenecen a esta clase. - Las comparaciones de string son las más rápidas de evaluar. Ningún delimitador especial es necesario aunque las cadenas deben ser cotizadas si contienen caracteres especiales. - Una expresión regular se debe delimitar usando /regex/. - Las expresiones lógicas permiten la construcción de textos bastante complejos sobre uno o más atributos y por lo tanto solo está disponible para archivos Shapefile. Las expresiones lógicas son delimitadas por [expresión]. Los nombres de atributos son delimitados por [atributo]. Estos nombres deben concordar con los nombres de los archivos Shapefile. Por ejemplo ([AGE] > 50 and '[COLOR]' eq 'RED'). Se pueden hacer mezclas de expresiones con las clases de las coberturas.

label: Señala el inicio de un objeto label.

maxsize [integer]: Tamaño máximo, en píxeles, para dibujar un símbolo. Valor predeterminado es 50.

minsize [integer]: Tamaño mínimo, en píxeles, para dibujar un símbolo. Valor predeterminado es 0.

name [string]: Nombre que se usará en la leyenda para esta clase. Si no se indica la clase no se mostrará en la leyenda.

outlinecolor [r][g][b]: Color para el contorno de polígonos y algunos marcadores. Los símbolos de línea no soportan color de contorno.

size [integer]: Altura, en píxeles, del símbolo/modelo a usar. Valor predeterminado es 1.

symbol [integer]: Número del símbolo, del archivo de símbolos, para usar en todas las características si la tabla de atributos no es usada. Valor predeterminado es 0, lo cual indica un simple píxel.

text [string]: Texto estático para etiquetar elementos con esta clase. Esto reemplaza los valores obtenidos del atributo labelitem. La cadena se puede dar como expresión delimitada con (). Esto permite concatenar varios atributos en una sola etiqueta.

Objeto FEATURE (Elemento)

class [string]: String usado para clasificar esta característica. En casos que la cobertura tenga múltiples clases definidas, el string es evaluado en lugar de las expresiones de clase. Esto es lo mismo que poner datos desde un campo en un archivo Shapefile.

text [string]: String usado para etiquetar este elemento.

Objeto LEGEND (Leyenda)

embed [true|false]: Es la imagen de leyenda que se muestra. El valor predeterminado es false.

imagecolor [r][g][b]: Color con que se inicializa la leyenda. label: Señala el inicio del objeto etiqueta.

outlinecolor [r][g][b]: Color del contorno de los símbolos.

position [ul|uc|ur|ll|lc|lr]: Posición de la imagen de leyenda. Valor predeterminado es lr.

size [x][y]: Tamaño de los símbolos en píxeles. Valor predeterminado es 20 por 10.

spacing [x][y]: Espacio entre símbolos (y) y etiquetas (x), en píxeles. Valor predeterminado es 5 por 5.

status [on|off]: Determina si muestra la imagen de leyenda. Valor predeterminado es off.

Objeto QUERY (Consulta)

expression [string]: Tres tipos de expresiones son ahora soportados para definir miembros de consultas. Comparaciones de string, expresiones regulares y expresiones lógicas simples. Si no se da una expresión entonces todas las características pertenecen a esta clase. - Las comparaciones de string son las más rápidas de evaluar. Ningún delimitador especial es necesario aunque las cadenas deben ser cotizadas si contienen caracteres especiales. - Una expresión regular se debe delimitar usando /regex/. - Las expresiones lógicas permiten la construcción de textos bastante complejos sobre uno o más atributos y por lo tanto solo está disponible para archivos Shapefile. Las expresiones lógicas son delimitadas por [expresión]. Los nombres de atributos son delimitados por [atributo]. Estos nombres deben concordar con los nombres de los archivos Shapefile. Por ejemplo ([AGE] > 50 and '[COLOR]' eq 'RED'). Se pueden hacer mezclas de expresiones con las clases de las coberturas.

join: Señala el inicio del objeto consulta relacional.

template [filename|url]: Archivo modelo o URL usado para presentar el resultado al usuario.

