

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA ENSENADA
MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA



“Algoritmo de selección de datos espaciales en un servidor de mapas”

Tesis:

Que para obtener el grado de

Maestro en Ingeniería

Presenta:

LUIS ANGEL MONGE DE LA CRUZ

ENSENADA BAJA CALIFORNIA AGOSTO DEL 2009

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

**FACULTAD DE INGENIERÍA
UNIDAD ENSENADA**

**“Algoritmo de selección de datos espaciales
en un servidor de mapas”**

TESIS

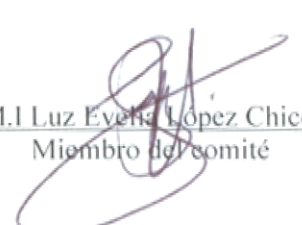
Que para obtener el grado de maestría en ingeniería presenta:

Luis Angel Monge De La Cruz

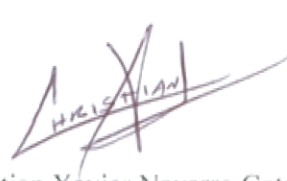
Aprobada por:



M.I. Juan Pablo Torres Herrera
Director de tesis



M.I. Luz Evelyn López Chico
Miembro del comité



M.C. Christian Xavier Navarro Cota
Miembro del comité

Ensenada Baja California, México. Junio 2009

**ASUNTO: Voto aprobatorio sobre trabajo
De tesis de grado de Maestro en ingeniería**

Dr. Juan Iván Nieto Hipólito
Coordinador de Posgrado
Facultad de Ingeniería-Ensenada
P R E S E N T E

Después de haber efectuado una revisión minuciosa sobre el trabajo de tesis presentado por el C. LUIS ANGEL MONGE DE LA CRUZ para poder presentar la defensa de su examen y obtener el grado de Maestro en Ingeniería, me permito comunicarle que he dado mi voto aprobatorio, sobre el trabajo titulado:

“Algoritmo de selección de datos espaciales en un servidor de mapas”

Esperando reciba la presente de conformidad, quedo de Usted.

Ensenada, B.C. a día 6 del mes de junio del 2009



M.I Juan Pablo Torres Herrera

c.c.p expediente

**ASUNTO: Voto aprobatorio sobre trabajo
De tesis de grado de Maestro en ingeniería**

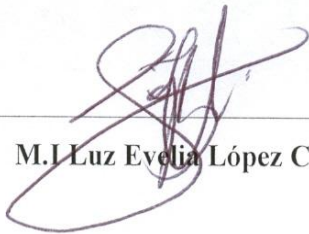
Dr. Juan Iván Nieto Hipólito
Coordinador de Posgrado
Facultad de Ingeniería-Ensenada
P R E S E N T E

Después de haber efectuado una revisión minuciosa sobre el trabajo de tesis presentado por el C. LUIS ANGEL MONGE DE LA CRUZ para poder presentar la defensa de su examen y obtener el grado de Maestro en Ingeniería, me permito comunicarle que he dado mi voto aprobatorio, sobre el trabajo titulado:

“Algoritmo de selección de datos espaciales en un servidor de mapas”

Esperando reciba la presente de conformidad, quedo de Usted.

Ensenada, B.C. a día 6 del mes de junio del 2009



M.I Luz Evelyn López Chico

c.c.p expediente

**ASUNTO: Voto aprobatorio sobre trabajo
De tesis de grado de Maestro en ingeniería**

Dr. Juan Iván Nieto Hipólito
Coordinador de Posgrado
Facultad de Ingeniería-Ensenada
P R E S E N T E

Después de haber efectuado una revisión minuciosa sobre el trabajo de tesis presentado por el C. LUIS ANGEL MONGE DE LA CRUZ para poder presentar la defensa de su examen y obtener el grado de Maestro en Ingeniería, me permito comunicarle que he dado mi voto aprobatorio, sobre el trabajo titulado:

"Algoritmo de selección de datos espaciales en un servidor de mapas"

Esperando reciba la presente de conformidad, quedo de Usted.

Ensenada, B.C. a día 6 del mes de junio del 2009



M.C Christian Xavier Navarro Cota

c.c.p expediente

DEDICATORIA

A mis padres

Martha De La Cruz Casimiro y Rafael Monge Calles por su amor, apoyo incondicional, comprensión y por fomentar en mí sólidos valores que me ayudan a ser mejor persona cada día.

A mis hermanos

Martha, Rafael y Nancy por motivarme a seguir adelante, a entender que no hay límites, que todo se puede lograr si realmente se quiere.

A Alejandro, Marlenne, Stephanie y Johan

Por ser parte de mi familia, por estar presentes siempre.

“Todo es cuestión de resistir y luchar de forma constante, seguir tu propio camino y no el establecido por los demás, defender tus ideales sin claudicar” *Luis Angel Monge*

AGRADECIMIENTOS

A mi director de tesis

M.I Juan Pablo Torres Herrera, por su tiempo y dedicación brindada en el desarrollo del proyecto de tesis, por su apoyo y consejos durante la licenciatura y maestría.

A mis sinodales

M.I Luz Evelia López Chico y M.C Christian Xavier Navarro Cota, por su apoyo, dedicación y enseñanzas durante la licenciatura y maestría.

Al Dr. Fortunato Espinoza Barreras y M.I Jesús Juárez García

Por su valiosa colaboración en el proyecto de tesis con el estudio de vulnerabilidad sísmica de los puentes urbanos de Ensenada

A mis maestros

Dr. Juan Iván Nieto, Dr. Jesús Zamarripa, Dr. Josué Álvarez, M.I Juan Pablo Torres Herrera, M.I Luz Evelia López Chico, M.C Christian Xavier Navarro Cota, M.C Elitania Jiménez García, M.I Haydee Meléndez Guillen, M.I Víctor Velázquez Mejía

Al Dr. Gilles Lavigne, Dr. Arturo Serrano Santoyo, Dr. Lewis McAnally, Dr. Javier Organista y M.T.R.I. Julio César Cano

Por permitirme ser partícipe de sus proyectos, la confianza brindada y apoyo.

A mis compañeros y amigos todos

Texangary Manjarrez, Alejandra Pérez, Victoria Guerrero, Humberto Terríquez, Diana Cárdenas, Cynthia Ornelas, Mayra Montalvo, Natalie Millán, Abraham Abiud Jasmin, Luis Antonio Pérez, Carlos Figueroa, Juan Carlos Leñero, Oscar Agundez, Damián Díaz, Ricardo Aldan, Luis Adrian Lechuga, Rafael Ruiz, Alejandro Ruelas, Adrian Nájera.

A CONACYT

Por su apoyo económico durante la realización de la maestría

“Uno de los mejores placeres es lograr lo que se creía imposible” *Luis Angel Monge*

ALGORITMO DE SELECCIÓN DE DATOS ESPACIALES EN UN SERVIDOR DE MAPAS

En los últimos años, ha existido un incremento en el uso y diversificación de aplicaciones espaciales, Paul Longley atribuye este hecho a tres factores: 1.- los avances tecnológicos en redes y hardware, 2.- la amplia disponibilidad del Internet y 3.- la demanda de acceso de datos espaciales [Longley, 2005]. Hoy en día existen diversas aplicaciones espaciales enfocadas a: delitos, riesgos geológicos, tráfico vehicular, ecosistemas, entre otras. Estas aplicaciones espaciales son soportadas y desarrolladas a partir de servidores de mapas los cuales se clasifican en: comerciales (Ej. ArcGIS Server, MapXtreme), código abierto (Ej. GeoServer, UMN MapServer) y servicio gratuito (Ej. Google Maps, Yahoo Maps).

Los servidores de mapas permiten a los usuarios interactuar con los datos espaciales por medio de una interfaz web (aplicación espacial), posibilitando la visualización y consulta de información geográfica contenida en los datos espaciales, además los servidores de mapas permiten integrar contenido multimedia, ofrecen una forma rápida de proporcionar información geográfica actualizada, son capaces de obtener datos espaciales de diversas fuentes de datos locales y remotas, entre otras.

Sin embargo los servidores de mapas enfrentan un problema muy importante en relación a la disponibilidad de datos espaciales en las fuentes de datos, debido a que cuando un usuario solicita un dato espacial por medio de una aplicación espacial al servidor de mapas, éste último debe obtener el dato espacial de una fuente de datos, pero si el dato espacial no se encuentra en dicha fuente, no se podrá visualizar o consultar el dato hasta que el administrador de la fuente de datos se percate de la no disponibilidad, implicando algunas veces un tiempo considerable en solucionar el problema.

A pesar de que actualmente existen mecanismos que buscan garantizar la disponibilidad de los datos espaciales en las fuentes de datos como: la redundancia de dispositivos de lado del servidor, esquemas de datos espaciales distribuidos y programas de respaldo. Estos mecanismos no resuelven de manera específica la problemática que enfrentan los servidores de mapas; ya que son soluciones temporales, asíncronas y ajenas al control del servidor de mapas sobre las fuentes de datos. Un servidor de mapas requiere soluciones en tiempo real, de forma específica, además de contar con cierto control sobre las fuentes de datos.

Por lo tanto en el presente proyecto de investigación se desarrolló un algoritmo de selección integrado al servidor de mapas el cual es una solución alternativa, específica, en tiempo real y que mejora la disponibilidad de los datos espaciales; ya que el algoritmo es capaz de obtener datos espaciales de fuentes alternas tanto locales como remotas, de forma automatizada y transparente al usuario por medio de dos métodos propuestos: optimista y pesimista, cuando exista la no disponibilidad de un dato espacial en una fuente de datos.

INDICE

CAPITULO 1. INTRODUCCIÓN

	pág.
1.1 Antecedentes	1
1.2 Planteamiento del problema.	4
1.3 Pregunta de investigación.	5
1.4 Hipótesis	5
1.5 Objetivos	6
1.6 Importancia del estudio	7
1.7 Limitaciones del estudio	8

CAPITULO 2. ESTADO DEL ARTE

2.1 Datos espaciales	9
2.2 Servidores de mapas	10
2.3 Características representativas de servidores de mapas de código abierto y comercial	12
2.4 Tipos de usuarios en servidores de mapas	15
2.5 Servidores de mapas de servicios gratuitos	16
2.6 Diferencia de servidores de mapas de código abierto, comerciales y servicios gratuitos	18
2.7 Tipos de aplicaciones espaciales en entornos web para servidores de mapas	19
2.7.1 Aplicación espacial enfocada a delitos	19
2.7.2 Aplicación espacial para el control del tráfico vehicular	20
2.7.3 Aplicación espacial para la disponibilidad de propiedades	21
2.7.4 Aplicación espacial enfocada a incendios	22
2.7.5 Aplicación espacial para ecosistemas	23
2.7.6 Aplicación espacial enfocada a finanzas	24
2.7.7 Aplicación espacial para el control epidemias	25
2.7.8 Aplicación espacial para la agricultura	26
2.7.9 Aplicación espacial de áreas de recreación	27
2.7.10 Aplicación espacial de riesgos geológicos-volcanes	28
2.7.11 Aplicación espacial de riesgos geológicos-sísmicos	29
2.7.12 Aplicación espacial para la investigación y planeación urbana	30
2.8 Mecanismos que buscan garantizar las fuentes de datos espaciales	31
2.8.1 Redundancia de dispositivos	31
2.8.2 Respaldo de datos	33
2.8.3 Datos espaciales distribuidos	34

CAPITULO 3. METODOLOGIA

3.1 Metodología de desarrollo de software	36
3.2 Creación de prototipos	38

CAPITULO 4. BOSQUEJO

4.1 Descripción general del administrador	39
4.2 Descripción general de los especialistas	39
4.3 Descripción general del proyecto	40
4.4 Descripción general de común acuerdo: desarrollador, especialistas y proyecto	40

CAPITULO 5. ESPECIFICACIÓN

5.1	Requerimientos de aplicación espacial	41
5.2	Requerimientos del algoritmo	42
5.2.1	Esquemas de funcionamiento de los métodos OPTIMISTA y PESIMISTA	43
5.3	Requerimientos de usuarios	44
5.4	Requerimientos adicionales	45
5.5	Fundamentación de la selección del servidor de mapas	46
5.6	Servidor de mapas UMN MapServer	48
5.6.1	Funcionamiento integral de los elementos del servidor de mapas	51
5.6.2	¿Qué es el MAPFILE?	53
5.6.3	MAPFILE: Estructura interna	54
5.6.4	MAP	55
5.6.5	LAYER	56
5.6.6	LAYER:CLASS	57
5.6.7	LAYER:CLASS:LABEL	58
5.6.8	LAYER:CLASS:STYLE	59
5.6.9	LAYER:CLASS:SYMBOL	60
5.6.10	LAYER:CLASS: Etiquetas claves independientes	61
5.6.11	LAYER:GRID	61
5.6.12	LAYER:JOIN	62
5.6.13	LAYER:FEATURE	62
5.6.14	LAYER: Etiquetas claves independientes	63
5.6.15	LEGEND	64
5.6.16	QUERYMAP	64
5.6.17	REFERENCE	65
5.6.18	WEB	65
5.6.19	SCALEBAR	66
5.6.20	Comentarios, etiquetas de finalización e inicialización en el MAPFILE	66
5.7	Casos de uso	67
5.8	Hardware y software	69
5.9	Configuración de la red de trabajo	70

CAPITULO 6. DESARROLLO

Fase 1: Diseño y desarrollo de aplicación espacial

6.1	Digitalización	72
6.2	Distribución del contenido	76
6.3	Funcionamiento para la consulta de datos	77
6.4	Catalogo de imágenes	78

Fase 2: Diseño de esquema de selección de datos espaciales

6.5	¿Dónde y cómo se adaptara el esquema de selección de datos espaciales?	81
6.6	LAYER: Nuevas etiquetas claves independientes	83
6.7	Crear etiqueta objeto ALTERNATIVE	85
6.8	Diseño específico esquema de selección de datos espaciales: Resultado final de la interpretación de requerimientos	87
6.9	Identificación de fuente primaria y fuente alterna en el MAPFILE	88

Fase 3: Desarrollo de algoritmo de selección de datos espaciales

6.10	Pruebas a entornos de desarrollo	89
6.11	Interpretación del esquema general y esquema específico a pseudocódigo	92
6.12	Identificación de los archivos fuentes del servidor de mapas a modificar y forma de implementar el esquema de selección de datos espaciales.	93

CAPITULO 7. VALIDACIÓN

7.1	Validación: pruebas de ACEPTACIÓN y ALFA	97
7.2	Pruebas de aceptación	98
7.3	Pruebas ALFA	99

CAPITULO 8. CONCLUSIONES

8.1	Resultados Finales	104
8.2	Conclusiones	107
8.3	Aportaciones	108
8.4	Trabajo Futuro	109

BIBLIOGRAFÍAS**ANEXO**

Manual de desarrollo para el servidor de mapas UMN MapServer

LISTA DE FIGURAS

Fig.	Título	Pág.
1.1	Capas sobrepuestas	2
1.2	Problemática que enfrenta un servidor de mapas	4
2.1	Imagen raster	9
2.2	Imagen vectorial	9
2.3	Aplicación espacial enfocada a delitos reportados en Clearwater Florida, Estados Unidos	19
2.4	Aplicación espacial enfocada a tráfico vehicular de Houston Texas, Estados Unidos	20
2.5	Aplicación espacial enfocada a disponibilidad de propiedades en Savannah y el condado de Chatam en Atlanta, Estados Unidos	21
2.6	Aplicación espacial en un entorno web enfocada a la gestión de incendios en Estados Unidos	22
2.7	Aplicación espacial enfocada al ecosistema del Golfo de Maine, Estados Unidos.	23
2.8	Aplicación espacial web enfocada a finanzas globales	24
2.9	Aplicación espacial enfocada a la detección de epidemias en Wurzburg Alemania	25
2.10	Aplicación espacial enfocada a la agricultura en España	26
2.11	Aplicación espacial enfocada a áreas recreativas de Minnesota, Estados Unidos	27
2.12	Aplicación espacial enfocada al riesgo geológico del volcán Popocatepetl en México	28
2.13	Aplicación espacial enfocada a riesgo geológico en sismos en el Distrito Federal México	29
2.14	Aplicación espacial enfocada a la investigación y planeación urbana de Ensenada Baja California, México	30
2.15	Esquema distribuido de bases de datos en la actualidad	34
2.16	Datos espaciales compartidos bajo un esquema de área local	35
3.1	Metodología de desarrollo evolutivo	36
3.2	Esquema de construcción de prototipos	38
5.1	Funcionamiento del método OPTIMISTA	43
5.2	Funcionamiento del método PESIMISTA	43
5.3	Arquitectura del servidor de mapas UMN MapServer	50
5.4	Funcionamiento integrado de los elementos del servidor de mapas	52
5.5	Ejemplo de archivo Mapfile	53
5.6	Estructura del archivo del Mapfile (.MAP)	54
5.7	Etiquetas claves independientes que pertenecen a MAP	55
5.8	Estructura de etiquetas objetos y etiquetas claves independientes que pertenecen a LAYER	56
5.9	Estructura de etiquetas objetos y etiquetas claves independientes que pertenecen a CLASS	57
5.10	Estructura de la etiqueta objeto LABEL	58
5.11	Estructura de la etiqueta objeto STYLE	59
5.12	Estructura de la etiqueta objeto SYMBOL	60
5.13	Etiquetas claves independientes que contiene la etiqueta objeto CLASS	61
5.14	Etiquetas claves independientes que contiene la etiqueta objeto GRID	61
5.15	Etiquetas claves independientes que contiene la etiqueta objeto JOIN	62
5.16	Etiquetas claves independientes que contiene la etiqueta objeto FEATURE	62
5.17	Etiquetas claves independientes que contiene la etiqueta objeto LAYER	63
5.18	Etiquetas claves independientes que contiene la etiqueta objeto LEGEND	64
5.19	Etiquetas claves independientes que contiene la etiqueta objeto QUERYMAP	64
5.20	Etiquetas claves independientes que contiene la etiqueta objeto REFERENCE	65
5.21	Etiquetas claves independientes que contiene la etiqueta objeto WEB	65
5.22	Estructura de la etiqueta objeto SCALEBAR	66
5.23	Caso de uso del cliente	67
5.24	Caso de uso del administrador	68
5.25	Configuración de la red de trabajo	70
6.1	Principales fallas sísmicas de la región de Baja California, México.	71

6.2	Epicentros de temblores de la región de Baja California, México registrados en el transcurso del año 2005	72
6.3	Integración de capas de la ciudad de Ensenada, BC y puentes	74
6.4	Integración de capas del poblado del Sauzal en Ensenada, BC y puentes.	74
6.5	Integración de capas del Ex Ejido Chapultepec en Ensenada, BC., y puente	75
6.6	Interfaz de edición de tablas DBF. 1.- Habilitar opción de edición de tablas. 2.- Introducir datos en editor	75
6.7	Distribución de las secciones de la aplicación espacial	77
6.8	Resultado de la aplicación espacial	78
6.9	Catalogo de fotos en la aplicación espacial	78
6.10	Esquema de selección de datos espaciales	79
6.11	Identificación del lugar específico para implementar el esquema de selección de fuentes de datos espaciales	82
6.12	Creación de etiquetas claves independientes para la etiqueta objeto LAYER	84
6.13	Creación de etiqueta objeto ALTERNATIVE	85
6.14	Declaración múltiple de fuentes alternas ALTERNATIVE	87
6.15	Esquema adaptado a la estructura interna del Mapfile	87
6.16	Identificación de fuente primaria y alterna en el Mapfile	88
7.1	Entorno controlado para la realización de pruebas ALFA	106
7.2	El servidor de mapas UMN MapServer detecta el esquema específico propuesto	107
7.3	Registro de actividades: detección de fuentes	107
7.4	Visualización del informe de actividades	110
7.5	Declarar CONFIG a DEBUG para generar informe de actividades	110
8.1	Aplicación espacial	112
8.2	Esquema general de selección de datos espaciales	113
8.3	Esquema específico de selección de datos espaciales	113
8.4	Sitio web GeoConstructor	114
9.1	Instalación del depurador y entorno de gráfico	120
9.2	Abrir entorno gráfico KDbg	120
9.3	Adjuntar ejecutable mapserv desde KDBG	121
9.4	Resultado obtenido a partir del ejecutable mapserv	122
9.5	Adjuntar archivo fuente desde KDBG	122
9.6	Resultado obtenido al adjuntar un archivo fuente	123
9.7	Forma de introducir valores de prueba en KDbg	124
9.8	Ubicación de los valores de prueba	125
9.9	Archivo Mapfile contiene los valores de prueba	125
9.10	Agregar watch y breakpoint en KDbg	127
9.11	Editor de texto vía línea de comandos de CentOS 5	128
9.12	Compilación de UMN MapServer por línea de comandos	129
9.13	Diagrama para el desarrollo de código en el servidor de mapas UMN MapServer	130

LISTAS DE TABLAS

Tabla	Título	pág.
1.1	Ejemplos de servidores de mapas	1
1.2	Algunos factores que pueden ocasionar la no disponibilidad de datos espaciales en fuentes de datos	2
2.1	Lista de características representativas de servidores de mapas de código abierto y comerciales	14
2.2	Lenguajes de programación soportados por las APIs de las compañías proveedoras de servicios gratuitos	17
2.3	Clasificación de redundancia de dispositivos del lado del servidor físico	31
2.4	Cotización aproximada de dispositivos redundantes	32
2.5	Software para el respaldo de datos	33
2.6	Soporte espacial para bases de datos	34
5.1	Requerimientos de aplicación espacial	41
5.2	Requerimientos del algoritmo	42
5.3	Requerimientos de usuarios	44
5.4	Requerimientos adicionales	45
5.5	Análisis comparativo de servidores de mapas de código abierto	47
5.6	Caso de uso cliente consultar información geográfica	67
5.7	Caso de uso cliente visualizar datos espaciales	67
5.8	Caso de uso administrador configurar Mapfile	68
5.9	Caso de uso administrador visualizar informe	68
6.1	Datos de los puentes urbanos de la ciudad de Ensenada y resultados del estudio de vulnerabilidad	73
6.2	Entornos de desarrollo para la programación, depuración y compilación del código fuente, calificados del 1 al 10 de acuerdo a la funcionalidad y adaptabilidad para trabajar con el código fuente de UMN Mapserver	90
6.3	Archivos fuentes modificados del servidor de mapas UMN MapServer	93
6.4	Tabla de funciones implementadas en el mapfile.c	95
6.5	Función implementada en el archivo mapdraw.c	96
7.1	Cumplimiento de expectativas y requerimientos	98
7.2	Características del entorno controlado	99
7.3	Validación de etiquetas y mensajes de error	108
7.4	Validación de fuentes primaria y alternas no están disponibles	108
7.5	Validación de etiquetas y mensajes de error (ALTERNATIVE)	109

Introducción

1

En este capítulo se abordan los antecedentes del tema de investigación, se define la problemática que enfrentan los servidores de mapas en relación a la obtención de datos espaciales, pregunta de investigación e hipótesis, así como también el objetivo general y específicos. Para concluir este capítulo se plantea la importancia de desarrollar el presente proyecto de investigación y sus limitaciones.

1.1 Antecedentes

En los últimos años, ha existido un incremento en el uso y diversificación de aplicaciones espaciales¹, Paul Longley atribuye este hecho a tres factores: 1.-los avances tecnológicos en redes y hardware, 2.- la amplia disponibilidad del Internet y 3.- la demanda de acceso de datos espaciales [Longley, 2005].

Hoy en día, existen diversas aplicaciones espaciales enfocadas a: delitos, tráfico vehicular, disponibilidad de propiedades, incendios, ecosistemas, finanzas, epidemias, agricultura, áreas de recreación, riesgos geológicos, investigación y planeación urbana, entre otros (Ver Capítulo 2). Christian Harder afirma que la finalidad de las aplicaciones espaciales es ayudar a responder preguntas o problemas relacionados con información geográfica [Harder, 1998].

De acuerdo con Pau Serra para el funcionamiento de las aplicaciones espaciales es necesario contar un servidor de mapas que permita a los usuarios interactuar con los datos espaciales por medio de una aplicación cliente [Serra, 2002]; esta interacción involucra los aspectos visualización, superposición, consulta de datos espaciales, entre otros.

Actualmente hay una diversidad de servidores de mapas comerciales, código abierto y de servicios gratuitos que están al alcance de la mayoría de las personas (Ver Tabla 1.1); en comparación de hace una década la situación era muy distinta, Andreas Neumann comenta que el costo y la complejidad del hardware y software necesario para el uso y desarrollo de aplicaciones espaciales, era una barrera para usuarios y desarrolladores independientes; ya que la adquisición de software y hardware era por parte de pocas compañías, institutos y agencias [Neumann, 2008]. La situación planteada anteriormente provocaba que existiera poco desarrollo de aplicaciones espaciales; a partir de iniciativas de código abierto o de servicios gratuitos fue posible que el conjunto de desarrolladores independientes, compañías, institutos y agencias, pudieran adquirir y mejorar el funcionamiento de algún software.

Comerciales	Código Abierto	Servicios Gratuitos
Autodesk Mapguide Enterprise	UMN MapServer	Yahoo Maps
ArcGIS Server, ArcIMS.	GeoServer	Google Maps
MapXtreme	MapGuide	Live Search Maps

Tabla 1.1. Ejemplos de servidores de mapas

Los servidores de mapas comerciales y de código abierto pueden obtener datos espaciales de varias fuentes de datos: remotas y locales, almacenados en bases de datos y directamente de archivos:

¹ Aplicación cliente que permite interactuar con datos espaciales a través de una interfaz web

raster y vectoriales; lo que permite a un servidor de mapas combinar los datos espaciales de varias fuentes, las cuales finalmente son presentadas al usuario en una aplicación cliente a manera de capas superpuestas (Figura 1.1). En el caso de los servicios gratuitos como: Live Search Maps, Google Maps y Yahoo Maps se desconoce su funcionamiento y arquitectura interna; debido a que estos servicios solo permiten la manipulación del lado del cliente con la ayuda de APIs.

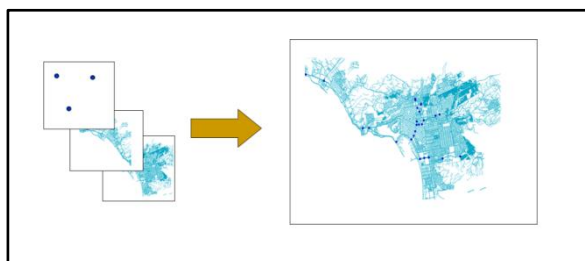


Figura 1.1. Capas superpuestas

Una aspecto importante de la interacción con los datos espaciales en un servidor de mapas es la fuente de los datos; regularmente recae la responsabilidad en el proveedor de servicios de mapas el proporcionar las fuentes de datos y tratar de garantizar su disponibilidad; sin embargo se enfrentan a diversas situaciones que pueden estar fuera de su alcance (Ver Tabla 1.2).

Factor	Ejemplo
Falla del hardware	Procesador, disco duro, fuente de poder, entre otros.
No tener privilegios de lectura	Base de datos, dato espacial.
Falla del software	Sistema operativo o base de datos.
El dato espacial no existe	Fue borrado, cambio de nombre o nunca existió.
Error humano en configuración de fuentes de datos	Nombre del dato espacial, base de datos o tabla, la dirección IP o puerto, password o usuario de la base de datos.
Externo	La pérdida de la señal de internet, corte electricidad.

Tabla 1.2. Algunos factores que pueden ocasionar la no disponibilidad de datos espaciales en fuentes de datos

En la actualidad se busca garantizar la disponibilidad de las fuentes de datos espaciales con algunos mecanismos como: 1.- Redundancia de dispositivos del lado del servidor, 2.- Esquemas datos espaciales distribuidos y 3.- Programas para realizar respaldos de datos (Ver Capítulo 2). Ian Sommerville define disponibilidad como la probabilidad de que un sistema este funcionando y brinde los servicios requeridos por los usuarios [Sommerville, 2002].

Sin embargo ninguno de los mecanismos mencionados, permiten que los servidores de mapas tengan cierto control en la disponibilidad de las fuentes de datos, además son soluciones

temporales y asíncronas. Normalmente si no está disponible una fuente de datos, el proveedor de servicios elimina la fuente de datos del servidor de mapas y proporciona manualmente una nueva que contenga el dato espacial requerido o bien se corrige el problema que causa la no disponibilidad del dato espacial, teniendo en cuenta que el problema puede ser variado y que algunas veces implica dedicar un tiempo considerable para solucionarlo (Ver Tabla 1.2).

La situación planteada anteriormente es un problema; ya que, si un usuario no tiene acceso al servidor de mapas y necesita visualizar un dato espacial y este no se encuentra disponible, tendría que esperarse a que la persona responsable de administrar el servidor de mapas, se percate de este problema y lo solucione.

1.2 Planteamiento del problema

Los servidores de mapas pueden obtener datos espaciales de diferentes fuentes de datos: de forma remota y local, almacenados en bases de datos y directamente de archivos: raster y vectorial. Cuando se realiza la petición de un dato espacial al servidor de mapas por medio de una aplicación espacial, el servidor de mapas utiliza las fuentes de datos que fueron previamente configuradas él, para obtener el dato espacial requerido.

Sin embargo, si el dato espacial no está disponible en la fuente de datos; la aplicación espacial no mostrará el dato espacial (ver Figura 1.2); esto se debe a que los servidores de mapas no contemplan algún método que permita:

1. Identificar la falla de una fuente de datos
2. Proporcionar de manera automática y transparente al usuario una fuente alterna para obtener el dato espacial requerido en caso que la fuente primaria no se encuentre disponible o falle.

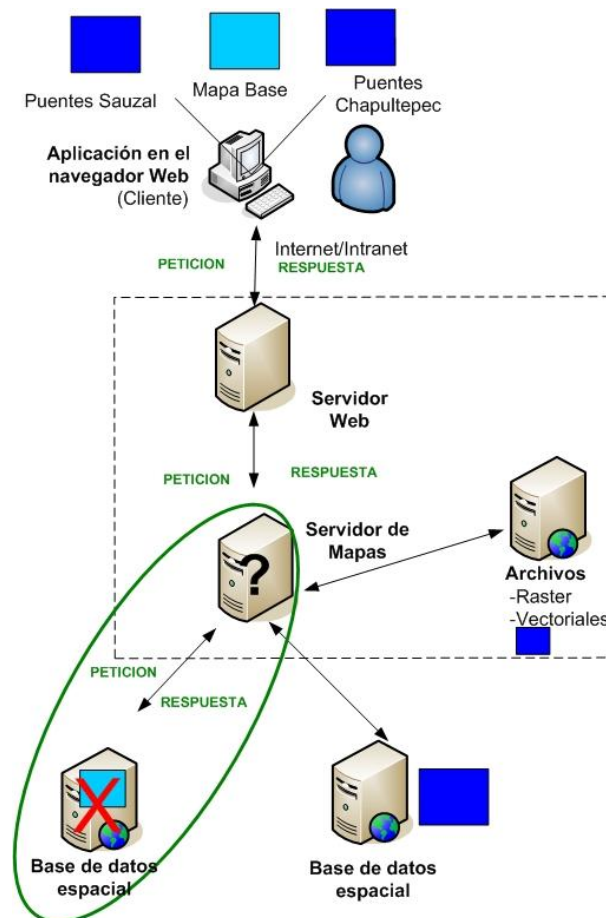


Figura 1.2. Problemática que enfrenta un servidor de mapas

1.3 Pregunta de Investigación

¿Cómo puede obtener un servidor de mapas datos espaciales que se encuentren distribuidos físicamente, alternos a la fuente primaria, de forma automatizada y transparente al usuario?

1.4 Hipótesis

Con el desarrollo de un algoritmo integrado a un servidor de mapas, que realice una selección automatizada y transparente de datos espaciales alternos a la fuente primaria.

1.5 Objetivo

Desarrollar un algoritmo de selección de datos espaciales para un servidor de mapas, que permita obtener datos espaciales de manera automatizada, transparente al usuario y de fuentes alternas distribuidas físicamente.

Para ello se deben cumplir los siguientes objetivos específicos:

1. Analizar e identificar los requerimientos del algoritmo.
2. Diseñar aplicación espacial y esquema de selección de datos espaciales
3. Desarrollar algoritmo de selección de datos espaciales para un servidor de mapas.
4. Evaluar el algoritmo con la ayuda de la aplicación espacial.

1.6 Importancia del estudio

Actualmente existen mecanismos que buscan garantizar la disponibilidad de los datos espaciales: 1.- redundancia de dispositivos del lado del servidor, 2.- Esquemas de datos espaciales distribuidos y 3.- Programas para realizar respaldos de datos. Sin embargo estos mecanismos no resuelven de manera específica la problemática que enfrentan los servidores de mapas en relación a la obtención de datos espaciales ante la falla de una fuente de datos.

Por lo tanto el algoritmo de selección de datos espaciales propuesto y que está integrado al servidor de mapas UMN MapServer, ofrece una solución alternativa, específica, en tiempo real y que mejora la disponibilidad de los datos espaciales; ya que permite al servidor de mapas tener varias fuentes alternas de cada dato espacial lo que implica que si la fuente primaria falla, el algoritmo es capaz de obtener el mismo dato espacial de fuentes alternas, de forma automatizada y transparente al usuario, siempre y cuando estas sean declaradas.

Ayudará a particulares, empresas, organizaciones o dependencias proveedoras de servicios geográficos mejorar la disponibilidad de los datos espaciales en servidores de mapas. Cabe mencionar que el grado de impacto negativo o repercusión que se tenga al no estar disponible un dato espacial cuando este sea requerido, dependerá del enfoque y relevancia de la aplicación espacial. Pudiendo afectar de diversas formas a la organización, empresa o dependencia que preste servicios a través de servidores de mapas y a los mismos usuarios, estas repercusiones pueden ir desde la no visualización de un dato espacial hasta las pérdidas económicas, daños en propiedades y recursos naturales, muertes, entre otros. Es por eso que es importante contemplar métodos que ayuden a mejorar la disponibilidad de los datos espaciales en servidores de mapas.

1.7 Limitaciones del estudio

Para el desarrollo del proyecto:

- Se utilizó software de código abierto, por lo tanto no implicó ningún costo económico.
- Se trabajó en un ambiente controlado en la Universidad Autónoma de Baja California, Facultad de Ingeniería Ensenada bajo un entorno de red de área local, utilizando computadoras personales.

Los datos espaciales:

- Son de tipo vectorial (.SHP).
- Pertenecen a un estudio de vulnerabilidad sísmica de los puentes urbanos de Ensenada [Juárez et. al., 2007], este estudio fue realizado por especialistas en la gestión de riesgos.

El algoritmo:

- No verifica la integridad de los datos espaciales.
- Obtiene los datos espaciales a partir de las fuentes primarias y alternas que sean definidas por el administrador en el servidor de mapas.
- Obtiene los datos espaciales de las fuentes alternas con la ayuda de dos métodos propuestos en este proyecto de investigación: OPTIMISTA y PESISMISTA.
- Se implementó en el servidor de mapas de código abierto UMN MapServer 5.4.2.

Estado del Arte

2

Se plantean y analizan cuatro temáticas que involucran al presente proyecto de investigación: datos espaciales, servidores de mapas, aplicaciones espaciales y mecanismos que buscan garantizar la disponibilidad de las fuentes de datos. El motivo de investigar las cuatro temáticas antes mencionadas, se debe a que no se encontraron métodos o algoritmos que estuvieran directamente relacionados con el proyecto de investigación, sin embargo las cuatro temáticas abordadas en este capítulo contribuyen al entendimiento y desarrollo del proyecto.

2.1 Datos espaciales

Los datos espaciales contienen información sobre la forma de un objeto geográfico, localización geográfica y la relaciones entre ellos, por lo general son almacenados como coordenadas y topología [Heather, 2001]. Los datos espaciales son representados en dos formatos raster y vectorial [Longley et al, 2005]. Algunos ejemplos son:

- Formatos raster son: JPG, GIF, PNG, TIFF (Ver Figura 2.1).
- Formatos vectoriales: SHP, SVF, DWF, SVG, GML (Ver Figura 2.2).

Los datos espaciales en formato vectorial tienen ciertas ventajas en comparación de los raster; según ESRI los datos espaciales generados en formato vectorial tienen menor tamaño, pueden editarse, son rápidos de dibujar, fáciles de leer y escribir [ESRI, 1998]; las ventajas que proporcionan los formatos vectoriales impactan cuando el medio para interactuar con los datos espaciales es un entorno Web; ya que el tiempo de respuesta o de visualización de los datos tiende a reducirse debido a las características que poseen. Por otro lado los datos espaciales en formato raster tienen la ventaja de brindar una imagen más cercana a la realidad, influyendo la calidad de la imagen con el incremento en el tamaño del dato espacial lo cual tiende a aumentar el tiempo de respuesta cuando se solicita el dato espacial en un entorno Web.



Figura 2.1. Imagen raster .



Figura 2.2. Imagen vectorial.

2.2 Servidores de mapas

Hoy en día existe una diversidad de servidores de mapas comerciales, de código abierto y de servicios gratuitos. Tyler Mitchell define a un servidor de mapas como el motor que permite la visualización de mapas en una página Web [Mitchell, 2005]. Los mapas son generados a partir de datos espaciales que se encuentran almacenados de forma local o remota, es decir los servidores de mapas tienen la capacidad de integrar datos especiales diversas fuentes de datos.

Los servidores de mapas contribuyen a facilitar de forma rápida y accesible datos espaciales a través de la web [Padrón et al, 2003]. El uso de la Web como medio de diseminación de mapas puede ser considerado como de los mayores avances en relación a la cartografía y ha abierto nuevas oportunidades como: el desarrollo de aplicaciones espaciales tiempo real, actualizaciones de datos y software con mayor frecuencia y más baratos, la distribución de fuentes de datos y el compartimiento de información geográfica, entre otros [Neumann, 2008]. Sin embargo esto ha implicado muchos retos debido a restricciones técnicas, entre las cuales destaca Andreas Neumann: los derechos de autor, el ancho de banda limitado para algunos dispositivos, cuestiones de seguridad y confiabilidad, entre otros [Neumann, 2008].

Según Tyler Mitchell las características generales que poseen los servidores de mapas son: 1.- generación mapas, 2.-superposición visual de capas de datos espaciales en formato raster o vectorial,3.- capacidad de respuestas a peticiones relacionadas con información descriptiva de los datos espaciales que son visualizados, 4.-las capacidades de geoprocésamiento en referencia a los cambios de proyección geográfica, inserción y edición de nuevos elementos espaciales y 5.- gestión de bases de datos descriptivas [Mitchell,2005].

Una parte importante de los servidores de mapas son precisamente los mapas generados a partir de datos espaciales, actualmente estos presentan diversas ventajas y desventajas [Neumann, 2008], Andreas Neumann las define de la siguiente manera:

Ventajas

- Ofrecen una forma rápida de proporcionar información actualizada.
- El hardware necesario para implementar un servidor de mapas es barato.
- Funcionan bajo diversos sistemas operativos y navegadores Web
- Es posible la integración de contenido multimedia y la inserción de enlaces o hiperligas
- Pueden combinar datos espaciales de diferentes fuentes de datos
- Tienen la capacidad de permitir la personalización, configuración y diseño de mapas.

- Están habilitados para trabajar bajo un ambiente colaborativo. Ejemplo: OpenStreetMap Project, permite la edición y creación de mapas libres de forma colaborativa [openstreetmap,2009]

Desventajas

- La fiabilidad del internet y la infraestructura de servidores de Web, aun no son suficientemente buenas. Especialmente si los mapas dependen de fuentes de datos externas y fuentes de datos distribuidas, el autor original no puede garantizar la disponibilidad de la información.
- A pesar de que existen herramientas libres y comerciales para crear aplicaciones espaciales aun siguen siendo complicados de desarrollar.
- La adquisición de datos espaciales es cara, sin embargo existen algunas instituciones u organizaciones en el mundo que los proporcionan de manera gratuita.
- Muchas personas siguen renuentes a la publicación de datos espaciales, ya que en algunas partes del mundo son caros.
- en dispositivos móviles
- El limitado espacio de la pantalla de dispositivos móviles para visualización de datos espaciales.

2.3 Características representativas de servidores de mapas de código abierto y comercial

Para la realización de la tabla comparativa de características de servidores de mapas de código abierto y comercial (Ver Tabla 2.1) se consultaron las páginas webs oficiales. Esta tabla está dividida en 8 categorías: nombre, tipo, plataforma, desarrollo interno, desarrollo externo, soporte de bases de datos, estándares y formatos soportados. Los resultados obtenidos mediante un análisis comparativo son los siguientes:

- De los 11 servidores de mapas, 6 son comerciales y 5 de código abierto.
- De los 11 servidores de mapas, solo 4 podían utilizarse en más de 4 plataformas distintas, 3 servidores de mapas corresponden a licencias comerciales: ArcIMS, ArcGIS Server y MapInfo MapXtreme; y uno de código abierto UMN MapServer.
- De los 11 servidores de mapas solo 5 permiten el desarrollo interno: GeoServer, UMN MapServer, Mapnik, MapGuide OS, Deegree, todos los anteriores de código abierto. Los lenguajes de programación que predominan son: Java y C++.
- De los 11 servidores de mapas, 9 permiten el desarrollo externo, es decir el desarrollo de aplicaciones espaciales o interfaces para conectarse al servidor de mapas, solamente GeoMediaWeb y Mapnik no se encontró información referente al desarrollo al externo.
- De los 11 servidores de mapas, 8 soportan más de 4 bases de datos distintas, 5 servidores de mapas son comerciales: GeomediaWeb, Mapinfo MapXtreme, Manifold, ArcIMS, ArcGIS Server y 3 de código abierto: Deegree, MapServer, Geoserver.
- Todos los servidores de mapas cuentan con estándares de la Open Geospatial Consortium (OGC). La OGC es un consorcio de 382 compañías, agencias gubernamentales y universidades se encarga del desarrollo de estándares para la interoperabilidad del software enfocado a la información geográfica [OGC, 2009].
- De los 11 servidores de mapas 10 contaban con el soporte de diversos formatos vectoriales y raster, algunos de los servidores de mapas en sus páginas web hacían referencia a que eran compatibles con GDAL y OGR, que son un conjunto de librerías que dan el soporte de diversos formatos raster y vectoriales respectivamente [GDAL, 2009]. En cambio otros solamente indicaban solo los formatos con los que eran compatibles.
- De los 11 servidores de mapas, solo Geomedia Web Map no cuenta con algún medio electrónico o comunidad de usuarios que permita para externar dudas en relación al desarrollo de aplicaciones espaciales.

Los servidores de mapas de código abierto en comparativa con los servidores de mapas comerciales representan buena elección para los usuarios, debido a que permiten el desarrollo tanto interno como externo, no implican un costo económico, la mayoría son multiplataforma y cuentan con estándares de la OGC, soporte de diversas bases de datos y formatos raster y vectoriales; además

existen comunidades o medios de contacto para cada uno de los servidores, donde un usuario puede retroalimentarse en cuanto su uso y características. Este tipo de servidores de mapas son *ad hoc* para usuarios exigentes que desean desarrollar aplicaciones espaciales según sus gustos o con interés en contribuir en el mejoramiento del servidor de mapas; la posible limitante que pueda tener un usuario, es el conocimiento o habilidades que tenga entorno al lenguaje de programación interno y externo del servidor de mapas.

En el caso de los servidores de mapas comerciales son una alternativa para las empresas u organizaciones que están dispuestas a pagar un costo económico por un servidor de mapas, pero con la condicional de que estos ofrezcan una solución rápida, sencilla de instalar y configurar, que no tengan defectos (*bugs*) y que las características que manejen sean sofisticadas o que ninguno de los otros servidores de mapas de código abierto o servicio les proporcionen.

Normalmente una organización o empresa tiende a adquirir un servidor de mapas comercial a consecuencia de que estos les brindan la seguridad y garantía de su funcionamiento, además este tipo de servidores de mapas son menos complicados de instalar y configurar, no es necesario un conocimiento avanzado para realizar estas actividades. En contraparte los servidores de mapas de código abierto requieren que el usuario tenga conocimientos avanzados si éste se instala bajo un sistema operativo Linux debido a la compilación de librerías, configuración e instalación del servidor de mapas, en el caso de que la instalación de los servidores de mapas de código abierto se realice en una plataforma Windows, no existe complicación, porque hay soluciones integrales o paqueterías que contienen las librerías compiladas, el servidor web y el servidor de mapas.

Nombre	Tipo	Plataforma	Desarrollo Interno	Desarrollo externo (aplicaciones)	Soporte bases de datos	Estándares	Formatos soportados	Comunidad de usuarios/ Contacto	Enlace
ArcIMS 9.2	C	Windows, AIX, Solaris, HP-UX	NI	.NET, Javascript, Java	SQL Server, Informix, DB2 y Oracle	WMS, WFS	DWG, DXF, DGN, SHP, JPEG, GIF, entre otros	Si. Foros y correo electrónico	[ArcIMS,2009]
ArcGIS Server 9.2	C	Microsoft Windows Server, Linux, Solaris	NI	APIs para .NET, Java, Javascript y Flex	Oracle, Microsoft AccessSQL Server, Informix, DB2 y PostgreSQL	WMS, WCS, WFS, WFS-T, locator, globe,	ENVI, JFIF, ECW, JPEG, GIF, SHP, VPF, KML, GML, entre otros	Si. Foros y correo electrónico	[ArcGIS Server,2009]
Autodesk Mapguide Enterprise	C	Windows, Linux	NI	APIs para .NET, Javascript, PHP	Oracle, SQL server	NI	DWF	Si. Blogs, correo electrónico	[MapGuide,2009]
GeoServer 1.7.4	CA	Windows, Linux, Mac OS	Java	Javascript, PHP y XML	Oracle, ArcSDE, DB2, MySQL, PostgreSQL	WCS, WMS, WFS	GDAL y OGR	Si. Correo electrónico, chat y blog	[GeoServer,2009]
UMN MapServer 5.4.2	CA	Windows, Linux, Solaris, Mac OS	C	Mapscript: PHP, Java, Perl, C#, Ruby, Python.	ARCSD, Oracle, PostgreSQL, MySQL	WMS, WFS, WMC, WCS, SLD, Filter Encoding, GML, SOS, OM	GDAL y OGR	Si. Correo electrónico, chat	[MapServer,2009]
MapInfo MapXtreme 2005	C	Windows, Solaris, Linux, IBM AIX, HP UX	NI	Java, ASP.NET	DB2, SQL Server, Informix, Oracle	WMS, WFS, ML	TAB, JPEG, TIFF, GeoTIFF, SHP, Northwood, GML	Si. Foro	[MapXtreme,2009]
MapGuide OS 2.0.2	CA	Windows y Linux	C++	PHP, ASP.NET, Java/JSP y Javascript	MySQL, ArcSDE, ODBC	WMS, WFS	GDAL, DWF, SHP, SDF	Si. Correo electrónico	[MapGuideOS,2009]
GeomediaWeb	C	Windows	NI	NI	MGE, MGE Segment Manager, Oracle, Access, SQL Server	WMS, WFS, SOAP/ WSDL, OPENLS,	SVG, JPEG, PNG, ActiveCGM Vector Format, GML	No.	[GeomediaWeb,2009]
Manifold 8	C	Windows	NI	.NET: C#, VB.NET, VBScript, JScript, Perl, Python	Oracle, DB2, PostgreSQL, SQL Server, MySQL	WMS, WFS-T	SHP, MID/MIF, TIGERline, TAB, NTF, DGN, BMP, GIF, JPEG, entre otros.	Si. Foro	[Manifold,2009]
Mapnik 0.6.0	CA	Windows, Linux, Mac OS	C++, Python	NI	PostgreSQL	WMS	GDAL y OGR	Si. Correo electrónico y chat	[Mapnik,2009]
Deegree 2.2	CA	Windows, Linux	Java	Java	PostgreSQL, ArcSDE, Oracle, MySQL, ODBC, JDBC, MapInfo	WMS, WFS, WCS, CSW, SOS, WTS/WPVS, WSS	PNG, GIF, JPEG, GeoTiff, ECW	Si. Correo electrónico	[Deegree,2009]

C=Comercial, CA= Código abierto, NI= No hay información

Tabla 2.1. Lista de características representativas de servidores de mapas de código abierto y comerciales

2.4 Tipos de usuarios en servidores de mapas

Según Tyler Mitchell existen dos tipos de personas que utilizan los mapas en la web: 1. Los proveedores de servicios y 2.- los usuarios finales. [Mitchell, 2005]. En primera instancia los proveedores de servicios pueden estar integrados por un conjunto de personas por ejemplo: programador y diseñador de aplicaciones, administrador de servidor y personas especializadas en: la recolección, análisis y digitalización de la información, entre otros. Todos ellos con el único objetivo de brindar un servicio satisfactorio a los usuarios finales; de forma general los proveedores de servicios normalmente son responsables de mantener la disponibilidad de las fuentes de datos y del sitio web, de brindar una interfaz gráfica amigable y datos espaciales fiables, entre otros.