Objeto QUERYMAP (Mapa de la consulta)

color [r][g][b]: Color que caracteriza el enmarcado. Valor predeterminado es amarillo.

size [cois] [rows]: Tamaño, en píxeles, de la imagen. Valor predeterminado definido en el objeto mapa.

style [NORMAL|HILITE|SELECTED]: Indica como son manejadas las características seleccionadas. Las coberturas no requeridas son dibujadas de la forma normal. - NORMAL: Dibuja todas los elementos de acuerdo con las opciones de cada cobertura. - MILITE: Dibuja los elementos seleccionados usando COLOR, los no seleccionados son dibujadas de forma normal. - SELECTED: Dibuja solo los elementos seleccionados de forma normal.

Objeto JOIN (Consulta relacional)

from [ítem]: Adjunta un campo en un archivo Shapefile.

name [string]: Nombre único para el join. Valor requerido.

table [file]: Nombre del archivo de base de datos (se debe indicar el path completo) a adjuntar.

template [filename]: Archivo modelo (Página Web) usado con una o muchas consultas. El modelo es procesado una vez para cada registro y solo puede contener referencias a campos de la tabla adjunta.

to [Ítem]: campo adjunto en la tabla relacionada.

type [multiple|single]: Tipo de consulta. Valor predeterminado es single.

Objeto REFERENCE (Mapa de Referencia)

color [r][g][b]: Color con que se dibuja la caja de referencia. Para no rellenar se debe colocar cualquier componente en -1. Valor predeterminado es rojo.

extent [minx][miny][maxx][maxy]: La extensión espacial de la imagen de referencia.

image [filename]: Nombre completo de la imagen de referencia. Debe ser una imagen GIF.

outlinecolor [r][g][b]: Color usado para el contorno de la caja de referencia. Para no usar contorno se debe colocar cualquier componente en -1.

size [cols][rows]: Tamaño, en píxeles, de la imagen de referencia.

status [on|off]: Determina si muestra el mapa de referencia. Valor predeterminado es off.

Objeto SCALEBAR (Barra Escalar)

backgroundcolor [r][g][b]: Color para el fondo de la barra de escala.

color [r][g][b]: Color usado para dibujar todas las características si la tabla de atributos no es usada.

imagecolor [r][g][b]: Color para inicializar la barra escalar.

intervals [integer]: Número de intervalos en que se dividirá la barra escalar. Valor predeterminado es 4.

label: Señala el inicio del objeto etiqueta.

outlinecolor [r][g][b]: Color usado para el contorno de intervalos individuales. Para no usar contorno se debe colocar cualquier componente en -1, lo cual es el valor predeterminado.

position [ul|uc|ur|ll|lc|lr]: Ubicación de la barra escalara en la imagen. Valor predeterminado es lr.

postlabelcache [true|false]: Solo para uso con barras de escala insertadas. Le indica a MapServer que inserte la barra de escala después que todas las etiquetas han sido dibujadas. Valor predeterminado es false.

size [cols][rows]: Tamaño, en píxeles, de la barra de escala.

status [on|off|embed]: Indica si la imagen de la barra debe ser creada e insertada. Valor predeterminado es off.

style [integer]: Selecciona el estilo de la barra de escala. Estilos validos son 0 y 1.

units [feet|inches|kilometers|meters|miles]: Unidad de salida de la barra de escala, valor predeterminado es miles.

Objeto WEB (Interfaz Web)

empty [url]: URL al cual serán enviados los usuarios si una consulta falla. Si no se define, se usa el valor error, con lo cual se presentará un mensaje de error al usuario.

error [url]: URL al cual serán enviados los usuarios cuando ocurra un error. Si no es definido se presentará un mensaje de error al usuario.

footer [filename]: Modelo (página Web) a usar antes que cualquier dato sea enviado. Solo

para modo multiresultado (consultar varias coberturas con un solo click).

header [filename]: Modelo (página Web) a usar después que cualquier dato sea enviado.

imagepath [path]: Ruta del directorio temporal para almacenar archivos e imágenes. Debe tener permiso de escritura para el usuario Web.

imageurl [path]: URL base para la ruta de imagen. No es necesaria si se indica la dirección en el archivo modelo.

log [filename]: Archivo para almacenar el historial de Mapserver. Debe tener permiso de escritura para el usuario Web.