Por otro lado los usuarios finales son los consumidores de los servicios proporcionados por el proveedor, hacen uso de las funcionalidades que pueda proporcionar una aplicación espacial con la finalidad de resolver sus necesidades geográficas.

2.5 Servidores de mapas de servicios gratuitos

Los servidores de mapas: Google Maps, Yahoo Maps, Live Search Maps , Map24 y Mapquest; son conocidos como servicios gratuitos, esto se debe a que proporcionan a los usuarios: mapas, imágenes satelitales, videos, información de tráfico vehicular, información climática y funciones de localización de sitios (ciudades, países) de manera gratuita. Sin embargo el hecho de definirlos como gratuitos, no implica que un usuario quede exento de registrarse bajo ciertas condiciones de uso. Tal como lo expresa en su página web GOOGLE acerca del servicio gratuito google maps:

1. “Al utilizar Google Maps o cualquier dato o información al que se acceda a través de dicha aplicación, se compromete a respetar las Condiciones de servicio de Google, así como los términos y condiciones estipulados en el presente documento. En el caso de los usuarios particulares, Google Maps, incluidos los resultados de búsqueda locales, los mapas y las imágenes fotográficas, se ofrece únicamente para su uso personal y no comercial. En el caso de los usuarios de empresas, Google Maps se ofrece únicamente para su uso interno y no podrá redistribuirse con fines comerciales, salvo si se accede a los datos de los mapas y se muestran estos a través de la API de Google Maps, de acuerdo con lo especificado en los términos y condiciones de dicho servicio.” [GOOGLE,2009].
2. “Un usuario no podrá copiar, modificar, crear trabajos derivados, realizar trabajos de ingeniería inversa, desensamblaje o tratar de descubrir el código fuente del Software o de cualquier parte del mismo (y no permitirá que otros realicen tales actividades) salvo por lo expresamente permitido o exigido por Ley, o salvo que Google le haya autorizado expresamente y por escrito a llevar a cabo tales actuaciones.” [GOOGLE,2009]
3. “Google le otorga una licencia personal, mundial, exenta de regalías, intransferible y no exclusiva para utilizar el software que Google le proporciona como parte de los Servicios que se le ofrecen.” [GOOGLE,2009]

A pesar de tener ciertas restricciones de uso los servicios gratuitos, estos disponen de APIs (*Interfaz de Programación de Aplicaciones*), Ray Horak define una API como una aplicación que contiene un conjunto de rutinas las cuales permiten que otra aplicación interactúe con la aplicación que contiene las rutinas [Horak,2008].

Las compañías proveedoras como GOOGLE y YAHOO delimitan las funcionalidades y lenguajes de programación que se pueden utilizar sobre los servicios gratuitos que proveen mediante el uso de APIs. En el caso de las compañías Google y Yahoo permiten a un usuario insertar en su sitio web los servicios gratuitos (Google Maps, Yahoo Maps), manipular mapas, crear interfaces gráficas y añadir contenidos con la ayuda de la API propietaria de cada compañía.

En relación a los lenguajes de programación soportados por las APIs son variados para cada una de las compañías proveedoras de servicios gratuitos (Ver Tabla 2.2). De los 5 servidores de mapas de servicios gratuitos mostrados en la tabla anterior, podemos destacar lo siguiente:

- A pesar de MapQuest y Map24 no son las principales proveedoras de servicios gratuitos, si poseen una mayor cantidad de APIs soportadas a comparación de Yahoo Maps, Google Maps y Live Search Maps; lo que beneficia ampliamente a los usuarios, debido a que pueden elegir el lenguaje de programación que mejor se adapte a sus necesidades o limitaciones en cuanto al manejo de algún lenguaje.
- Live Search Map de Microsoft solamente permite el uso de una API Javascript
- Google Maps y Yahoo Maps las dos grandes empresas proveedoras de servicios gratuitos solamente permiten la interacción por medio de dos APIs, JavaScript y ActionScript en el caso de Google Maps y para Yahoo Maps Ajax y ActionScript.

Los servidores de mapas de servicios gratuitos son frecuentemente utilizados cuando el usuario u organización no cuenta con una infraestructura o recursos económicos que permitan construir un entorno de trabajo para el funcionamiento de un servidor de mapas y sus fuentes de datos.

Las proveedoras de servicios GOOGLE, YAHOO, MICROSOFT, AOL y NAVTEQ permiten a los usuarios utilizar los servidores de mapas de servicios gratuitos de forma sencilla y amigable; proporcionado una solución rápida a la necesidad de un usuario como: la ubicación de una oficina, centros recreativos de una ciudad, población, entre otros; sin embargo es importante tomar en cuenta las limitaciones que cada una de las proveedoras de servicios designan a los usuarios: ya que de eso dependerá los alcances de la aplicación espacial que pueda generar un usuario.

Compañía	Servicio	Lenguajes	Enlace
GOOGLE	Google Maps	JavaScript y ActionScript	[API GoogleMaps,2009]
YAHOO	Yahoo Maps	Ajax, ActionScript.	[API YahooMaps,2009]
AOL	MapQuest	C++,Action Script, Java, JavaScript y .NET	[API MapQuest,2009]
Microsoft	Live Search Maps	JavaScript	[API LiveSearchMaps,2009]
NAVTEQ	Map24	Ajax,Java,C# .NET, VB .NET, PHP 5, c++	[Map24,2009]

Tabla 2.2. Lenguajes de programación soportados por las APIs de las compañías proveedoras de servicios gratuitos

2.6 Diferencia de servidores de mapas de código abierto, comerciales y de servicios gratuitos

Paul Ramsey plantea en su artículo *Mashing up the Enterprise* las diferencias que existen entre los servidores de mapas de servicio gratuito y de código abierto, haciendo énfasis en la persona encargada de desarrollar aplicaciones espaciales.

1. La persona encargada de desarrollar aplicaciones espaciales a partir de servicios gratuitos como Google Maps, no necesita tener un alto nivel técnico para comprender su funcionamiento, solamente requiere conocer los lenguajes HTML y JavaScript o ActionScript [Ramsey, 2006].
2. La persona encargada de desarrollar aplicaciones espaciales a partir de código abierto necesita un amplio conocimiento en bases de datos, UNIX, proyecciones de mapas, formatos de datos espaciales, compilación de software, entre otros [Ramsey, 2006].

Paul Ramsey no menciona a los servidores de mapas comerciales, sin embargo podrían clasificarse en un nivel intermedio entre los servicios gratuitos y de código abierto para el desarrollo de aplicaciones espaciales. La mayoría de los servidores de mapas comerciales son para sistemas operativos Windows (Ver Tabla 2.1), están exentos de la compilación de software o librerías y en consecuencia los servidores de mapas comerciales normalmente son fáciles de instalar. Sin embargo si es necesario tener conocimientos de bases de datos, proyecciones de mapas, formatos de datos espaciales y lenguajes de programación para crear o modificar interfaces gráficas de las aplicaciones espaciales.

Por otro lado una forma más concreta de diferenciar los distintos servidores de mapas es por el tipo de desarrollo que permiten:

- **Desarrollo Interno:** Capacidad de crear, modificar o mejorar el funcionamiento del servidor de mapas a partir de su código fuente.
- **Desarrollo externo:** Capacidad de crear, modificar o mejorar el lado visual, es decir la aplicación cliente.

Con base en la anterior diferenciación, los servidores de mapas de código abierto se sitúan dentro del desarrollo interno como externo, en cambio los servidores de mapas de comerciales y de servicios gratuitos normalmente están restringidos al desarrollo externo. Por lo tanto es importante conocer las fortalezas y debilidades de un servidor de mapas al momento de emprender un proyecto que involucre el uso de este tipo de tecnologías, ya que algunas veces por desconocimiento se puede elegir un servidor de mapas con funcionalidades limitadas o que no satisfagan las necesidades del usuario.

2.7. Tipos de aplicaciones espaciales en entornos web para servidores de mapas

Las aplicaciones espaciales son conocidas también como aplicaciones cartográficas o aplicaciones geográficas. Para propósito de este proyecto de investigación se manejará el concepto de aplicaciones espaciales. A consecuencia de que no se encontró una definición formal de una aplicación espacial, se definirá como un programa cliente que proporciona al usuario una interfaz gráfica a través de un navegador web, la cual permite el proceso de interacción con los datos espaciales con la ayuda de un servidor de mapas.

El objetivo de este apartado es mostrar la diversidad de aplicaciones espaciales que pueden ser utilizadas en los servidores de mapas.

2.7.1 Aplicación espacial enfocada a delitos

La policía de Clearwater Florida en Estados Unidos tiene una aplicación (Ver Figura 2.3) para llevar un control de los delitos denunciados vía telefónica, los cuales normalmente son por violaciones, prostitución, robo de vehículos, drogas, asaltos a mano armada, homicidios, piratería, entre otros., además esta aplicación sirve para determinar las zonas más conflictivas de Clearwater, aplicar estrategias para reducir los delitos y prevenir a la comunidad acerca de las zonas con incidencia delictiva.

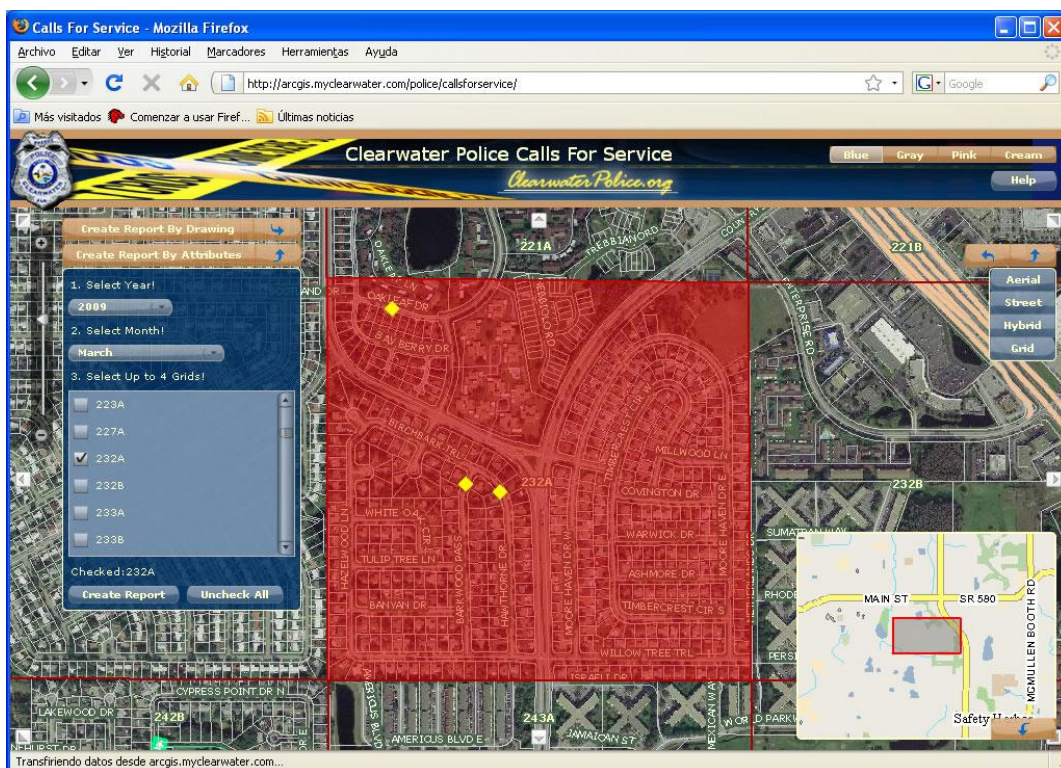


Figura 2.3. Aplicación espacial en un entorno web enfocada a delitos reportados en Clearwater Florida, Estados Unidos.

Fuente: <http://www.myclearwater.com/info/about/geography/index.asp>

2.7.2 Aplicación espacial para el control del tráfico vehicular

El consorcio Houston TranStar de Houston Texas Estados Unidos, tiene esta aplicación (ver Figura 2.4) para mostrar: el tránsito vehicular, zonas con incidentes, carriles cerrados, cámaras en vivo en carreteras, zonas recreativas, centrales de camiones y la promedio velocidad en que circulan los vehículos.

Según la información obtenida en el sitio web del Houston TranStar los beneficios de que ha tenido esta aplicación en Houston Texas son: la reducción en la duración de viajes, consumo de combustible y emisiones de bióxido de carbono, hidrocarburos y óxidos de nitrógeno. Además esta aplicación ha servido para realizar gestiones de emergencias y advertencias para disminuir el número de muertes, lesiones, daños en propiedad ocasionados por inundaciones y otros problemas relacionados con el clima o eventos humanos.

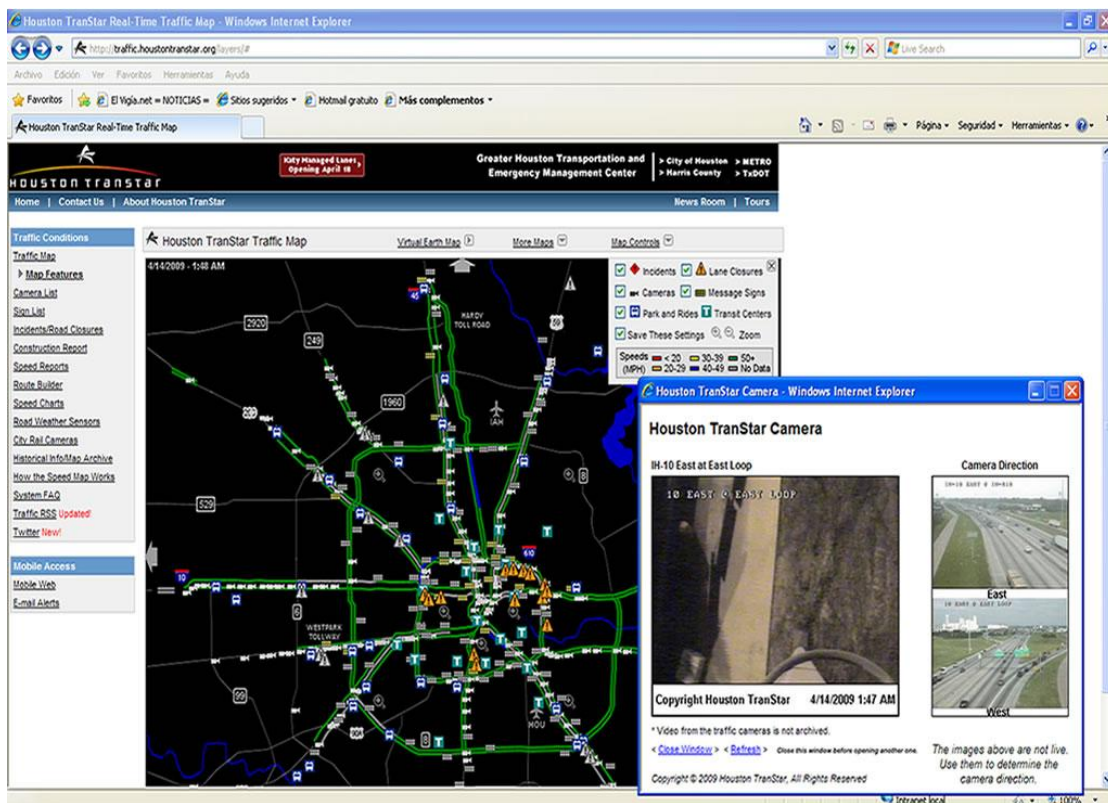


Figura 2.4. Aplicación espacial en un entorno web enfocada a tráfico vehicular de Houston Texas, Estados Unidos.

Fuente: <http://traffic.houstontranstar.org/layers/>

2.7.3 Aplicación espacial para la disponibilidad de propiedades

Savannah Economic Development Authority (SEDA) es una organización de Atlanta Estados Unidos que tiene el objetivo de mejorar el nivel de vida de las personas de Savannah y el condado de Chatam por medio de la atracción de inversión, creación de trabajos y el apoyo de las empresas establecidas [seda.org,2009]. La organización SEDA tienen esta aplicación (Ver Figura 2.5) para proporcionar información de interés para inversionistas como: las propiedades disponibles para establecerse en Savannah y el condado de Chatam, distancias con el aeropuerto más cercano o puerto embarcación, medidas de los terrenos, calles principales que circundan a la propiedad, entre otras.

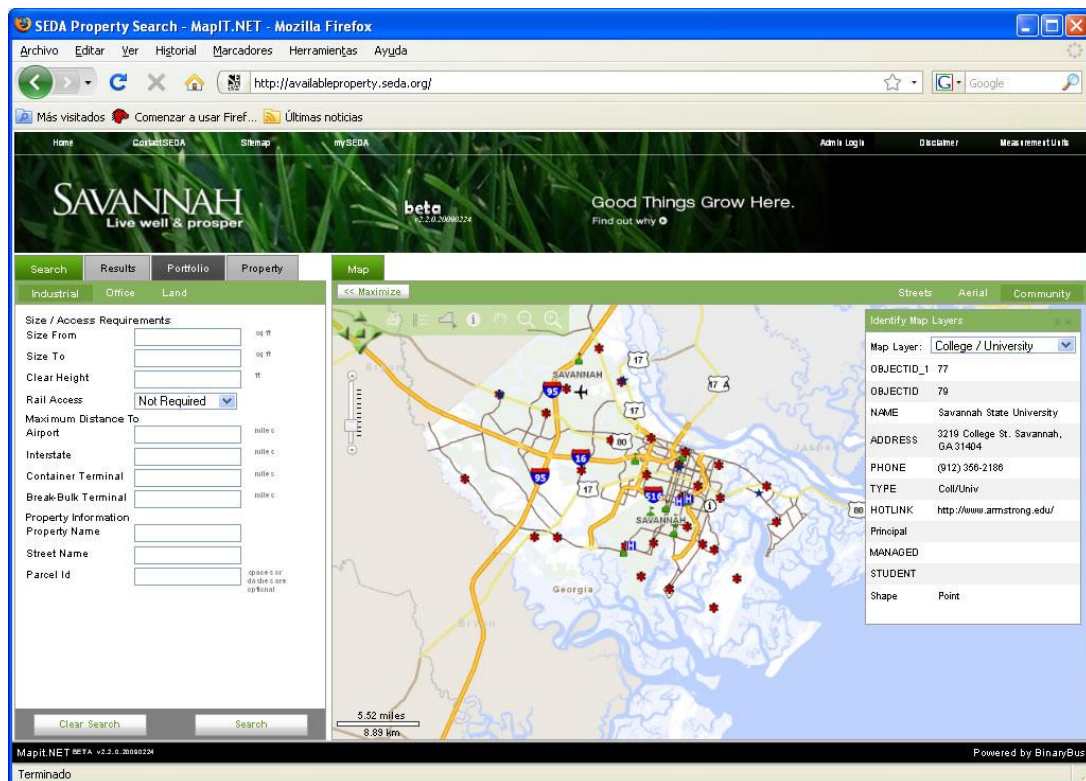


Figura 2.5. Aplicación espacial en un entorno web enfocada a disponibilidad de propiedades en Savannah y el condado de Chatam en Atlanta, Estados Unidos
Fuente: <http://availableproperty.seda.org/>

2.7.4 Aplicación espacial enfocada a incendios

La Multi-Agencia de Coordinación Geoespacial (GEOMAC) integrada por técnicos y expertos del departamento interior de gestión de incendios de Estados Unidos, tuvo la necesidad de implementar una aplicación enfocada a incendios (Ver Figura 2.6); ya que en el año 2000 ocurrieron múltiples incendios en Estados Unidos que ocasionaron la pérdida de cientos de estructuras y recursos naturales, esto en consecuencia de que no contaban con centros de coordinación para el combate de incendios, por lo cual a partir de esta aplicación les permitió distribuir y construir los centros de coordinación de manera más efectiva.

Como comenta GEOMAC en su sitio web esta aplicación proporciona de forma precisa y completa la proximidad de incendios en zonas pobladas de Estados Unidos y Alaska; tiene un rol muy importante la actualización de los datos espaciales en el servidor de mapas por lo que utilizan: GPS, imágenes infrarrojas y satelitales; ayudando a los bomberos a localizar en tiempo real los incendios.

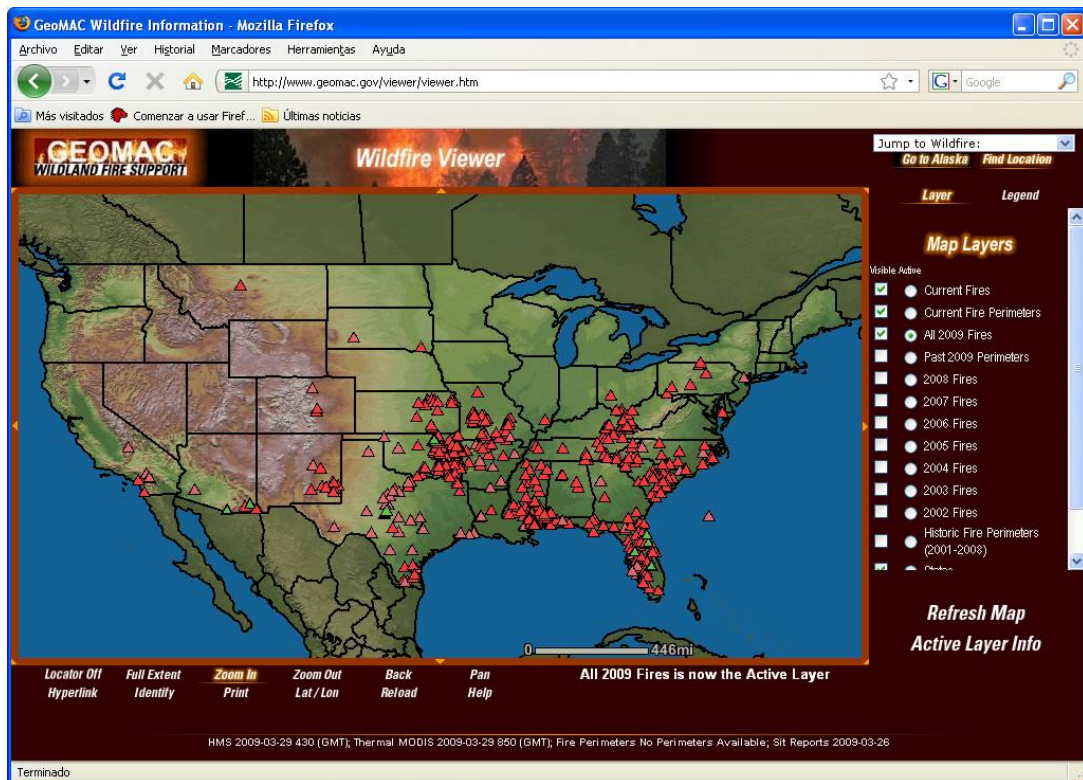


Figura 2.6. Aplicación espacial en un entorno web enfocada a la gestión de incendios en Estados Unidos

Fuente: <http://www.geomac.gov/viewer/viewer.htm>

2.7.5 Aplicación espacial para ecosistemas

El consejo del Golfo de Maine en ambientes marinos de Estados Unidos tiene esta la aplicación (Ver Figura 2.7) para mantener y mejorar la calidad del ambiente del Golfo de Maine. Como comentan en su sitio web esta aplicación sirve para monitorear los lugares: contaminados, de pesca, acuicultura y los cambios en la temperatura, entre otros.

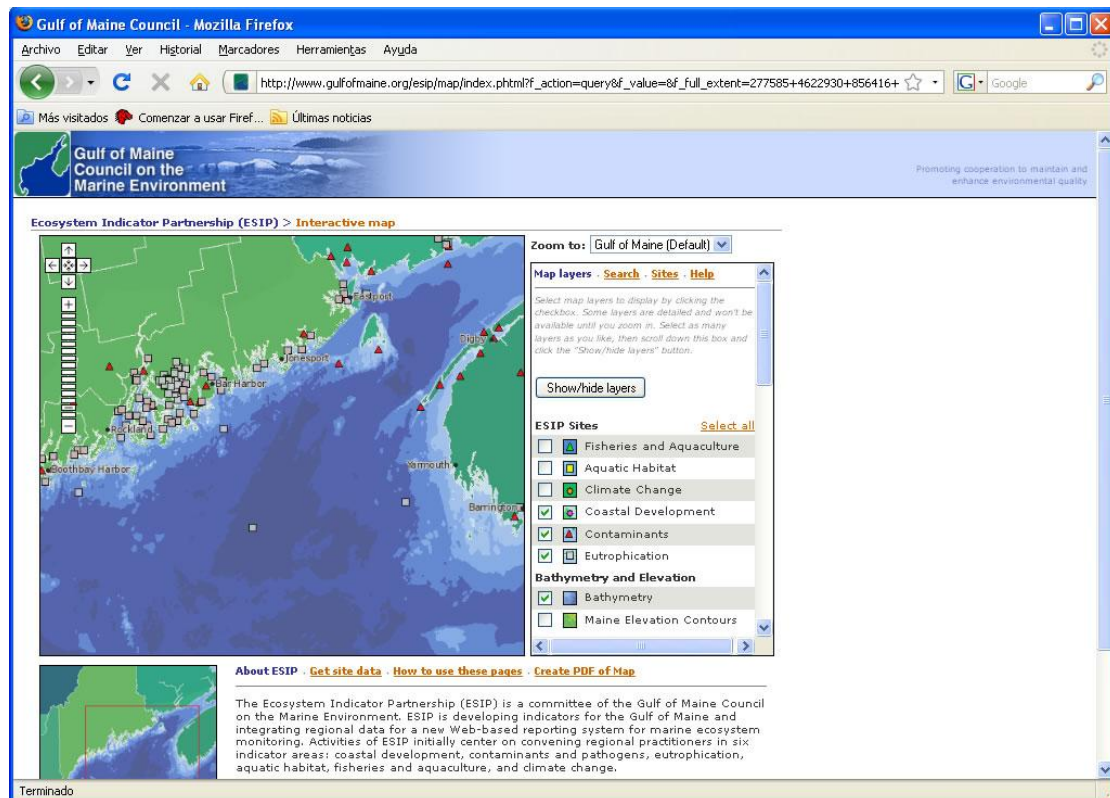


Figura 2.7. Aplicación espacial en un entorno web enfocada al ecosistema del Golfo de Maine, Estados Unidos.

Fuente: <http://www.gulfofmaine.org/esip/>

2.7.6 Aplicación espacial enfocada a finanzas

Financemaps tiene en su sitio web una aplicación espacial que permite: visualizar el comportamiento de la compra-venta de divisas en todo el mundo, el porcentaje de pérdidas y ganancias económicas de los países, la economía más fuerte y débil en las últimas semanas, meses y años, entre otros.

La forma de representar los cambios financieros en la aplicación espacial es por medio de un rango de colores, estos tienen un valor porcentual que permite determinar el comportamiento de las divisas. Por ejemplo si existe una fuerte devaluación de la divisa de un país, se visualizara de color rojo automáticamente (Ver Figura 2.8).

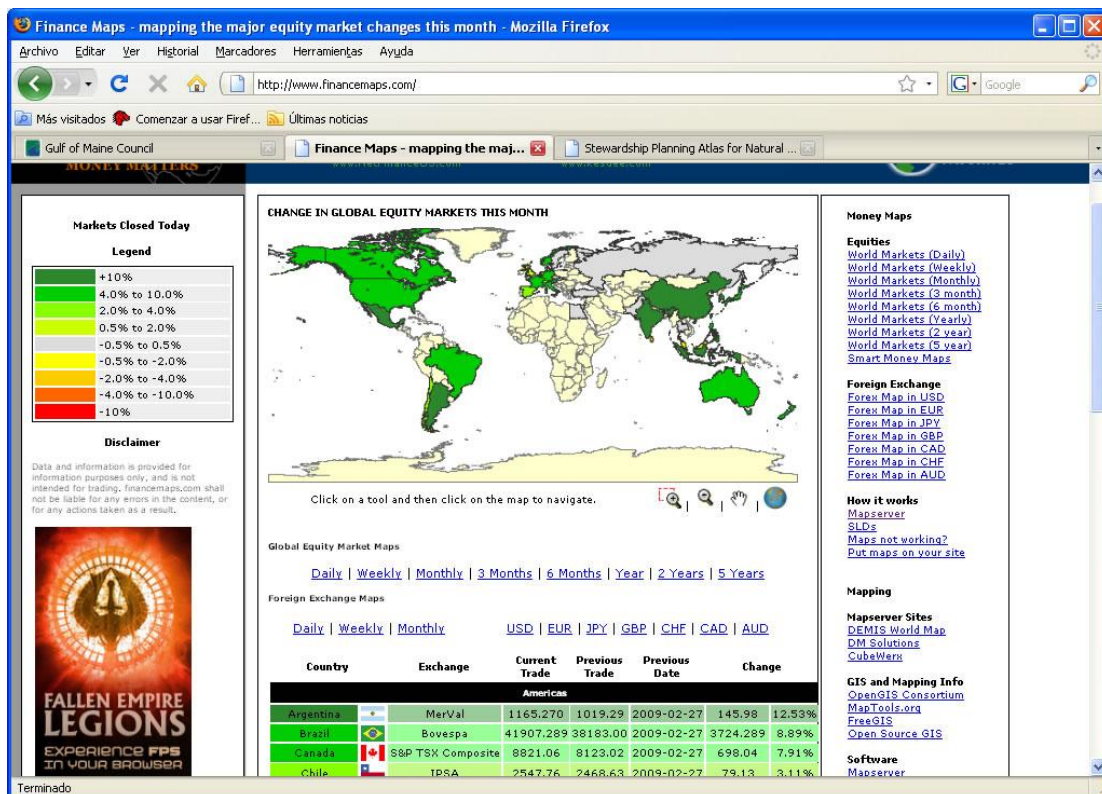


Figura 2.8. Aplicación espacial en un entorno web enfocada a finanzas globales.
Fuente: <http://www.financemaps.com/>

2.7.7 Aplicación espacial para el control de Epidemias

El National Reference Centre For Meningococci(NRZM) del Instituto para la Higiene y Microbiología de Wurzburg, Alemania, utiliza la aplicación EpiScanGIS para detectar los brotes de la enfermedad *Meningococcal* con finalidad de minimizar el número de infecciones en la población. Como comentan en el sitio web de NRZM la detección rápida de la epidemia es con la ayuda del software SaTScan el cual escanea semanalmente una base de datos que proporciona la información para la aplicación espacial EpiScanGIS (Ver Figura 2.9). Esta aplicación permite al NRZM visualizar las zonas impactadas por la epidemia, así como también el número de casos registrados, decesos, rango de edades de personas enfermas, entre otros.

John Gilroy define a la enfermedad Meningococcal como una bacteria infecto-contagiosa que causa problemas de inflamación en estructuras que cubren el encéfalo y medula espinal, esta bacteria afecta el extracto respiratorio, causa trombosis profunda e infarto cerebral, algunas puede provocar hasta la muerte [Gilroy,2001].

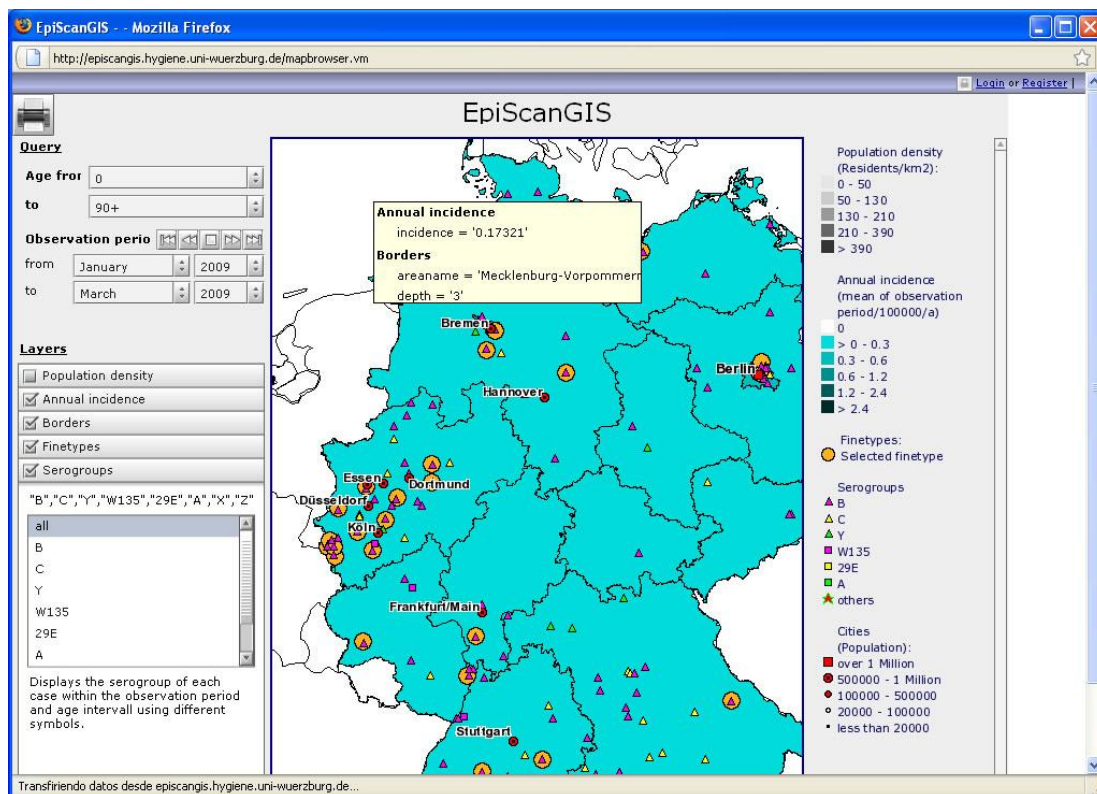


Figura 2.9. Aplicación espacial en un entorno web enfocada a la detección de epidemias en Wurzburg Alemania

Fuente: <http://episcangis.hygien.uni-wuerzburg.de/mapbrowser.vm>

2.7.8 Aplicación espacial para la agricultura

El Ministerio de Medio Ambiente Rural y Marino (MARM) del gobierno España utiliza esta aplicación (Ver Figura 2.10) para mostrar las zonas de cultivo, industrias agroalimentarias, agroturismo, regadíos, estaciones meteorológicas, entre otras; con propósitos de difusión para personas afines al sector agrario y además para tener un control de la distribución de las actividades económicas rurales que se realizan en España.

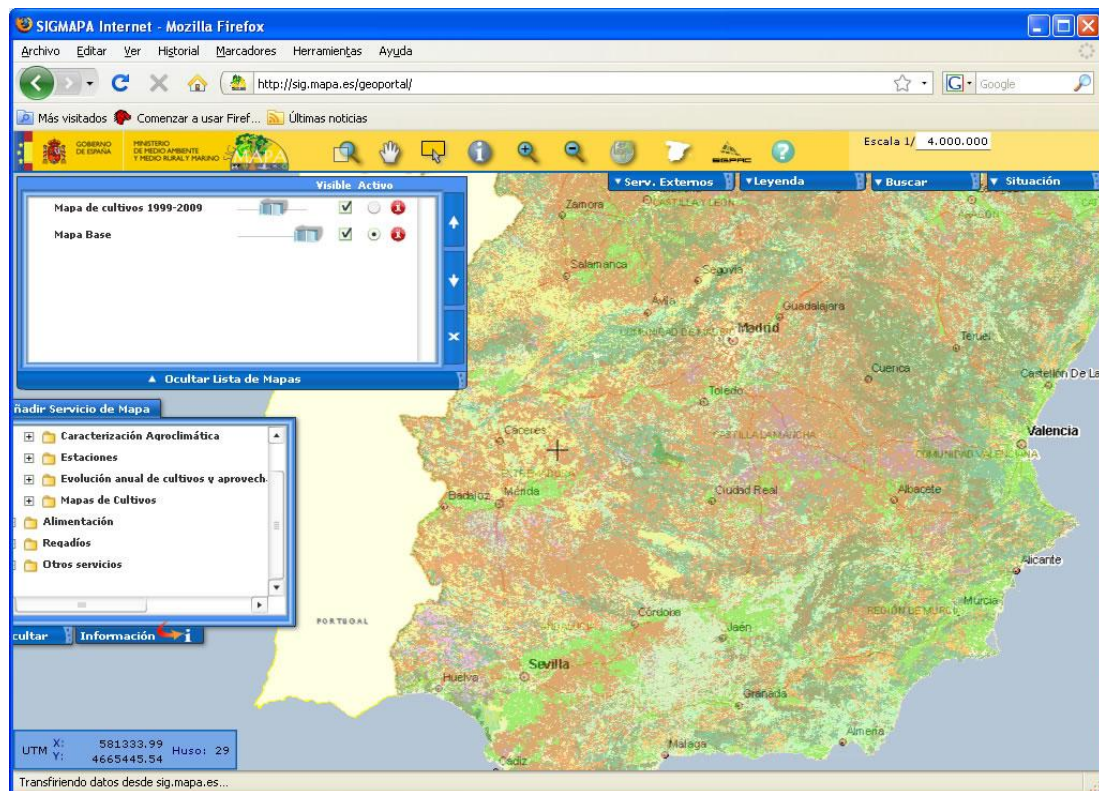


Figura 2.10. Aplicación espacial en un entorno web enfocada a la agricultura en España.
Fuente: <http://sig.mapa.es/geoportal/>

2.7.9 Aplicación espacial para áreas de recreación

El Departamento de Recursos Naturales de Minnesota, Estados Unidos, utiliza esta aplicación (Ver Figura 2.11) para promover las áreas recreativas estatales y federales de Minnesota como: lagos, ríos, parques, bosques, entre otros; para atraer más turismo y ganancias al estado.

La información que muestra esta aplicación es: la ubicación geográfica, clima, áreas de picnic, áreas de acampar, oficinas de información y seguridad, aserraderos, las actividades que el turista puede realizar en ciertas zonas, entre otros.

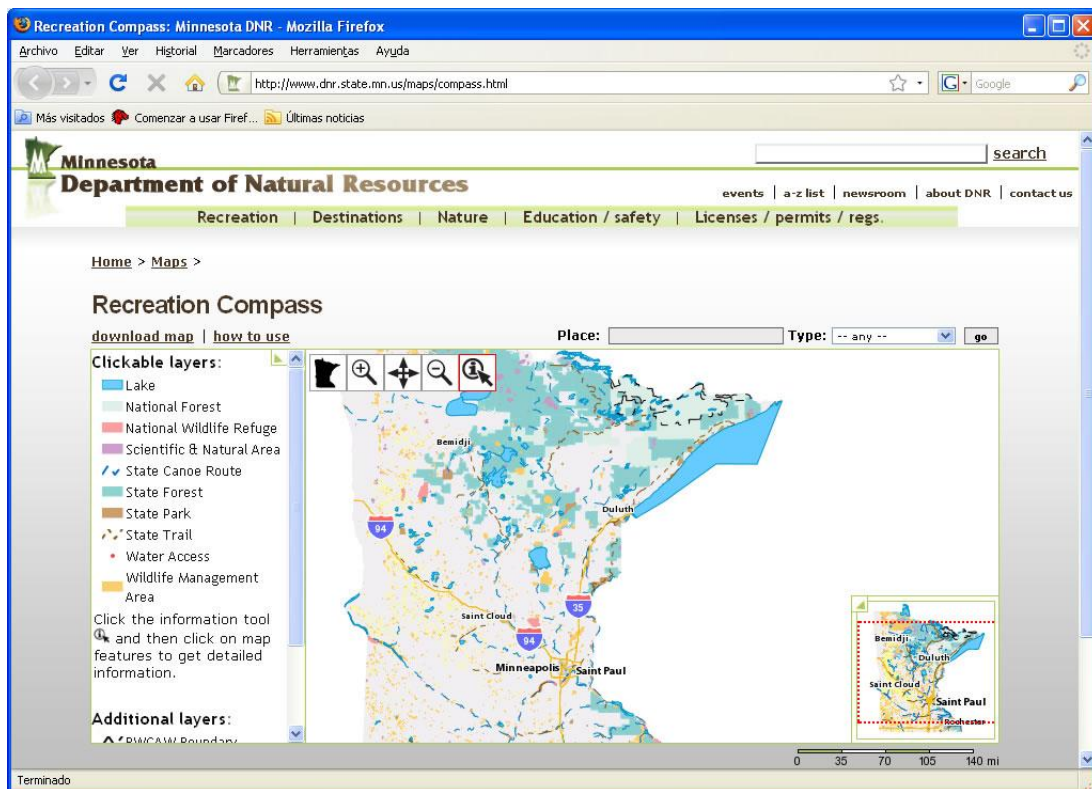


Figura 2.11. Aplicación espacial en un entorno web enfocada a áreas recreativas de Minnesota, Estados Unidos.

Fuente: <http://www.dnr.state.mn.us/maps/compass.html>

2.7.10 Aplicación espacial de riesgos geológicos-Volcanes

El Centro Nacional de Prevención de Desastres (CENAPRED) de México, utiliza esta aplicación (ver Figura 2.12) para: implantar medidas de prevención de desastres, evaluación de pérdidas materiales y humanas, atender las necesidades de una emergencia, mejorar la calidad en la contratación de seguros de la infraestructura pública, mostrar rutas de evacuación, posibles zonas impactadas por la caída de cenizas, material volcánico y flujo de lava del Volcán Popocatepetl.

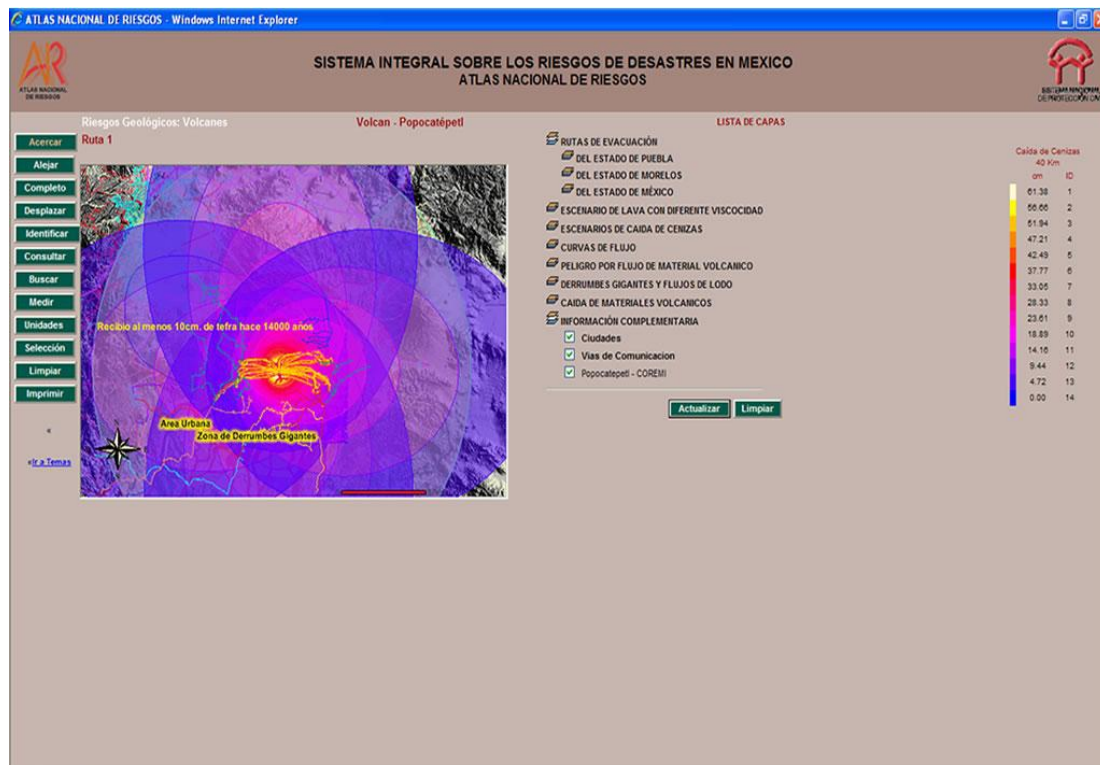


Figura 2.12. Aplicación espacial en un entorno web enfocada al riesgo geológico del volcán Popocatepetl en México

Fuente: <http://atl.cenapred.unam.mx/metadataexplorer/index.html>

2.7.11 Aplicación espacial de riesgos geológicos-sísmicos

El Centro Nacional de Prevención de Desastres (CENAPRED) utiliza esta aplicación (Ver figura 2.13) con los mismos objetivos que la aplicación de riesgos geológicos los cuales son: implantar medidas de prevención de desastres, evaluación de pérdidas materiales y humanas, atender las necesidades de una emergencia y mejorar la calidad en la contratación de seguros de la infraestructura pública.

En este caso la aplicación está enfocada al riesgo sísmico en el Distrito Federal, la información que muestra es acerca de las diversas edificaciones, clasificándolas por el tipo construcción y tipo de material, el número de niveles de la estructura con la finalidad de determinar la vulnerabilidad de las estructuras y las zonas con mayor impacto en caso de un sismo o terremoto.