maxscale [double]: Escala máxima a la cual esta interfaz es valida. Cuando un usuario requiere un mapa de la menor escala Mapserver retorna un error. Por esto maxtemplate será usado para proveer de una interfaz más amigable al usuario.

maxtemplate [filename|url]: Archivo modelo o URL usado para presentar los resultados al usuario cuando se presente una llamada a la menor escala.

minscale [double]: Escala mínima a la cual esta interfaz es válida. Cuando un usuario requiere un mapa de la mayor escala Mapserver retorna un error. Por esto mintemplate será usado para proveer de una interfaz más amigable al usuario.

mintemplate [filename|url]: Archivo modelo o URL usado para presentar los resultados al usuario cuando se presente una llamada a la mayor escala.

template [filename|url]: Archivo modelo o URL usado para presentar los resultados al usuario de modo interactivo.

Anexo B - Archivo de Configuración nicaragua.map

MAP

```
SIZE 600 400
EXTENT -89 10 -83 16
IMAGECOLOR -1 -1 -1
IMAGETYPE JPEG
SHAPEPATH "shapes"
FONTSET "fonts/fonts.list"
```

```
UNITS DD
```

```
PROJECTION
    "proj=utm"
    "zone=16N"
```

```
END
```

```
SYMBOL
    NAME 'circle'
    TYPE ELLIPSE
    POINTS 1 1 END
    FILLED TRUE
```

```
END
```

```
SYMBOL
    NAME "star"
    TYPE VECTOR
    POINTS
        0 0.375
        0.35 0.375
        0.5 0
        0.65 0.375
        1 0.375
        0.75 0.625
        0.875 1
        0.5 0.75
        0.125 1
        0.25 0.625
        0 0.375
```

```
END
FILLED TRUE
STYLE
    1 20 1 20
```

```
        END
    END

SCALEBAR
    IMAGECOLOR 255 255 255
    LABEL
        COLOR 0 0 0
        SIZE TINY
    END
    STYLE 1
    SIZE 100 2
    COLOR 0 0 0
    UNITS KILOMETERS
    INTERVALS 3
    TRANSPARENT TRUE
    STATUS EMBED
    OUTLINECOLOR 0 0 0
END

LEGEND
    KEYSIZE 12 12
    LABEL
        TYPE BITMAP
        SIZE MEDIUM
        COLOR 0 0 89
    END
    STATUS ON
END

WEB
    TEMPLATE global.html
    IMAGEPATH "/var/www/ms_tmp/"
    IMAGEURL "/ms_tmp/"
END

# Layer de fondo
LAYER
    NAME Nicaragua
    TYPE POLYGON
    STATUS DEFAULT
    DATA nic_admin

    CLASSITEM 'DEPTO'
    CLASS
```

```
    NAME 'Boaco'
    EXPRESSION 'Boaco'
    STYLE
        OUTLINECOLOR 100 100 100
        COLOR 180 180 180
    END
END

CLASS
    NAME 'Carazo'
    EXPRESSION 'Carazo'
    STYLE
        OUTLINECOLOR 100 100 100
        COLOR 210 210 190
    END
END

CLASS
    NAME 'Chinandega'
    EXPRESSION 'Chinandega'
    STYLE
        OUTLINECOLOR 100 100 100
        COLOR 150 200 170
    END
END

CLASS
    NAME 'Chontales'
    EXPRESSION 'Chontales'
    STYLE
        OUTLINECOLOR 100 100 100
        COLOR 170 125 110
    END
END

CLASS
    NAME 'Esteli'
    EXPRESSION 'Esteli'
    STYLE
        OUTLINECOLOR 100 100 100
        COLOR 100 150 150
    END
END
```



```
CLASS
  NAME 'Granada'
  EXPRESSION 'Granada'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 150 220 150
  END
END
```

```
CLASS
  NAME 'Jinotega'
  EXPRESSION 'Jinotega'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 210 210 110
  END
END
```

```
CLASS
  NAME 'Lago Xolotlan'
  EXPRESSION 'Xolotlan'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 59 106 242
  END
END
```

```
CLASS
  NAME 'Lago Cocibolca'
  EXPRESSION 'Cocibolca'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 59 106 242
  END
END
```

```
CLASS
  NAME 'Leon'
  EXPRESSION 'Leon'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 190 130 130
  END
END
```

```
CLASS
  NAME 'Madriz'
  EXPRESSION 'Madriz'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 180 180 230
  END
END
```

```
CLASS
  NAME 'Managua'
  EXPRESSION 'Managua'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 230 200 180
  END
END
```

```
CLASS
  NAME 'Masaya'
  EXPRESSION 'Masaya'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 230 100 100
  END
END
```

```
CLASS
  NAME 'Matagalpa'
  EXPRESSION 'Matagalpa'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 80 200 80
  END
END
```

```
CLASS
  NAME 'Nueva Segovia'
  EXPRESSION 'Nueva Segovia'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 200 90 180
  END
```

```
END

CLASS
  NAME 'RAAN'
  EXPRESSION 'RAAN'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 90 190 196
  END
END

CLASS
  NAME 'RAAS'
  EXPRESSION 'RAAS'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 190 160 180
  END
END

CLASS
  NAME 'Rio San Juan'
  EXPRESSION 'Rio San Juan'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 180 136 80
  END
END

CLASS
  NAME 'Rivas'
  EXPRESSION 'Rivas'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 186 180 80
  END
END
END # Fin del Layer de Fondo

# Layer de Ciudades
LAYER
  NAME cabeceras
  TYPE POINT
  STATUS OFF
```

```
DATA nic_cabmuni
LABELITEM 'CABEC_MUNI'
LABELMAXSCALE 5000000

CLASSITEM 'CABEC_MUNI'
CLASS
  NAME 'Capital'
  EXPRESSION 'Managua'
  OUTLINECOLOR 110 110 110

  LABEL
    SIZE TINY
    OUTLINECOLOR 255 255 255
    MINFEATURESIZE 40
    POSITION LL
    PARTIALS FALSE
  END
  STYLE
    SYMBOL 'star'
    COLOR 200 100 100
    SIZE 7
  END
END

CLASS
  NAME 'Ciudades'
  OUTLINECOLOR 110 110 110

  LABEL
    SIZE TINY
    OUTLINECOLOR 255 255 255
    MINFEATURESIZE 40
    POSITION LL
    PARTIALS FALSE
  END
  STYLE
    SYMBOL 'circle'
    COLOR 200 100 100
    SIZE 3
  END
END
END # Fin del Layer de Ciudades Principales
```

```
# Layer de Carreteras
LAYER
    NAME vias
    TYPE LINE
    STATUS OFF
    DATA nic_redvial
    CLASSITEM 'NOMBRE'

    CLASS
        NAME 'Carreteras'
        COLOR 255 134 124
    END
END # Fin del Layer de Carreteras

# Layer de Rios
LAYER
    NAME rios
    TYPE LINE
    STATUS OFF
    DATA nic_rios

    CLASSITEM 'NOMBRE'
    CLASS
        NAME 'Rios'
        COLOR 59 106 242
    END
END # Fin del Layer de Rios
END # Fin del Mapa
```