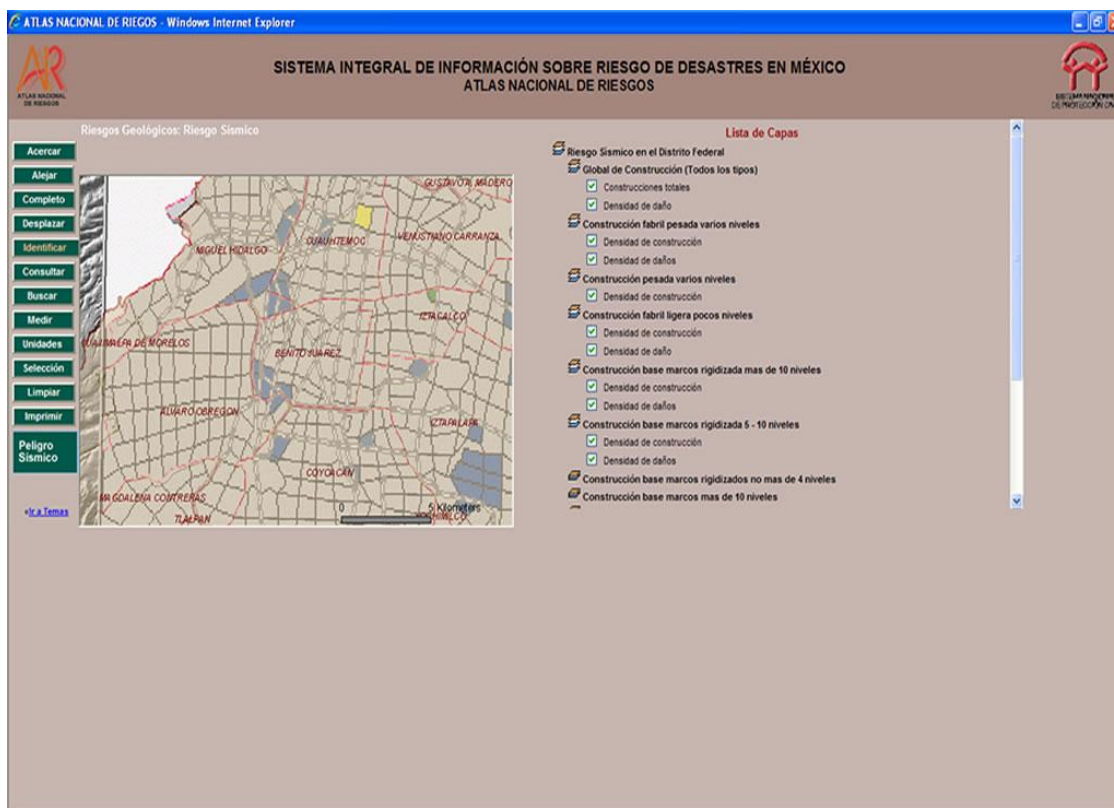


Figura 2.13. Aplicación espacial en un entorno web enfocada a riesgo geológico en sismos en el Distrito Federal México

Fuente: <http://atl.cenapred.unam.mx/website/RiesgosGeologicos/RiesgoSismicoDF/viewer.htm>

2.7.12 Investigación y Planeación

El Instituto de Municipal de Investigación y Planeación de Ensenada Baja California México, utiliza esta aplicación (Ver Figura 2.14) para la planificación y ordenamiento territorial de Ensenada, muestra vialidades, zonas agrícolas, poblados, hidrología, identifica las zonas de riesgo por deslizamiento en la ciudad, permite realizar mediciones de terrenos o distancia entre dos puntos de la ciudad, entre otras.

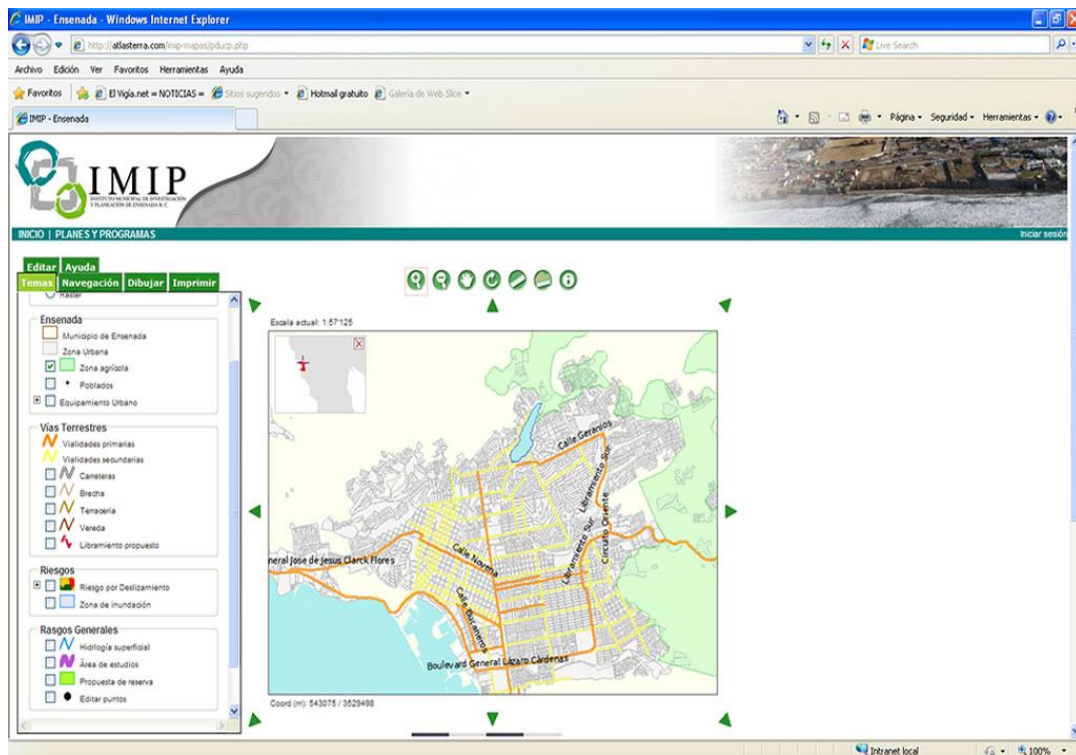


Figura 2.14. Aplicación espacial en un entorno web enfocada a la investigación y planeación urbana de Ensenada Baja California, México

Fuente: <http://atlasterra.com/imip-mapas/pduc.php>

2.8 Mecanismos que buscan garantizar las fuentes de datos espaciales

Un servidor de mapas obtiene datos espaciales de fuentes remotas y locales, almacenados en bases de datos y archivos: raster y vectoriales.

Fuente: La procedencia o el lugar donde están almacenados los datos espaciales.

Partiendo del punto anterior se realizó una investigación para identificar algoritmos para garantizar la disponibilidad de las fuentes de datos espaciales específicamente en los servidores de mapas, sin embargo, no se encontró ningún algoritmo. Por lo tanto se amplió el margen de investigación y se buscaron mecanismos externos a los servidores de mapas que buscan a garantizar las fuentes de datos espaciales. Los mecanismos encontrados se podrían denominar como genéricos, ya que no solamente son usados para los datos espaciales, sino para cualquier tipo de dato o información.

A continuación se presentan los mecanismos externos que buscan garantizar las fuentes de datos espaciales:

2.8.1 Redundancia de dispositivos

La redundancia de dispositivos se refiere a la duplicación de hardware en computadoras, suministro eléctrico, componentes de red y el balanceo de cargas [Linux-es, 2009]. A continuación se presenta una tabla que muestra duplicación de dispositivos (Ver Tabla 2.3).

Tipo	Dispositivos redundantes
Computadora	Fuentes de poder, discos duros, ventiladores, procesadores, tarjetas de red
Componentes de red	Router, Switch.
Suministro eléctrico	UPS, generadores eléctricos y líneas independientes de suministro
Balanceo de cargas	Clusters

Tabla 2.3. Clasificación de redundancia de dispositivos del lado del servidor físico.
Fuente: Información obtenida de Linux-es.org

La implementación de este tipo de tecnología en empresas o institutos implica un costo elevado. Sin embargo el grado de redundancia de un sistema, dependerá de su importancia y del dinero que una empresa o instituto esté dispuesto a perder cuando el sistema no está disponible por un fallo. [Linux-es, 2009]. Normalmente la redundancia de dispositivos se utiliza en servicios o sistemas críticos que necesitan estar disponibles de forma constante.

Sommerville define a *sistema crítico* como un sistema que puede provocar pérdidas económicas, daños físicos o amenazas de la vida humana en caso de la caída del sistema [Sommerville, 2002].

La redundancia de dispositivos garantiza que un servidor sea menos vulnerable a fallas relacionadas con hardware y que consecuencia los datos almacenados o los servicios en el servidor estén disponibles, siempre y cuando la falla no provenga del software.

A continuación se presenta una cotización (Ver Tabla 2.4) aproximada para la adquisición de dispositivos que permitan tener un sistema redundante, los dispositivos que se cotizaron fueron con base a la tabla *clasificación de redundancia de dispositivos del lado del servidor físico* (Ver Tabla 2.3). Los precios variarán dependiendo de la marca y características de los dispositivos o equipos, cabe señalar que la intención de esta cotización no es mostrar un costo total para un determinado esquema o arquitectura de una empresa o institución, sino simplemente mostrar los costos independientes de las tecnologías, para más información en relación a la descripción de los dispositivos acudir a la página web del fabricante.

Tipo	Dispositivos redundantes	Costo (Pesos)	Fuente
Computadora	Servidor IBM Bladecenter S 7 U Xeon DC 5120	\$89,295.46	[IntelCompras, 2009]
Componentes de red	Router D-Link DIR-855 Xtreme N Duo Media 5 puertos	\$4029.27	[dlink,2009]
	Switch D-Link DES-1008PA 8 puertos	\$1807.00	
Suministro eléctrico	UPS Sola Basic duración 30 minutos	\$4000.00	[CALCOM,2009]
	Generador eléctrico Mitsubishi 11Kw de Diesel	\$119,977.00	[gopower,2009]
Balanceo de cargas por software	Clusters –Linux Virtual Server	Sin costo	[linuxvirtualserver,2009]

Tabla 2.4. Cotización aproximada de dispositivos redundantes

Respecto al balanceo de cargas por software se utiliza bajo un esquema *clústeres* con la finalidad de mantener la disponibilidad de los servicios que se proporcionen en un servidor de aplicaciones, en caso de que falle un servidor de aplicaciones, el software que se encargará de seleccionar otro servidor de aplicaciones para que siga proporcionando los servicios que se ofrecían.

2.8.2 RespalDOS de datos

Normalmente una institución o empresa mantiene varios repositorios de datos como seguridad ante contingencias debido a: fallas en hardware o software, mantenimiento preventivo, robo de información o equipo, entre otras. En este sentido si existe alguna de estas contingencias la institución o empresa está preparada para utilizar uno de los repositorios de manera provisional para seguir proporcionado cualquier servicio que ésta preste.

El mantener varios respaldos de datos como seguridad ante contingencias ocasiona la redundancia de datos o información, sin embargo este mecanismo ayuda a garantizar su disponibilidad. Los respaldos de datos se realizan periódicamente, a criterio de la institución o empresa, esto se lleva a cabo manualmente o con la ayuda de software. Para el respaldo de datos de forma manual es una copia directa de un dispositivo de almacenamiento a otro (disco duro a disco duro) o dispositivo de almacenamiento a servicio FTP o viceversa, esta acción es en referencia copiar-pegar con la ayuda de teclado o ratón de una computadora. A continuación una tabla que muestra software para el respaldo de datos o información (Ver Tabla 2.5).

Tipo	Software	Fuente
Bases de datos	PHPMyAdmin-MySQL	[phpmyadmin,2009]
	PgMyAdmin- PostgreSQL	[phppgadmin,2009]
Multipropósito	Windows XP-Utilidad Copia de Seguridad	[microsoft,2009]
	Linux-AMANDA	[amanda,2009]
	Handy Backup	[handybackup2009]
	Norton Ghost	[symantec,2009]

Tabla 2.5 Software para el respaldo de datos

Aunque los ejemplos anteriores no son exclusivos del respaldo de datos espaciales, si representan una solución genérica para mantenerlos disponibles y en consecuencia servir como una fuente de datos provisional donde un servidor de mapas pueda obtener datos.

2.8.3 Datos espaciales distribuidos

Los datos espaciales se pueden mantener distribuidos de dos formas:

1. Sistemas de bases de datos espaciales distribuidas: remotas y locales (Ver Figura 2.15).



Figura 2.15. Esquema distribuido de bases de datos en la actualidad.

Para almacenar datos espaciales en una base de datos, es necesario que ésta cuente con soporte espacial para su funcionamiento. En algunas bases de datos esto se lleva a cabo a partir de la instalación librerías o módulos (Ver Tabla 2.6). Las bases de datos con soporte espacial son conocidas como bases de datos espaciales, Kennedy las define como una colección de datos espaciales y sus atributos, organizados de forma eficiente [Kennedy, 2000].

Base de datos	Modulo o librería	Fuente
PostgreSQL	PostGIS	[postgis,2009]
Mapinfo	Spatialware	[mapinfo,2009]
MySQL	MySQL Spatial extension	[dev.mysql,2009]
Oracle	Oracle Spatial	[oracle,2009]

Tabla 2.6 Soporte espacial para bases de datos

Como se puede observar en la figura anterior (Ver Figura 2.15), el usuario se conecta a un servidor de mapas a través de una aplicación espacial cliente, el usuario solicita datos espaciales y el servidor de mapas se encarga de obtenerlos de las bases de datos locales y de la base de datos remota.

2. Bajo un esquema de red de área local (Ver Figura 2.16)

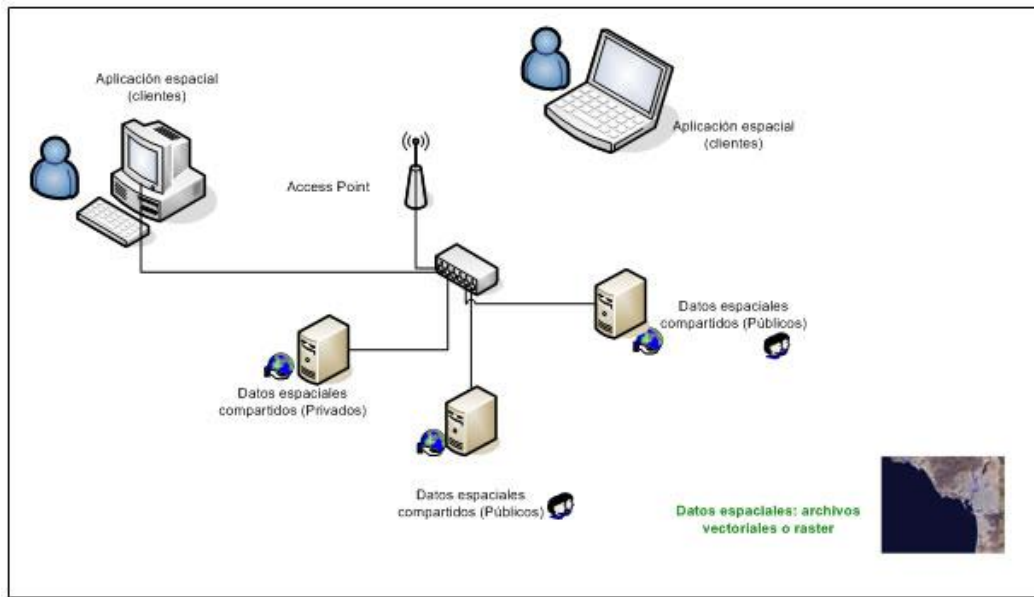


Figura 2.16. Datos espaciales compartidos bajo un esquema de área local

En la figura anterior (Ver Figura 2.16) los datos espaciales están compartidos y almacenados en cada computadora que compone la red, bajo este esquema los datos espaciales se pueden obtener directamente de los archivos raster o vectoriales, es decir los datos espaciales no están almacenados en una base de datos. Cabe señalar pueden incluirse bases de datos para almacenarse los datos espaciales, pero ya no se obtendrían directamente de los archivos raster y vectoriales.

Metodología

3

Se fundamenta la metodología de software para el desarrollo del proyecto y se explican de manera general cada una de las etapas que la constituyen.

3.1 Metodología de desarrollo de software

Para el cumplimiento de los objetivos del proyecto nos basamos en la metodología de desarrollo evolutivo exploratorio (Ver Figura 3.1) de acuerdo a Ian Sommerville permite desarrollar una implementación inicial, retroalimentar con el cliente y realizar mejoras a través de versiones hasta obtener la versión final, cumpliendo con las necesidades inmediatas del cliente [Sommerville, 2002]. Ian Somerville además comenta que el desarrollo evolutivo exploratorio empieza con las partes del sistema que se comprenden mejor y se van agregando nuevos atributos de acuerdo con las propuestas del cliente, lo que posibilita la entrega acelerada del sistema y el compromiso constante con cliente debido a la retroalimentación [Sommerville, 2002].

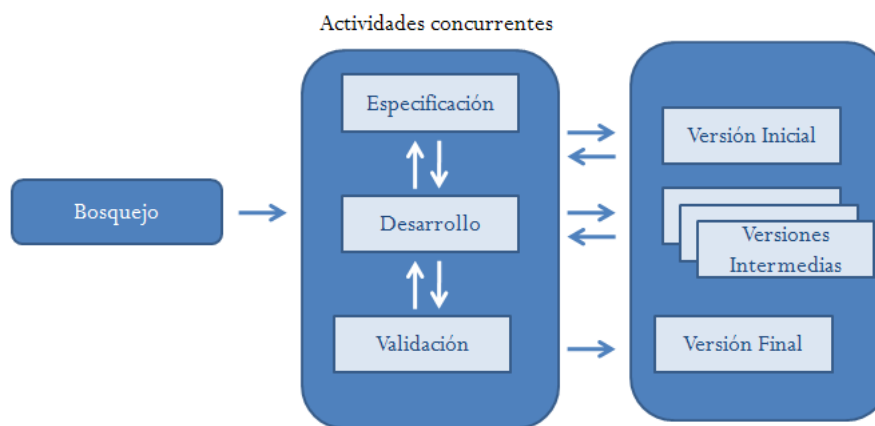


Figura 3.1. Metodología de desarrollo evolutivo
Fuente: Sommerville (2002)

Bosquejo: Consiste en indagar de manera general las necesidades del proyecto con base en los objetivos específicos planteados. Estas necesidades involucran la búsqueda de: la persona que fungirá como cliente, el enfoque y el posible alcance de la aplicación espacial a desarrollar, la adquisición de información para la elaboración de los datos espaciales, cantidad de computadoras requeridas.

Especificación: Con base en la etapa de bosquejo se plantean las necesidades específicas del cliente en relación a la aplicación espacial, se fundamenta la selección del servidor de mapas y se analiza su funcionamiento, se describen los requerimientos del algoritmo a desarrollar y arquitectura de computadoras.

Desarrollo: Se subdivide en tres fases: 1.- Diseño y desarrollo de aplicación espacial, 2.- diseño de esquema de selección de datos espaciales 3.- desarrollo del algoritmo de selección de datos espaciales para un servidor de mapas.

Validación: Esta etapa está enfocada a evaluar el funcionamiento de la aplicación espacial, el diseño del esquema de selección de datos espaciales y el desarrollo del algoritmo por medio de las pruebas aceptación y alfa.

Versiones: La metodología de desarrollo evolutivo exploratorio contempla el desarrollo de prototipos:

1. Versión inicial: el prototipo debe incluir los requerimientos básicos que permiten su funcionamiento, estos requerimientos son los de mayor prioridad para el usuario y que al mismo tiempo domina el programador.
2. Versión intermedia: El prototipo debe considerar los requerimientos con menor prioridad para el usuario y que al mismo tiempo son complejos de desarrollar para el programador, inclusive algunas veces nuevos requerimientos y modificaciones de requerimientos existente.
3. Versión final: es el sistema que cumple con los requerimientos establecidos.

3.2 Creación de prototipos

Para desarrollar un prototipo de manera eficaz se deben seguir ciertos pasos (ver figura 3.2)

1. Desarrollar especificación abstracta: No se considera necesario detallar ampliamente el funcionamiento que debe tener un sistema, si no solamente una descripción general que permita al programador desarrollar un prototipo funcional.
2. Construir prototipos: a partir de las especificaciones del usuario, el programador desarrolla un prototipo.
3. Utilizar sistema prototipo: se realizan pruebas de funcionamiento del prototipo y se compara con las especificaciones del usuario.
4. ¿Sistema apto?: Si el prototipo es funcional y cumple con las especificaciones del usuario puede considerarse como un sistema apto. En caso contrario se construye un nuevo prototipo que si cumpla con las especificaciones, se crean nuevas especificaciones o se modifican las existentes.
5. Entregar sistema: Si el prototipo es considerado apto y cumple con todas las especificaciones realizadas durante las iteraciones el sistema es candidato a una versión final.

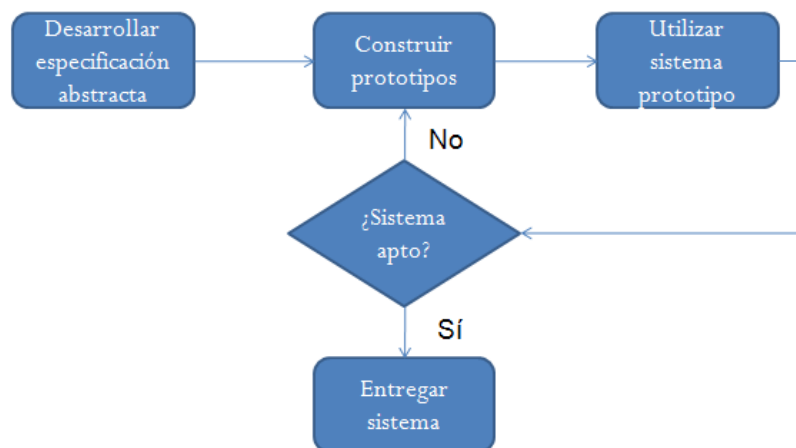


Figura 3.2. Esquema de construcción de prototipos
Adaptado de Sommerville (2002)

Bosquejo

4

Contiene las descripciones generales de los actores involucrados en el desarrollo del proyecto: administrador, especialistas en la gestión de riesgos; así como también las necesidades del proyecto y sus posibles alcances. Este capítulo sirve para tener una idea del rumbo que tomará el proyecto y para plantear las primeras reglas de responsabilidades compartidas, esto se lleva a cabo con la elaboración de una descripción general de las necesidades del proyecto y de las personas involucradas. El contar con una descripción general facilita la realización de las especificaciones formales del proyecto, las cuales se van mejorando conforme avanza su desarrollo.

4.1 Descripción general del administrador

- **Recursos humanos:** Con la finalidad de dar sustento al lado aplicativo del proyecto fue indispensable buscar personas especializadas en alguna área, que cumplieran cuatro requisitos básicos: 1.- experiencia, 2.- necesidad de una aplicación espacial en un entorno Web, 3.- compromiso en seguimiento del proyecto y 4.- disponibilidad para retroalimentación. A partir de estos requisitos se delimitó la búsqueda y fue posible la selección de un grupo de especialistas en la gestión de riesgos.
- **Datos:** El grupo de especialistas en la gestión de riesgos son los responsables de proveer la información necesaria para la digitalización de los datos espaciales.

4.2 Descripción general de los especialistas

- **Área de aplicación:** Los especialistas en gestión de riesgos indicarán el enfoque de la aplicación espacial.
- **Validación:** Los especialistas en la gestión de riesgos validarán solo la funcionalidad de la aplicación espacial del lado del cliente.
- **Aplicación espacial:** Los datos espaciales visualizados en la aplicación cliente deberán mostrar información geográfica como: nombre y coordenadas geográficas.

4.3 Descripción general del proyecto

- **Esquema de selección de datos espaciales:** El servidor de mapas a elegir deberá permitir la adaptación de un esquema de obtención de datos espaciales en fuentes alternas, lo que implica el desarrollo de un algoritmo dentro del servidor de mapas.
- **Hardware:** se requerirá por lo menos 3 computadoras propias
Computadora 1= Servidor Web y alojamiento del servidor de mapas
Computadora 2= servidor de datos.
Computadora 3= servidor de datos.
- **Software:**
 - 1.- Digitalización de información: el software que se utilizará será acorde a la experiencia del desarrollador.
 - 2.- Interacción con datos espaciales en la Web: se requerirá un servidor de mapas, bases de datos y una aplicación cliente de código abierto.

4.4 Descripción general de común acuerdo: desarrollador, especialistas y proyecto

En esta descripción general se negocia los posibles alcances del proyecto que permitan el desarrollo de una aplicación funcional y el cumplimiento de la mayoría de las necesidades de todos los involucrados.

- **Alcances del proyecto:** De acuerdo al tiempo de desarrollo del proyecto de investigación que se tiene, no es posible cumplir con todas las exigencias de los especialistas, pero si las suficientes para el desarrollo de una aplicación funcional y presentable. Debe considerarse como prioridad el diseño del esquema de selección de datos espaciales y el desarrollo del algoritmo en el servidor de mapas.

Especificación

5

Con base en el capítulo anterior (bosquejo) se plantean en el presente capítulo las necesidades específicas del cliente en relación a la aplicación espacial, se fundamenta la selección del servidor de mapas y se analiza su funcionamiento, se describen los requerimientos del algoritmo a desarrollar y funcionamiento de los métodos optimista y pesimista propuestos, finalmente se define la arquitectura de computadoras necesaria para el desarrollo e implementación del proyecto.

5.1 Requerimientos de aplicación espacial

Especificación	Descripción
Requerimiento 1: Formato datos espaciales	Se usarán vectoriales con extensión shapefile (.SHP)
Requerimiento 2: Enfoque de los datos espaciales	Los datos espaciales están enfocados a un estudio de vulnerabilidad sísmica de puentes urbanos de Ensenada, a partir de este estudio se obtendrá la información necesaria para generar los datos espaciales
Requerimiento 3: Número de capas de datos espaciales	Se deben generar seis capas de datos espaciales, tres para el mapa base de Ensenada Baja California que contempla: delegación municipal de El Sauzal de Rodríguez, ciudad de Ensenada y delegación municipal de Chapultepec: las tres capas restantes son para los puentes urbanos de cada uno de los sitios mencionados.
Requerimiento 4: Almacenamiento de datos espaciales	Los datos espaciales deben estar almacenados en bases de datos PostgreSQL/PostGIS y de forma local, directamente de archivos vectoriales en formato (.SHP)
Requerimiento 5: Funciones de interacción	La aplicación espacial debe permitir: 1.- activar y desactivar capas de datos espaciales, 2.-consultar información geográfica al seleccionar un puente, 3.-acercar y alejar el mapa, 4.- mostrar un mapa referencia para conocer la ubicación actual de visualización.
Requerimiento 6: Tipo de software	El servidor de mapas y aplicación cliente debe ser de código abierto.
Requerimiento 7. Interfaz	La aplicación cliente debe tener gráficos acordes la gestión de riesgos.
Requerimiento 8: Catalogo de fotos	La aplicación cliente deberá permitir la visualización de un catalogo de fotos al presionar sobre un puente urbano.

Tabla 5.1 Requerimientos de aplicación espacial

5.2 Requerimientos del algoritmo

Especificación	Descripción
Requerimiento 1: Análisis del funcionamiento interno del servidor de mapas	Se debe analizar el funcionamiento interno del servidor de mapas para identificar la forma viable de adaptar el algoritmo.
Requerimiento 2: Esquema	Se debe diseñar dos esquemas uno general y otro específico que de solución a la problemática planteada en este proyecto de investigación. El esquema general mostrará el comportamiento que podría tener el servidor de mapas ante la falla de una fuente de datos y el esquema específico mostrará la representación del algoritmo de selección de fuentes de datos de acuerdo al servidor de mapas seleccionado.
Requerimiento 3: Fuentes alternas	El algoritmo debe permitir la declaración de varias fuentes alternas.
Requerimiento 4: Métodos	El algoritmo contará con dos métodos para obtener las fuentes de datos, el método OPTIMISTA y el método PESIMISTA. <u>Ambos métodos funcionarán automáticamente y de forma transparente al usuario al momento que exista la falla de una fuente de datos.</u> Estos métodos deben ser configurables, es decir el proveedor de servicios o en este caso el administrador solo podrá decidir el tipo de método que utilizará para obtener los datos espaciales de las fuentes de datos alternas y no los usuarios (clientes). El método OPTIMISTA consiste en recorrer solo una vez todas las fuentes alternas que fueron proporcionadas manualmente por el administrador, es decir si la primer fuente alterna declarada no cuenta con el dato espacial requerido, el método se encargará de buscar en la siguiente fuente alterna y así consecutivamente, en caso que el dato espacial no se encuentre en ninguna fuente de datos alternas, se notificará mediante un mensaje de error (Ver Figura 5.1). El método PESIMISTA consiste en recorrer más de una vez todas las fuentes alternas que fueron proporcionadas manualmente por el administrador del servidor de mapas, en este caso el número de veces que se recorrerán todas las fuentes alternas debe ser controlado por el administrador, si el dato espacial no se encuentra en ninguna fuente de datos alternas a pesar de los N recorridos indicados por el administrador, se notificará mediante un mensaje de error (Ver Figura 5.2).
Requerimiento 5: Informes de actividades del servidor de mapas	Se implementará un mecanismo que genere un archivo de texto el cual proporcione información de las fuentes de datos como: fuente alterna donde se obtuvo el dato espacial y fuentes alternas que no estaban disponibles.

Tabla 5.2 Requerimientos de aplicación del algoritmo

5.2.1 Esquemas de funcionamiento de los métodos OPTIMISTA Y PESIMISTA propuestos

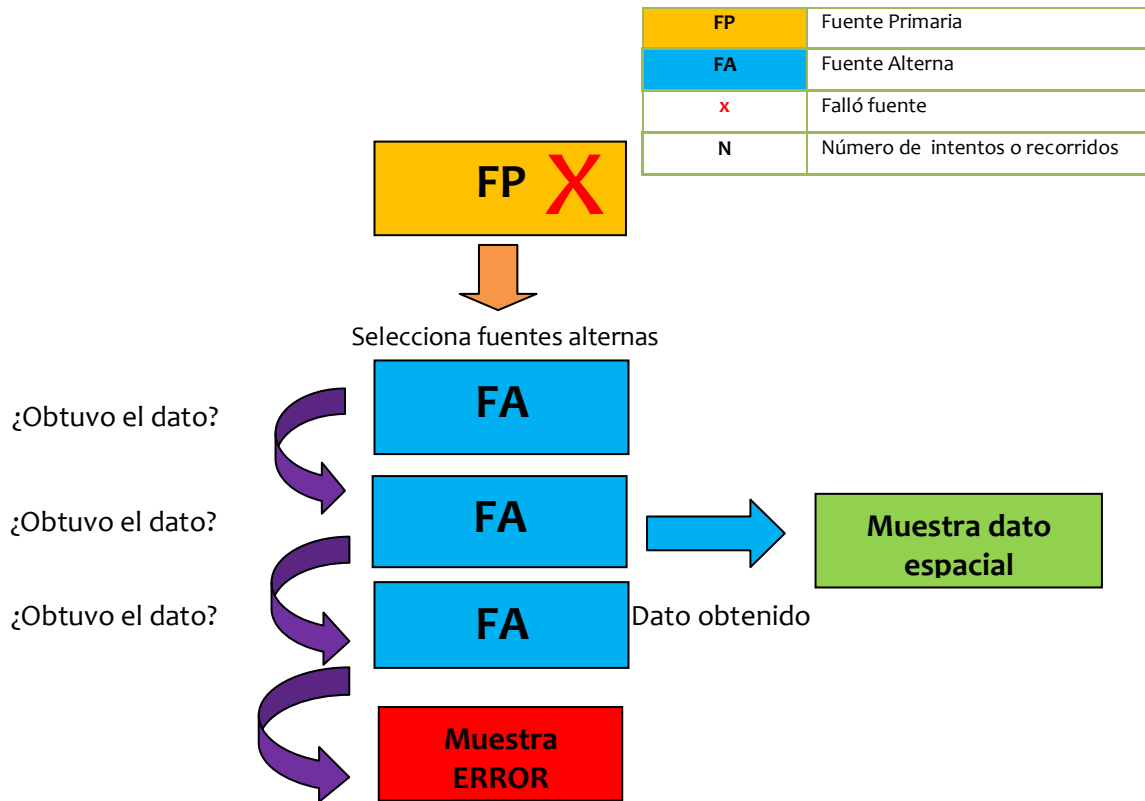


Figura 5.1 Funcionamiento método optimista

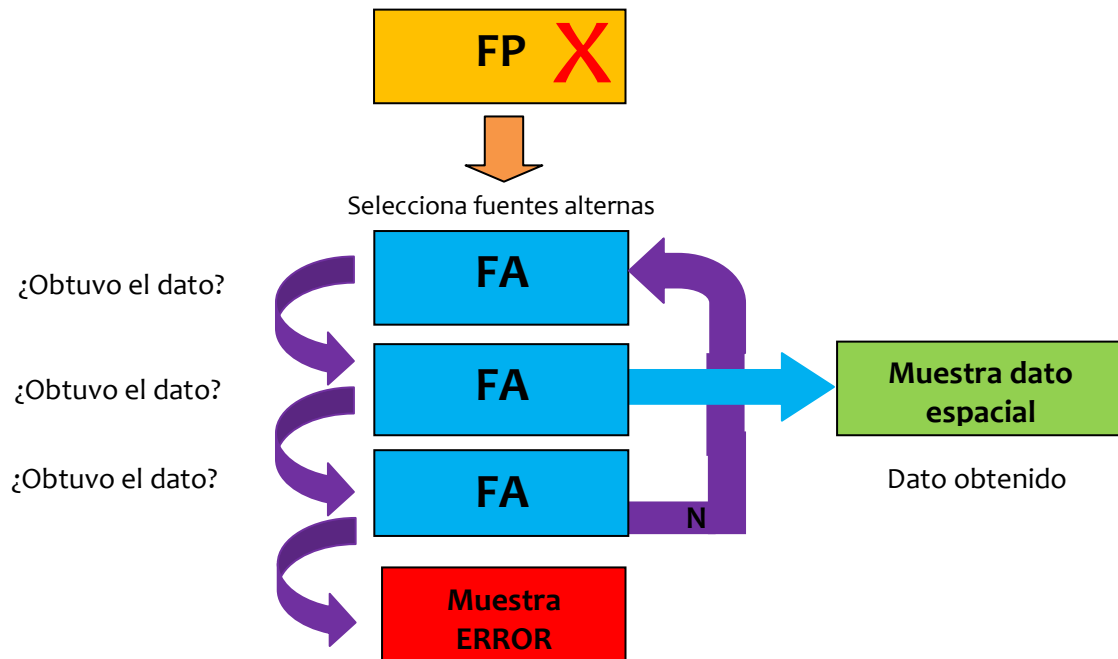


Figura 5.2 Funcionamiento método pesimista

5.3 Requerimientos de usuarios

Especificación	Descripción
Requerimiento 1: Administrador	El administrador es la persona responsable de configurar el servidor de mapas, proporcionar las fuentes de datos: primaria y alternas, elegir el método de selección de fuentes de datos, declarar fuentes alternas de datos espaciales.
Requerimiento 2: Usuario (Clientes)	Los usuarios o clientes solo podrán consultar y visualizar los datos espaciales por medio de una aplicación espacial vía web.

Tabla 5.3 Requerimientos de usuarios

5.4 Requerimientos adicionales

Especificación	Descripción
Requerimiento 1: Computadoras y software	Se necesitarán 3 computadoras personales para el desarrollo del proyecto, posteriormente se detallarán las características específicas del hardware: Dos Computadoras que funcionarán como servidor de bases de datos espaciales. Las bases de datos serán en PostgreSQL 8.2.4 + PostGIS 1.3.2 y el sistema operativo: CENTOS 5 y WINDOWS XP SP2 respectivamente. Una computadora para la instalación y configuración de un servidor Web, un servidor de mapas y una aplicación espacial. Además esta computadora almacenara datos espaciales localmente. En esta computadora deberá instalarse el siguiente software: ✓ Sistema Operativo: GNU Linux CentOS 5, versión del kernel-2.6.18 ✓ Apache-2.2 ✓ php-5.1.6, ✓ PostgreSQL 8.2.4 ✓ PostGIS 1.3.2 ✓ Librería FreeTDS 0.64 ✓ Librería Proj. 4.5.0 ✓ GEOS 2.2.3 ✓ Librería Gdal-1.4.0 ✓ UMN MapServer 5.4.2 ✓ Fusion 1.1 ✓ Joomla 1.5.7
Requerimiento 2: Red	Se trabajará bajo un esquema de red de área local en la Universidad Autónoma de Baja California (UABC) Facultad de Ingeniería Ensenada. Se deben solicitar 3 IPs al Departamento de Información Académica de la UABC y cuando menos una IP debe ser pública, para permitir el acceso desde fuera de las instalaciones de la UABC, con la finalidad de dar a conocer los avances del proyecto.
Requerimiento 3: Sitio web	Se deberá implementar un sitio web para integrar el proyecto de investigación y publicaciones de congresos y revistas que hayan sido elaborados durante la estadía en el posgrado.

Tabla 5.4 Requerimientos adicionales

5.5 Fundamentación de la selección del servidor de mapas

A partir de las tablas comparativas de los servidores de mapas de: código abierto, comerciales (Ver Tabla 2.1) y servicios gratuitos (Ver Tabla 2.2) que se elaboraron; se determinó el servidor de mapas que mejor se adaptaba a los requerimientos de la aplicación espacial planteados por los especialistas en la gestión de riesgos y los requerimientos del algoritmo de selección de fuentes de datos espaciales propuesto para este proyecto de investigación. Una de las necesidades básicas que debe cumplir el servidor de mapas a elegir es la capacidad de desarrollo interno, es decir que el servidor de mapas permita realizar mejoras al código fuente. Por lo tanto el conjunto de servidores de mapas a analizar se reduce a solamente los servidores de mapas de código abierto, ya que los servidores de mapas comerciales y de servicios gratuitos solo permiten el desarrollo externo.

A continuación, se presenta una tabla comparativa que presenta los servidores de mapas de código abierto y sus características (Ver tabla 5.5). Para llevar a cabo una selección adecuada del servidor de mapas se definieron 9 criterios:

El servidor de mapas:

1. Debe permitir el desarrollo interno y externo
2. No debe implicar un costo económico.
3. Tiene que tener soporte de diversas bases de datos
4. Tiene que tener soporte de librerías GDAL y OGR
5. Debe estar basado en múltiples estándares de la OGC
6. Funcionamiento sobre diversos sistemas operativos.
7. Contar con una comunidad de desarrolladores en la que se pueda externar preguntas o dudas, mediante: foros, correo, chat o cualquier otro medio.
8. Debe estar documentado.
9. A seleccionar de preferencia debe estar desarrollado en un lenguaje de programación que domine el desarrollador (usuario).

Con base en los criterios antes mencionados, el servidor de mapas seleccionado fue UMN MapServer, el factor que declinó la balanza hacia éste servidor de mapas fue el lenguaje de programación que se utiliza para el desarrollo interno. Debido a que el tiempo disponible con que se cuenta no es suficiente para aprender un nuevo lenguaje de programación, por lo cual se tomó la decisión de elegir un lenguaje de programación que se domina en este caso C. A pesar de que el lenguaje de programación jugó un papel importante en la selección del servidor de mapas, UMN MapServer cumple con todos los criterios establecidos, por lo cual debe considerarse una excelente selección para implementar el esquema de selección de datos espaciales.

Servidor de mapas	Desarrollo Interno/externo	Sin costo	Soporte de múltiples bases de datos	Librerías GDAL y OGR	Múltiples Estándares OGC	Diversos sistemas operativos	Comunidad de desarrollo	Documentación	Experiencia lenguaje de programación para el desarrollo interno
Geoserver 1.7.4	✓	✓	✓	✓	✓	✓	✓	✓	✓ x
Mapguide OS 2.0.2	✓	✓	✓	x	✓	✓	✓	✓	✓ X
Deegree 2.2	✓	✓	✓	x	✓	✓	✓	✓	✓ X
UMN MapServer 5.4.2	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mapnik 0.6.0	✓	NI	✓	x	✓	x	✓	✓	✓ X

Tabla 5.5 Análisis comparativo de servidores de mapas de código abierto

✓	Cumple
x	No cumple
✓ x	No suficiente experiencia
NI	No se encontró información

5.6 Servidor de mapas UMN MapServer

UMN MapServer es un servidor de mapas desarrollado por la Universidad de Minnesota (UMN) en colaboración con la Administración Nacional de Aeronáutica y del Espacio (NASA) y el Departamento de Recursos Naturales de Minnesota (MNDNR) [mapserver,2009]. Actualmente forma parte de un proyecto de la fundación Geoespacial de Código Abierto (OSGeo) la cual es una organización no lucrativa que apoya y promueve el desarrollo colaborativo de tecnologías abiertas y de datos espaciales [osgeo, 2009].

Jeff McKenna diseñó un esquema que muestra los elementos que constituyen el entorno del servidor de mapas UMN MapServer los cuales se clasifican de 3 formas: datos de entrada, servidor y datos de salida (Ver Figura 5.3) [McKenna, 2009].

1. Datos de entrada (*Fuentes de datos*)

Son los datos vectoriales y raster que pueden estar o no almacenados en bases de datos o ser proporcionados por otros servidores de mapas mediante servicios: Web Map Service (WMS), Web Feature Service (WFS).

Los servicios mencionados anteriormente comenta Tyler Mitchell que son especificaciones web de la OGC para acceder y compartir datos a través de internet [Mitchell, 2005]. Además Tyler Mitchell define el propósito principal de cada servicio, mientras que WMS sirve para compartir y solicitar mapas de datos vectoriales y raster en formato de imagen plana; el WFS solamente es con mapas de datos vectoriales y atributos en formato Geographic Markup Language (GML).

GML es un formato basado en XML para el intercambio de datos espaciales y tabulares (descriptivos) [Mitchell, 2009]

2. Servidor

El servidor está constituido por 4 partes:

- Archivo Mapfile (.MAP):
Archivo de configuración del servidor de mapas
- Aplicación del servidor de mapas CGI:
Ejecutable del servidor de mapas. Es el modo más sencillo de UMN MapServer, debido a que no requiere programación para su funcionamiento, solamente necesita el archivo de configuración Mapfile (.MAP) y una aplicación espacial que solicite los datos espaciales [Mitchell,2009]. Según el World Wide Web Consortium (W3C), CGI es conocido como la Interface de Puerta Común (Common Gateway

Interface), es un estándar que se utiliza para transmitir información entre una aplicación cliente y el servidor web. Un programa CGI puede escribirse en cualquier lenguaje que genere un ejecutable y que permita manipular cadenas de caracteres [W3C, 2009].

- **Aplicación MapScript:**

Jeff McKenna menciona que la aplicación MapScript es un modulo de UMN MapServer proporciona una interfaz de scripts para el servidor de mapas, que permite la construcción de aplicaciones espaciales de escritorio y web, además comenta que MapScript puede usarse de forma independiente a aplicación CGI y que requiere de programación de código para la interacción con las fuentes de datos espaciales [McKenna, 2009].

- **Servidor web:**

Permite alojamiento de aplicaciones y el intercambio de información

3. Datos de salida

Los datos de salida están relacionados al modo de visualización de los datos y/o generación de servicios. El servidor de mapas UMN MapServer tiene la capacidad de recibir servicios proporcionados por otros servidores y a su vez de generarlos, en este caso los servicios Web Map Service (WMS), Web Feature Service (WFS) y Web Coverage Service (WCS). Este último servicio según el Open Geospatial Consortium comenta que sirve para habilitar la el acceso interoperable a los datos espaciales de coberturas, estas coberturas se refieren a imágenes de satélite, fotografías áreas digitales, datos de elevación digital y otros fenómenos representados por valores en cada punto que se deseen medir [OGC, 2009].

Otros datos de salida que el del servidor de mapas es capaz de generar para su visualización son el formato flash Shockwave Flash (SWF), formato vectorial Scalable Vector Graphics (SVG) y Portable Document Format (PDF), Y finalmente HTML + Imagen permite una manipulación de la imagen visualizada mediante controles HTML y en el caso de Imagen la visualización es estática.

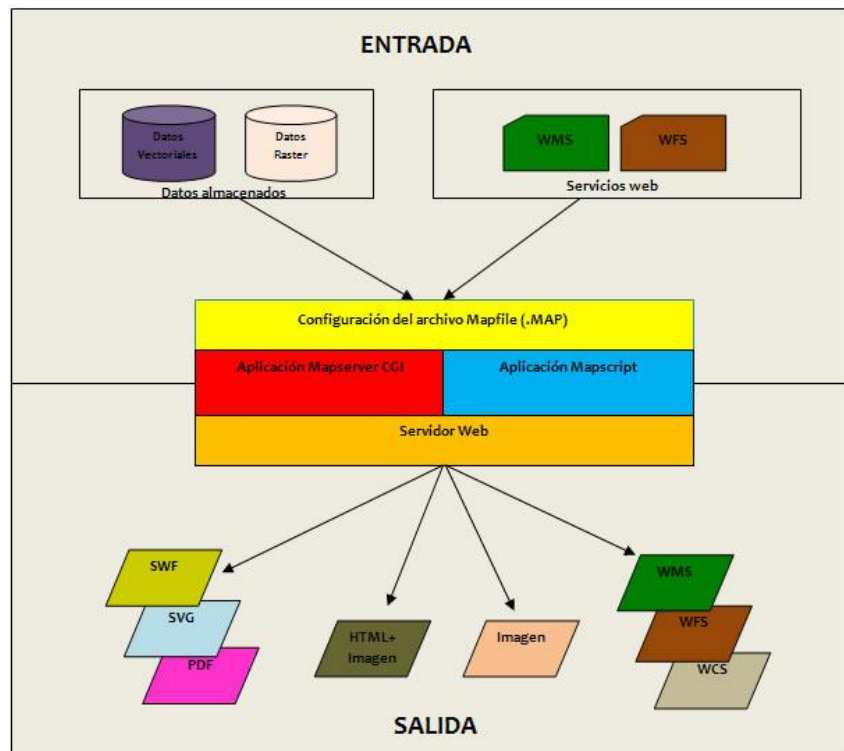


Figura 5.3. Arquitectura del servidor de mapas UMN MapServer
Fuente: Adaptado de McKenna (2009)

5.6.1 Funcionamiento integrado de los elementos del servidor de mapas

A continuación se muestra el funcionamiento de los elementos que integran el servidor de mapas (Ver Figura 5.4):

1. Aplicación espacial (cliente) a servidor de mapas:

A partir de una aplicación espacial se realiza la petición de datos espaciales al servidor Web, utilizando como medio de comunicación una Intranet o Internet, además en esta petición se solicita la invocación del CGI del servidor de mapas que opera en el servidor web. En caso que la aplicación espacial utilice MapScript para obtener datos espaciales, el CGI de MapServer no es requerido.

2. Servidor de mapas a fuentes de datos e inversa:

El servidor web realiza dos notificaciones a la aplicación CGI del servidor de mapas: 1.- que han sido invocado sus servicios y 2.- los datos espaciales son requeridos del archivo Mapfile.

La aplicación CGI del servidor de mapas procesa esta información y accesa el archivo Mapfile, para ayudarle a responder los siguientes cuestionamientos: ¿qué datos espaciales se visualizarán? , ¿cómo se visualizarán? y ¿donde están ubicadas las fuentes de datos?; entonces que a partir del Mapfile obtiene la ubicación de las fuentes de datos y características de los datos espaciales. De forma consecuente el CGI realiza una consulta a las fuentes de datos, obtiene los datos espaciales y se prepara para generar las imágenes que serán enviadas a la aplicación cliente.

3. Servidor de mapas a aplicación espacial (cliente):

Una vez que la aplicación CGI del servidor de mapas conoce la ubicación de las fuentes de datos y las características de los mismos, genera las imágenes y enviar respuesta a la aplicación cliente.