Anexo C - Archivo de Inicialización index.html

```
<html>
<head>
  <title> Mapa de Nicaragua - MapServer </title>
</head>
<body>
  <center> <h2> Mapa de Nicaragua </h2>
  <form method=GET action="/cgi-bin/mapserv">
    <input type="hidden" name="map"
    value="/var/www/nicaragua/nicaragua.map">
    <input type="hidden" name="zoomdir" value=0>
    <input type="hidden" name="zoomsize" value=2>
    <input type="hidden" name="program" value="/cgi-
bin/mapserv">
    <input type="submit" value="Mostrar Mapa">
  </form>
  </center>
</body>
</html>
```

Anexo D - Plantilla global.html

```

<html>
<head><title> Mapa de Nicaragua - MapServer </title></head>
<body><center><h2>Mapa de Nicaragua</h2>
<hr>
<form method=GET action="/cgi-bin/mapserv">
<br> <table width="800" border="1" cellspacing="1" cellpadding="1">

<tr> <!-- Primera Fila-->
<td>Selecci&oacute;n de Layers:<br>
<!-- Especificamos layer vectors -->
<input type="checkbox" name="layer" value="rios" [rios_check]>
Rios&nbsp;<input type="checkbox" name="layer"
value="vias"[vias_check]> Carreteras&nbsp;<
<input type="checkbox" name="layer"
value="cabeceras"[cabeceras_check]> Ciudades<br>
</td>

<td>
<center><input align="middle" type="submit" name="submit"
value="Redibujar Mapa"> </center>
</td>
</tr>

<tr> <!-- Segunda fila -->
<!-- MUESTRA EL MAPA GENERADO POR MAPSERVER -->
<td width="[mapwidth]">
<input name="img" type="image" src="[img]" width="[mapwidth]"
height="[mapheight]" border=0 alt="Mapa de Nicaragua">
</td>

<td rowspan="2" valign="top"><p> Legenda:
<br>
</p>
</td>
</tr>

<tr> <!-- Tercera fila -->
<!-- CONTROLES PARA LA MANIPULACION DEL MAPA -->
<td>
<center>Zoom In <input type=radio name=zoomdir value=1
[zoomdir_1_check]> Re-Centrar

```

```
<input type=radio name=zoomdir value=0 [zoomdir_0_check]> Zoom Out
<input type=radio name=zoomdir value=-1 [zoomdir_-1_check]> Tamaño del
Zoom
<input type=text name=zoomsize size=4 value=[zoomsize]>
</center>
</td>
</tr>

</table>

<input type="hidden" name="imgxy" value="[center_x] [center_y]">
<input type="hidden" name="imgext" value="[mapext]">
<input type="hidden" name="map" value="[map]">
<input type="hidden" name="savequery" value="true">
<input type="hidden" name="mapext" value="shapes">

</form></center>
</body>
</html>
```