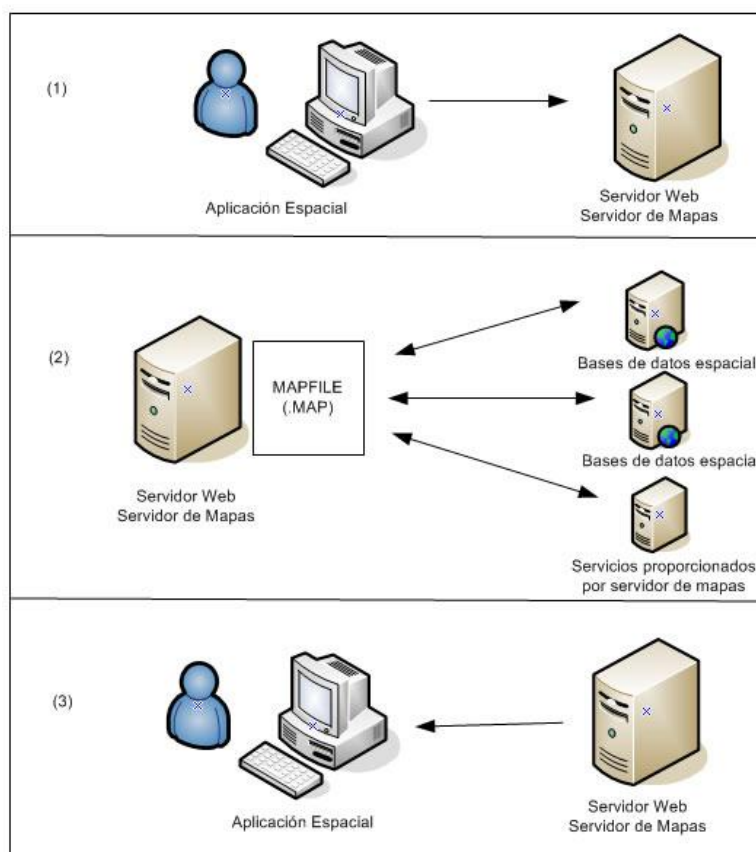


Figura 5.4. Funcionamiento integrado de los elementos del servidor de mapas

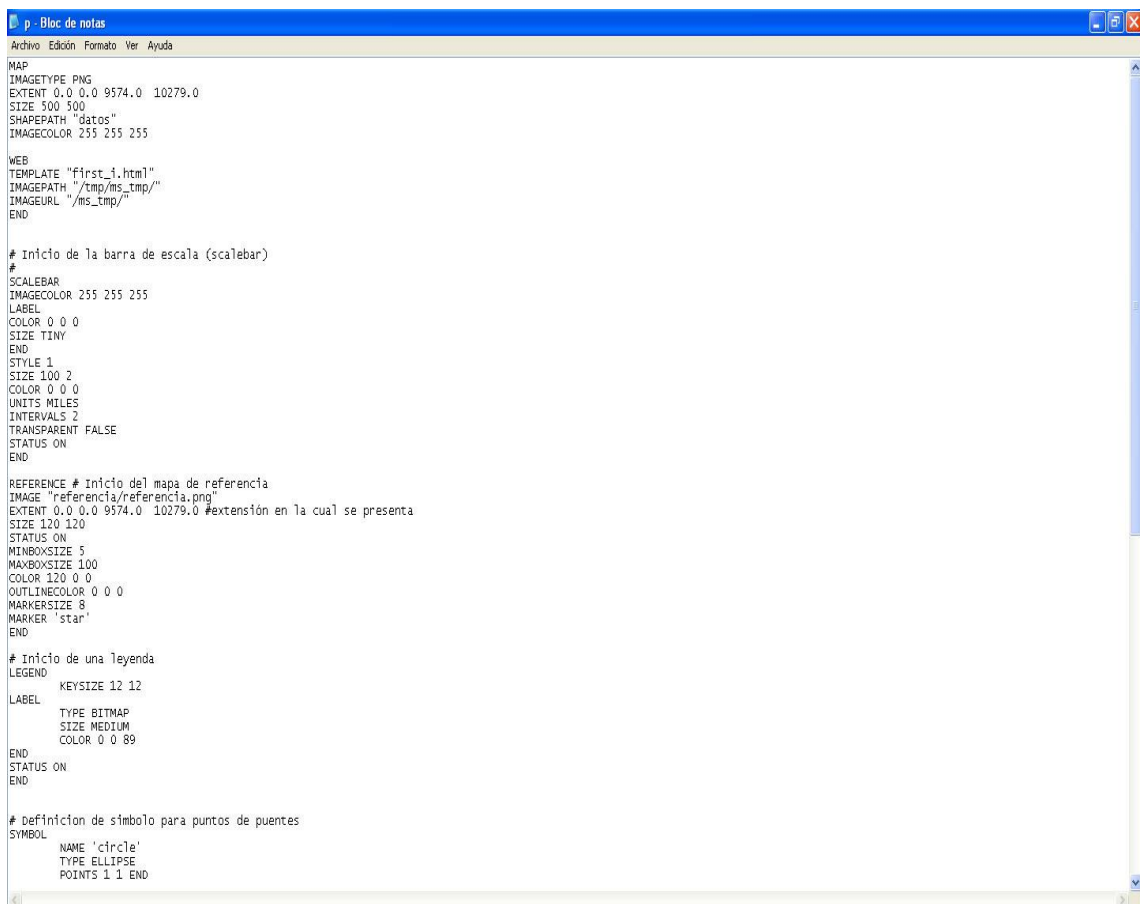
5.6.2 ¿Qué es el MAPFILE (.MAP)?

Jeff McKenna considera al Mapfile como el corazón del servidor de mapas UMN MapServer [Mckenna, 2009]. El Mapfile es un archivo de configuración de texto plano con extensión (.MAP), que ayuda a dar respuestas a las interrogantes del CGI de MapServer (Ver Figura 5.5).

Jeff Mackenna además explica que en el archivo Mapfile se definen: las fuente datos, leyendas, las proyecciones geográficas, la definición de características de los datos espaciales como color, etiquetas, símbolos, plantillas, entre otras [Mackenna, 2009]. Entonces partir de la información proporcionada en el archivo Mapfile, el servidor de mapas obtendrá los datos espaciales de las fuentes de datos y generará los mapas de imágenes [Kropla, 2005].

Finalmente dos características peculiares del Mapfile son:

- 1.- Fácil edición con cualquier editor de textos
- 2.- *Case sensitive*, no distingue entre mayúsculas y minúsculas.



```
MAP
IMAGETYPE PNG
EXTENT 0.0 0.0 9574.0 10279.0
SIZE 500 500
SHAPEPATH "datos"
IMAGECOLOR 255 255 255

WEB
TEMPLATE "first_1.html"
IMAGEPATH "/tmp/ms_tmp/"
IMAGEURL "/ms_tmp/"
END

# Inicio de la barra de escala (scalebar)
#
SCALEBAR
IMAGECOLOR 255 255 255
LABEL
COLOR 0 0 0
SIZE TINY
END
STYLE 1
SIZE 100 2
COLOR 0 0 0
UNITS MILES
INTERVALS 2
TRANSPARENT FALSE
STATUS ON
END

REFERENCE # Inicio del mapa de referencia
IMAGE "referencia/referencia.png"
EXTENT 0.0 0.0 9574.0 10279.0 #extensión en la cual se presenta
SIZE 120 120
STATUS ON
MINBOXSIZE 5
MAXBOXSIZE 100
COLOR 120 0 0
OUTLINECOLOR 0 0 0
MARKERSIZE 8
MARKER 'star'
END

# Inicio de una leyenda
LEGEND
KEYSIZE 12 12
LABEL
TYPE BITMAP
SIZE MEDIUM
COLOR 0 0 89
END
STATUS ON
END

# Definición de símbolo para puntos de puentes
SYMBOL
NAME 'circle'
TYPE ELLIPSE
POINTS 1 1 END
```

Figura 5.5 Ejemplo de archivo Mapfile.

5.6.3 Mapfile (.MAP): Estructura interna

El Mapfile del servidor de mapas UMN MapServer está estructurado con base a siete etiquetas principales denominadas objetos: 1.- MAP, 2.- LAYER, 3.-LEGEND, 4.-QUERYMAP, 5.-REFERENCE, 6.-WEB, 7.-SCALEBAR y un conjunto de etiquetas claves (keywords) independientes que son atributos (Ver Figura 5.6). Cabe señalar que dentro de cada etiqueta objeto existen etiquetas claves independientes y a su vez pueden existir otras etiquetas objeto, el término que podría definir mejor este comportamiento es etiquetas anidadas.

Por otro lado el objetivo de esta temática no es detallar la funcionalidad de todas las etiquetas objetos y etiquetas claves independientes. Para conocer la funcionalidad todas las etiquetas puede consultarse la siguiente página <http://mapserver.org/mapfile/index.html#mapfile>. Lo que sí se pretende en esta temática es: mostrar la estructura total del Mapfile con una breve descripción de las etiquetas principales; esto servirá para comprender la estructura interna del archivo Mapfile (.MAP) y ayudará responder la siguiente pregunta: ¿dónde y cómo se adaptará el esquema de selección de fuentes de datos espaciales propuesto?

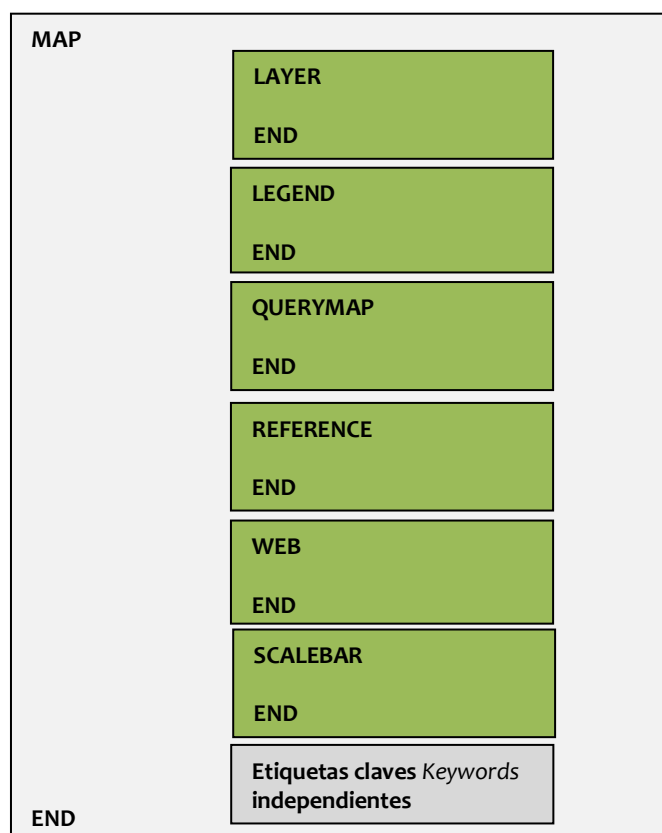


Figura 5.6. Estructura del archivo del Mapfile (.MAP)

5.6.4 MAP

La etiqueta objeto MAP es utilizada para inicializar un archivo Mapfile, esta etiqueta engloba a las etiquetas objeto: LAYER, LEGEND, QUERYMAP, REFERENCE, WEB (Ver Figura 5.6) y un conjunto de etiquetas claves independientes (Ver figura 5.7).

```

MAP
    INCLUDE "" // incluye .maps
    ANGLE [double]
    CONFIG [key] [value] //para errores cuando los datos se pierden
    DATAPATTERN [regular expression]
    DEBUG [off|on|0|1|2|3|4|5]
    EXTENT [minx] [miny] [maxx] [maxy]
    FONTSET [filename]
    IMAGECOLOR [r] [g] [b]
    IMAGEQUALITY [int]
    IMAGETYPE [gif|png|jpeg|wbmp|gtiff|swf|userdefined]
    INTERLACE [on|off]
    MAXSIZE [integer]
    NAME [name]
    SHAPEPATH [filename]
    SIZE [x][y]
    STATUS [on|off]
    SYMBOLSET [filename]
    SYMBOL END
    TEMPLATEPATTERN [regular expression]
    TRANSPARENT [on|off]
    UNITS [feet|inches|kilometers|meters|miles|dd]
    RESOLUTION [int]
    SCALEDENOM [double]
    SCALE [double]

END

```

Figura 3.7 Etiquetas claves independientes que pertenecen a MAP

5.6.5 LAYER

Etiqueta objeto principal donde se configuran el tipo de acceso a las fuentes de datos: base de datos, local, servicios; así como también los atributos de los datos espaciales como: color de relleno, color de contorno, tipo de dato espacial, entre otros.

La etiqueta objeto LAYER contiene las siguiente etiquetas objeto: CLASS, GRID, JOIN y FEATURE; y también etiquetas claves independientes (Ver Figura 5.8).

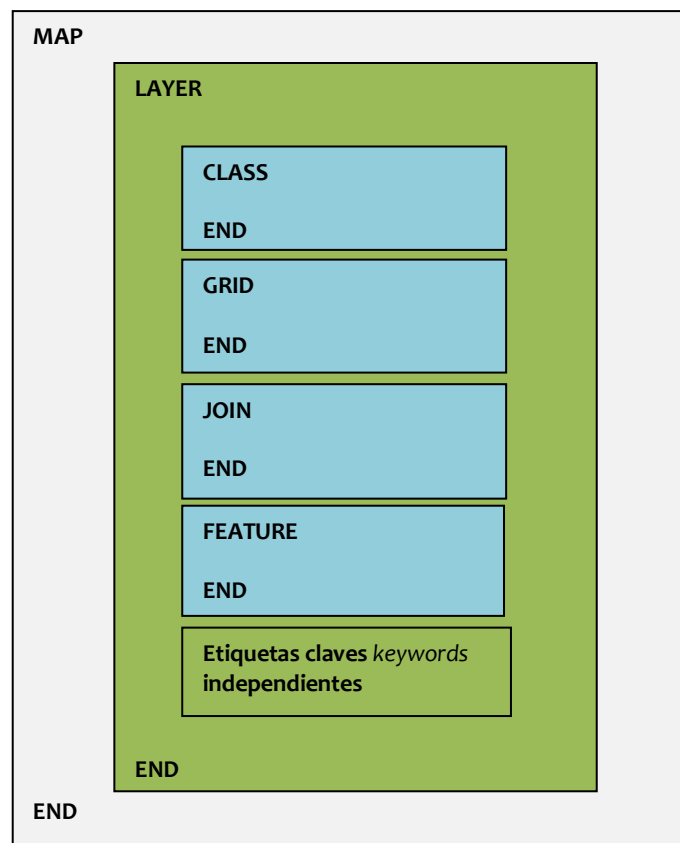


Figura 5.8 Estructura de etiquetas objetos y etiquetas claves independientes que pertenecen a LAYER

5.6.6 LAYER: CLASS

La etiqueta objeto CLASS pertenece a la etiqueta objeto LAYER. La etiqueta objeto CLASS a su vez contiene tres etiquetas objeto: LABEL, STYLE y SYMBOL; y también etiquetas claves independientes (Ver Figura 5.9).

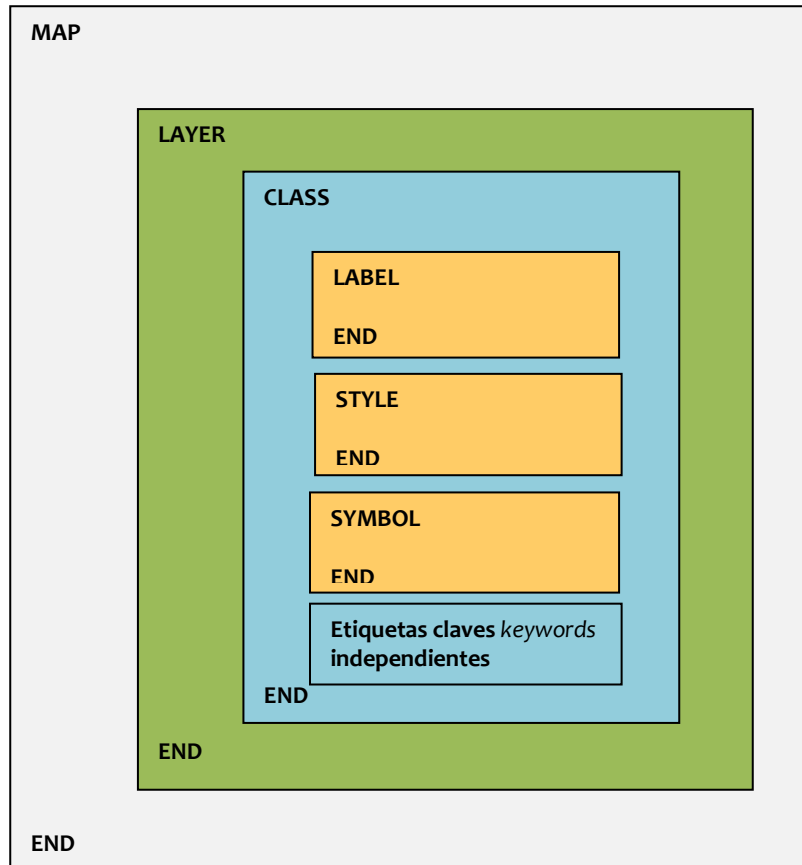


Figura 5.9 Estructura de etiquetas objetos y etiquetas claves independientes que pertenecen a CLASS

5.6.7 LAYER: CLASS: LABEL

La etiqueta objeto CLASS contiene a la etiqueta objeto LABEL la cual solo contiene a diversas etiquetas claves independientes (Ver Figura 5.10)

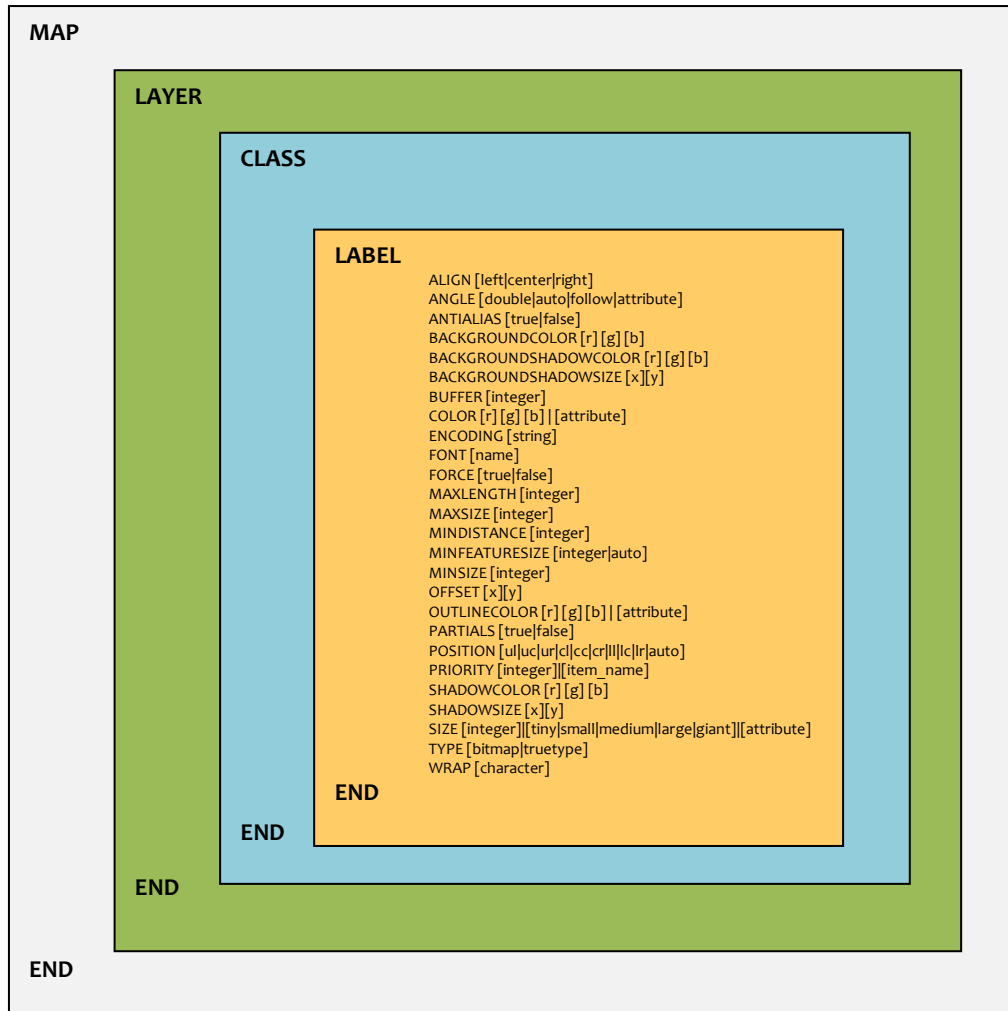


Figura 5.10 Estructura de la etiqueta objeto LABEL

5.6.8 LAYER: CLASS: STYLE

La etiqueta objeto CLASS contiene a la etiqueta objeto STYLE la cual solo contiene a diversas etiquetas claves independientes (Ver Figura 5.11)

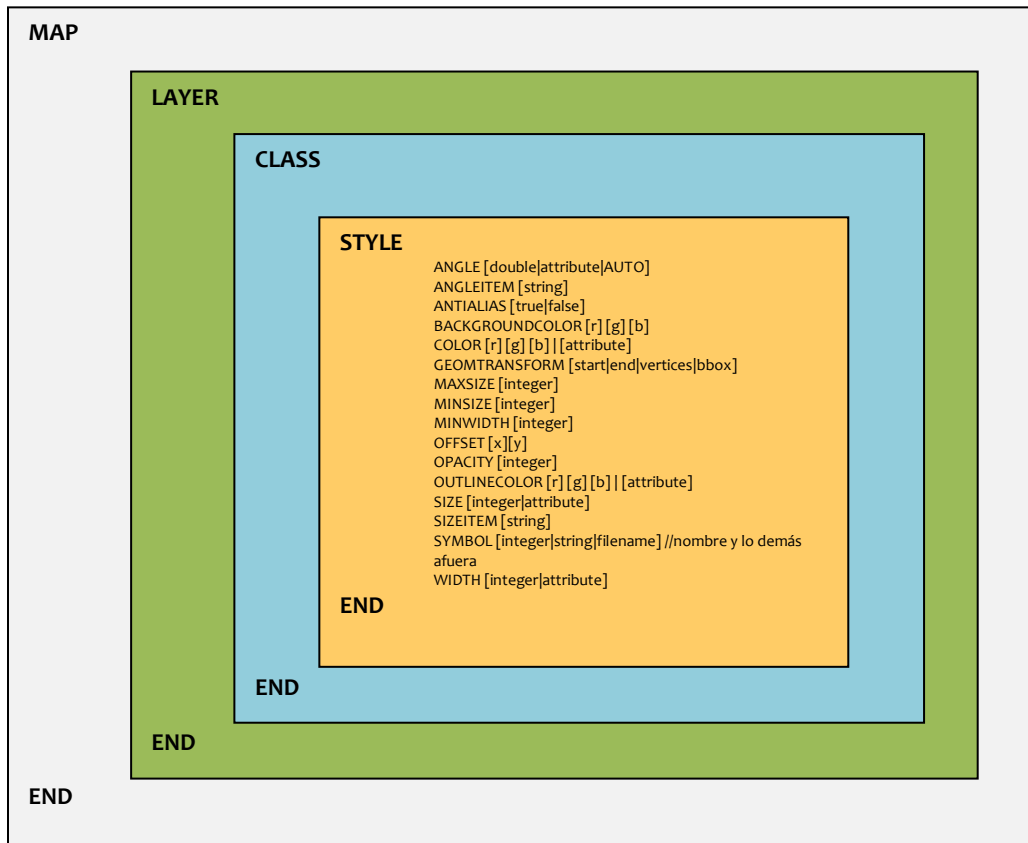


Figura 5.11. Estructura de la etiqueta objeto STYLE

5.6.9 LAYER: CLASS: SYMBOL

La etiqueta objeto CLASS contiene a la etiqueta objeto SYMBOL la cual solo contiene a diversas etiquetas claves independientes (Ver Figura 5.12)