Anexo E – PHP MapScript archivo index.php

```
<?php
//-----
//img2map - convierte coordenadas de imagen a coordenadas de mapa
function img2map($width,$height,$point,$ext){
    //Se requiere un punto válido
    if ($point->x && $point->y)
    {
        //Calcula grados por pixel
        $dpp_x = ($ext->maxx - $ext->minx)/$width;
        $dpp_y = ($ext->maxy - $ext->miny)/$height;

        //Calcula las coordenadas en el mapa
        $p[0] = $ext->minx + $dpp_x*$point->x;
        $p[1] = $ext->maxy + $dpp_y*$point->y;
    }
    return $p;
}//-----

//Valores predeterminados
$script_name = "index.php";

//Path
$map_path = "/var/www/nicaraguaphp/";
$map_file = "nicaragua.map";
$img_path = "/var/www/ms_tmp/";

//Navegación
$zoomsize=2;
$pan="CHECKED";
$zoomout="";
$zoomin="";

//Muestra los layers
$rios = "";
$vias = "";
$cabeceras = "";
$nicaragua = "CHECKED";
```

```
//Punto de click predeterminado
$clickx = 320;
$clicky = 240;
$clkpoint = ms_newPointObj();
$sold_extent = ms_newRectObj();

//Extension Predeterminada
$extent = array(-89,10,-83,16);
$max_extent = ms_newRectObj();
$max_extent->setExtent(-89,10,-83,16);

//Obtener un mapfile y crear un mapa desde el
$map = ms_newMapObj($map_path.$map_file);

//Si se invocó desde el form, usar la variables del form
//sino continuar y crear el mapa
if (( $_POST['img_x'] and $_POST['img_y']) or $_POST['refresh'])
{
    if ($_POST['refresh'])
    {
        $clickx = 320;
        $clicky = 240;
    }
    else
    {
        $clickx = $_POST['img_x'];
        $clicky = $_POST['img_y'];
    }

    //Establecer la ubicación del mouse click (necesario para el
zoom)
    $clkpoint->setXY($clickx,$clicky);

    //Cambiaron los layers seleccionados?
    if ($_POST['layer'])
    {
        $layer = join(",$_POST['layer']");
    }
    else
    {
        $layer = "";
    }
    $this_layer = 0;
```

```
if (preg_match("/rios/", $layer))
{
    $rios = "CHECKED";
    $this_layer = $map->getLayerByName('rios');
    $this_layer->set('status', MS_ON);
}
else
{
    $rios = "";
    $this_layer = $map->getLayerByName('rios');
    $this_layer->set('status', MS_OFF);
}
if (preg_match("/vias/", $layer))
{
    $vias = "CHECKED";
    $this_layer = $map->getLayerByName('vias');
    $this_layer->set('status', MS_ON);
}
else
{
    $vias = "";
    $this_layer = $map->getLayerByName('vias');
    $this_layer->set('status', MS_OFF);
}
if (preg_match("/cabeceras/", $layer))
{
    $cabeceras = "CHECKED";
    $this_layer = $map->getLayerByName('cabeceras');
    $this_layer->set('status', MS_ON);
}
else
{
    $cabeceras = "";
    $this_layer = $map->getLayerByName('cabeceras');
    $this_layer->set('status', MS_OFF);
}

//Obtener la extensión del mapa mostrado
if ($_POST['extent'])
{
    $extent = split(" ", $_POST['extent']);
}
```

```
//Establecer el mapa a la extensión obtenida
$map->setextent($extent[0],$extent[1],$extent[2],$extent[3]);

//Guardar esta extensión como un rectObj, necesario para el zoom
$sold_extent->setextent($extent[0],$extent[1],
                      $extent[2],$extent[3]);

//Calcular el factor de zoom y pasarlo el método zoomPoint
//
// zoomfactor = +/- N
// if N > 0 zoom in - N < 0 zoom out - N = 0 pan
//
$zoom_factor = $_POST['zoom']*$_POST['zsize'];

// Establece la dirección del zoom
if ($zoom_factor == 0){
    $zoom_factor = 1;
    $pan = "CHECKED";
    $zoomout = "";
    $zoomin = "";
} elseif ($zoom_factor < 0){
    $pan = "";
    $zoomout = "CHECKED";
    $zoomin = "";
} else {
    $pan = "";
    $zoomout = "";
    $zoomin = "CHECKED";
}
$zoomsize = abs( $_POST['zsize']);

//Zoom in/out en el clkpoint
$map->zoomPoint($zoom_factor,$clkpoint,$map->width,
              $map->height,$old_extent,$max_extent);
}

//El script fue invocado directamente
//o se ha finalizado el pan, zoom o seleccionado layers

//Establecer nombres únicos para el mapa, referencia y leyenda
$map_id = sprintf("%0.6d",rand(0,999999));

$image_name = "nic".$map_id.".png";
$image_url = "/ms_tmp/".$image_name;
```

```
$ref_name = "nicref".$map_id.".gif";
$ref_url = "/ms_tmp/".$ref_name;

$leg_name = "nicleg".$map_id.".png";
$leg_url = "/ms_tmp/".$leg_name;

//Dibujar y guardar la imagen del mapa
$image=$map->draw();
$image->saveImage($img_path.$image_name);

//Dibujar y guardar la imagen de referencia
$ref = $map->drawReferenceMap();
$ref->saveImage($img_path.$ref_name);

//Dibujar y guardar la imagen de la leyenda
$leg = $map->drawLegend();
$leg->saveImage($img_path.$leg_name);

//Guardar la extensión despues de pan y zoom
//una variable form separada por espacios
$new_extent = sprintf("%3.6f", $max_extent->minx). " "
    .sprintf("%3.6f", $max_extent->miny). " "
    .sprintf("%3.6f", $max_extent->maxx). " "
    .sprintf("%3.6f", $max_extent->maxy);

//Formatear la escala de la imagen a mostrar
$scale = sprintf("%10d", $map->scale);

//Convertir la coordenadas de click a coordenadas de mapas
list($x,$y) = img2map($map->width,$map->height,$clkpoint,$old_extent);
$x_str = sprintf("%3.6f", $x);
$y_str = sprintf("%3.6f", $y);

//Escribir el formulario HTML
?>

<html>
<head><title>Mapa de Nicaragua - PHP MapScript</title></head>
<body>
    <form method=post action="<?php echo $script_name;?>">
        <table width="100%" border="1">
            <tr><td width="60%" rowspan="6">
                <input name="img" type="image" src="<?php echo
$image_url;?>" width=640 height=480 border=2></td>
```



```
        </font></td></tr>
    </table>
</form>
</body>
</html>
```