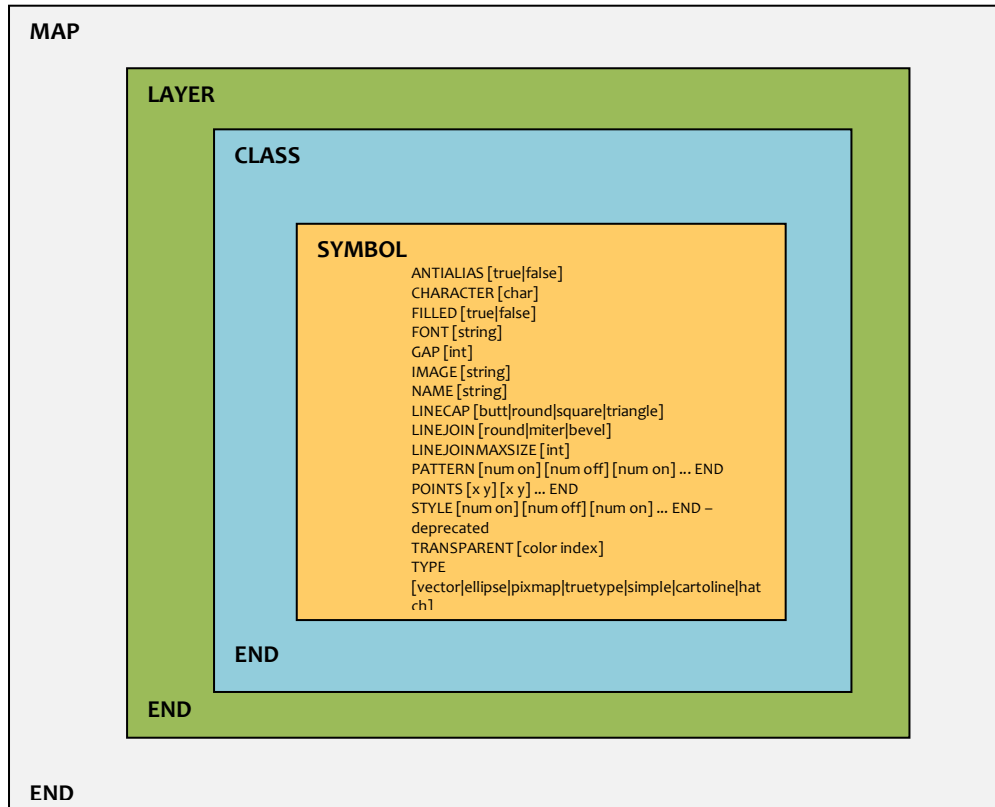


Figura 5.12. Estructura de la etiqueta objeto SYMBOL

5.6.10 LAYER: CLASS: Etiquetas claves independientes

La etiqueta objeto CLASS contiene diversas etiquetas claves independientes (Ver Figura 5.13)

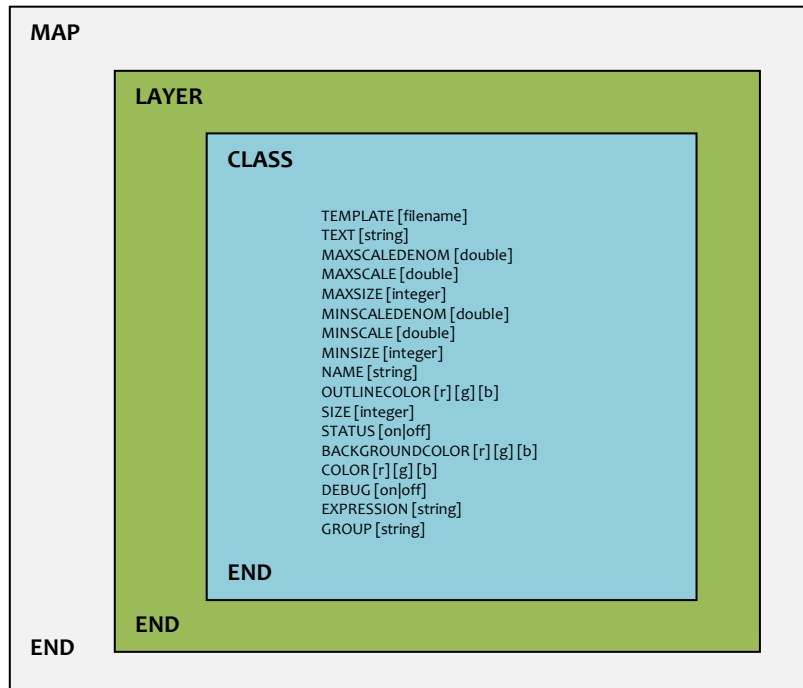


Figura 5.13. Etiquetas claves independientes que contiene la etiqueta objeto CLASS

5.6.11 LAYER: GRID

La etiqueta objeto GRID pertenece a la etiqueta objeto LAYER. La etiqueta objeto GRID a su vez contiene etiquetas claves independientes (Ver Figura 5.14).

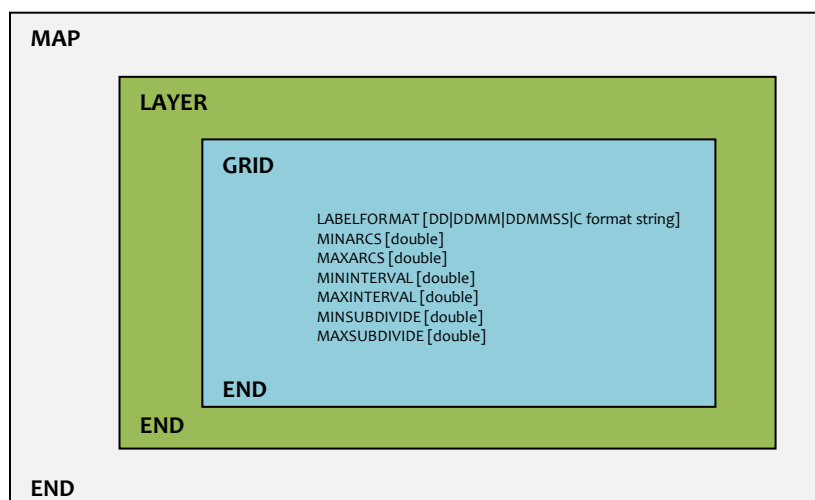


Figura 5.14. Etiquetas claves independientes que contiene la etiqueta objeto GRID

5.6.12 LAYER: JOIN

La etiqueta objeto JOIN pertenece a la etiqueta objeto LAYER. La etiqueta objeto GRID a su vez contiene etiquetas claves independientes (Ver Figura 5.15).

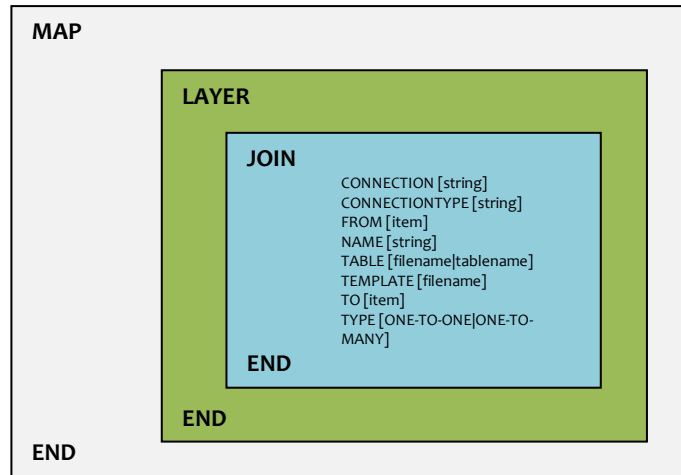


Figura 5.15. Etiquetas claves independientes que contiene la etiqueta objeto JOIN

5.6.13 LAYER: FEATURE

La etiqueta objeto FEATURE pertenece a la etiqueta objeto LAYER. La etiqueta objeto FEATURE a su vez contiene etiquetas claves independientes (Ver Figura 5.16).

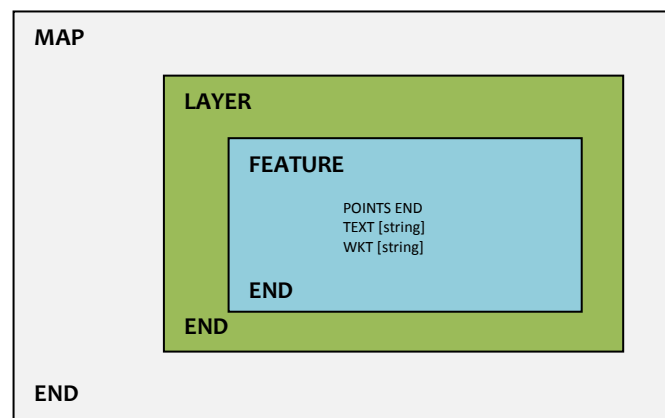


Figura 5.16. Etiquetas claves independientes que contiene la etiqueta objeto FEATURE

5.6.14 LAYER: Etiquetas claves independientes

La etiqueta objeto LAYER contiene diversas etiquetas claves independientes (Ver Figura 5.17)

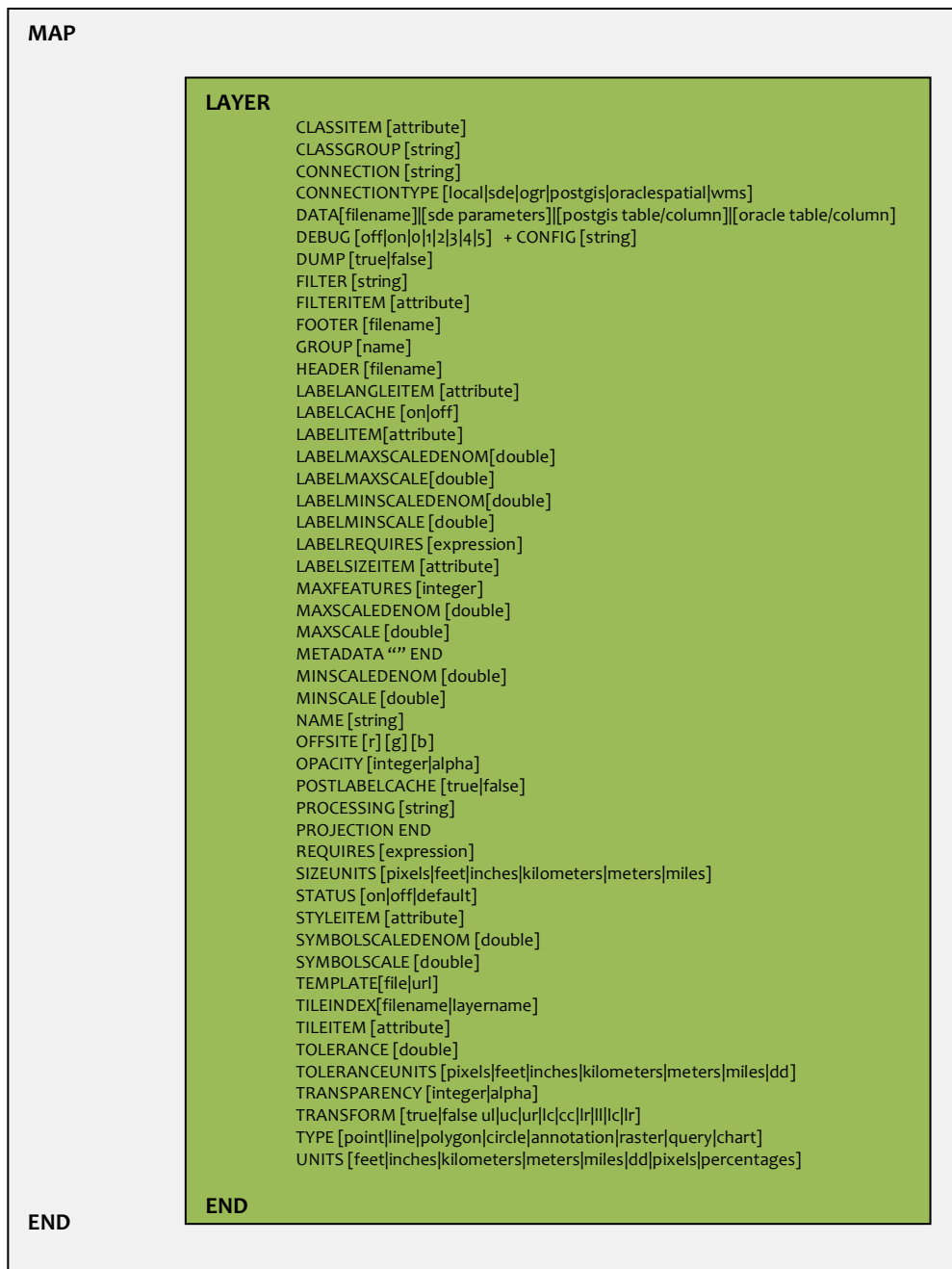


Figura 5.17. Etiquetas claves independientes que contiene la etiqueta objeto LAYER

5.6.15 LEGEND

Etiqueta objeto principal que sirve para mostrar una breve descripción e imagen de las capas de datos espaciales que se visualizan en una aplicación espacial (Ver Figura 5.18)

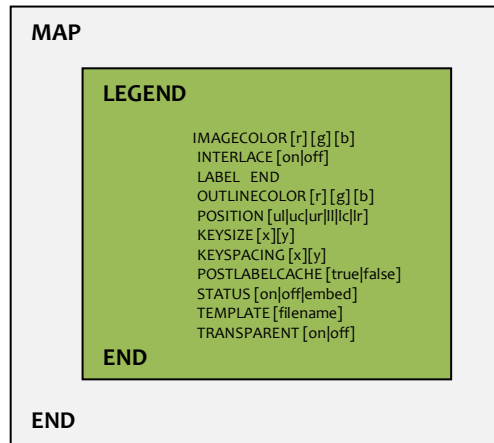


Figura 5.18. Etiquetas claves independientes que contiene la etiqueta objeto LEGEND

5.6.16 QUERYMAP

Etiqueta objeto principal que sirve para habilitar la consulta de información geográfica (información alfanumérica) de los datos espaciales (Ver Figura 5.19).

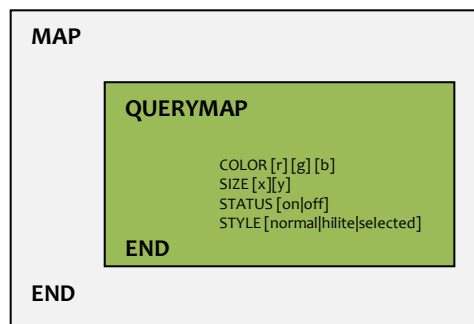


Figura 3.19. Etiquetas claves independientes que contiene la etiqueta objeto QUERYMAP

5.6.17 REFERENCE

Etiqueta objeto principal que sirve para proporcionar un mapa de referencia que muestre la ubicación actual de visualización en una aplicación espacial (Ver Figura 5.20)

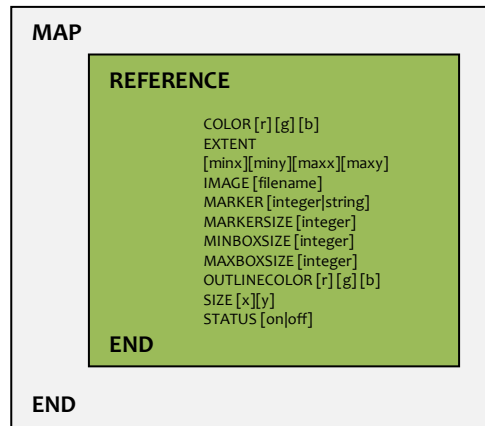


Figura 5.20. Etiquetas claves independientes que contiene la etiqueta objeto REFERENCE

5.6.18 WEB

Etiqueta objeto principal en la cual se declaran las rutas para almacenar: las imágenes temporales mapas y el archivo de actividades (logs) generados por el servidor de mapas; además se utiliza para realizar re-direccionamientos a paginas, entre otros (Ver Figura 5.21).

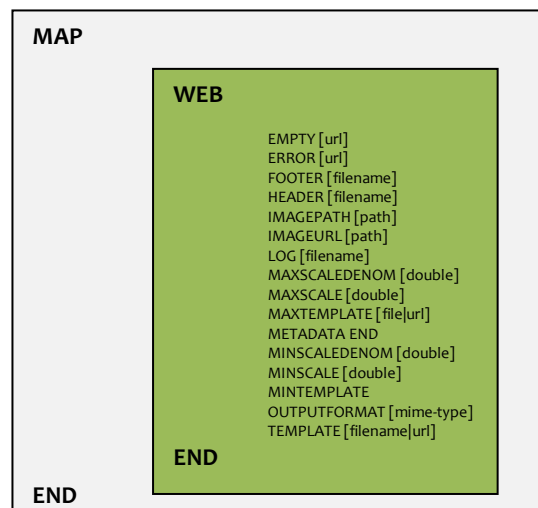


Figura 5.21. Etiquetas claves independientes que contiene la etiqueta objeto WEB

5.6.19 SCALEBAR

Etiqueta objeto principal que sirve para mostrar una barra medidora en una aplicación espacial; esta barra indica: los kilómetros, pulgadas, metros, millas y pies a los que está representado el mapa (Ver Figura 5.22).

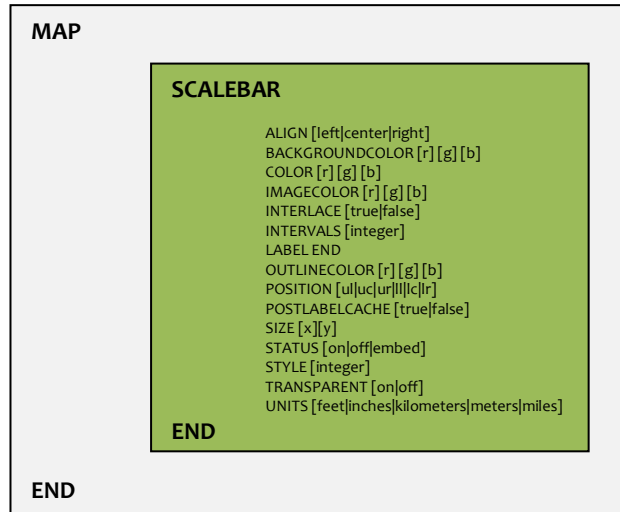


Figura 5.22. Etiquetas claves independientes que contiene la etiqueta objeto WEB

5.6.20 Comentarios, etiquetas de inicialización y finalización en Mapfile.

- **Etiqueta objeto MAP**
Para inicializar un archivo Mapfile
- **Etiqueta objeto END**
Para finalizar cualquier etiqueta objeto
- **#** se utiliza para insertar un comentario en el archivo Mapfile

5.7 Casos de uso

- **Casos de uso del cliente:** Interacción con los datos espaciales a través de una aplicaciones espacial

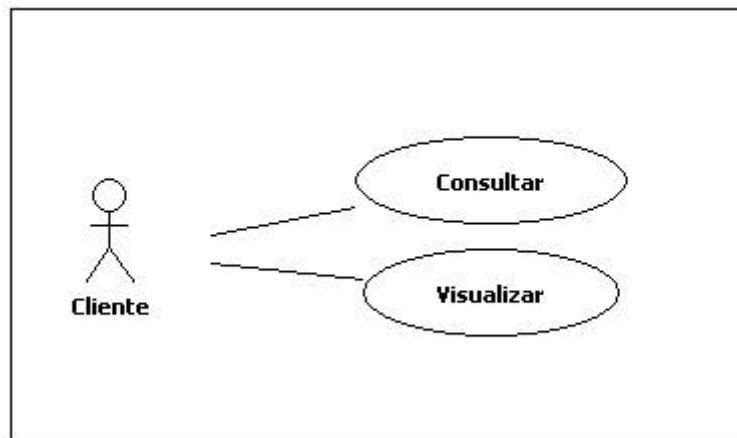


Figura 5.23 Caso de uso del cliente

Nombre:	Consultar información geográfica de datos espaciales
Actor:	Cliente
Propósito:	El cliente obtendrá información geográfica que describe a los datos espaciales visualizados en la aplicación espacial. Solamente los datos espaciales que representan los puentes están habilitados para realizar este tipo de consulta.
Descripción:	El cliente debe seleccionar la caja de verificación que identifica a la capa de datos espacial, automáticamente se envía la solicitud del dato espacial al servidor de mapas, este obtendrá el dato espacial y lo desplegará en la aplicación espacial. En este momento el cliente puede elegir el icono de consulta y obtendrá la información geográfica relacionada al dato espacial.

Tabla 5.6 Caso de uso cliente consultar información geográfica

Nombre:	Visualizar datos espaciales
Actor:	Cliente
Propósito:	El cliente visualizará los datos espaciales solo si se habilita la caja de verificación de la aplicación espacial
Descripción:	El cliente debe seleccionar la caja de verificación que identifica a la capa de datos espacial, automáticamente se envía la solicitud del dato espacial al servidor de mapas, este obtendrá el dato espacial y lo desplegará en la aplicación espacial.

Tabla 5.7 Caso de uso cliente visualización de datos espaciales

- **Administrador:** Configuración del Mapfile del servidor de mapas UMN MapServer

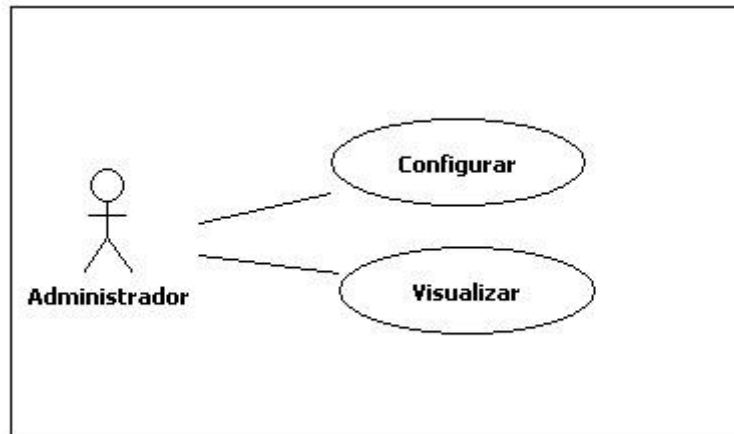


Figura 5.24 Caso de uso del administrador

Nombre:	Configurar Mapfile
Actor:	Administrador
Propósito:	El administrador del servidor de mapas configurará el Mapfile con los métodos y fuentes alternas datos espaciales.
Descripción:	El administrador del servidor de mapas solo necesitará configurar una sola vez los métodos y fuentes alternas de datos espaciales en el Mapfile.

Tabla 5.8 Caso de uso administrador configuración del Mapfile

Nombre:	Visualizar informe
Actor:	Administrador
Propósito:	El administrador del servidor de mapas podrá visualizar el informe de actividades de las fuentes de datos alternas.
Descripción:	El administrador del servidor de mapas debe abrir un archivo de texto que será generado por el servidor de mapas. Este archivo contiene un informe de actividades de las fuentes alternas

Tabla 5.9 Caso de uso administrador visualización de informe

5.8 Hardware

Para desarrollo e implementación del proyecto de investigación se utilizaron 3 computadoras personales y un switch con las siguientes características.

1. Una computadora para el servidor de mapas

- Procesador Pentium D 3.0
- 512 MB de memoria RAM
- Tarjeta de Red PCI 10/100Mb o Integrada
- Tarjeta de Video SVGA
- Disco Duro 250GB

2. Dos computadoras como servidor de bases de datos

- Procesador Pentium III 900 Mhz
- 128 MB de memoria RAM
- Tarjeta de Red PCI 10/100Mb o Integrada
- Tarjeta de Video SVGA
- Disco Duro 40 GB

3. Switch Zonet 8 Puertos

5.9 Configuración de la red de trabajo

Para llevar a cabo la instalación de sistemas operativos, bases de datos, gestores de bases de datos, librerías y el servidor de mapas fue necesaria la previa configuración de la red de trabajo con la finalidad de identificar cuáles computadoras funcionarían como servidores de bases de datos y cual como servidor de mapas (Ver Figura 5.25)

El servidor de mapas se instaló en la computadora que contaba con mejores características: como espacio en disco duro, memoria RAM y procesador; las dos computadoras restantes que contaban con características regulares se utilizaron como servidores de bases de datos. Por otro lado se logró conseguir 3 IPs públicas y un dominio <http://geoconstructor.ens.uabc.mx> los cuales están registrados en la UABC Ensenada, cabe señalar que se compraron dos dominios adicionales a un proveedor de servicios: <http://www.geoconstructor.com> y <http://www.geoconstructor.net> y estos dominios fueron redireccionados temporalmente a una de las IPs de la UABC, con la finalidad de dar a conocer el nombre de Geoconstructor y para continuar el proyecto en un futuro.

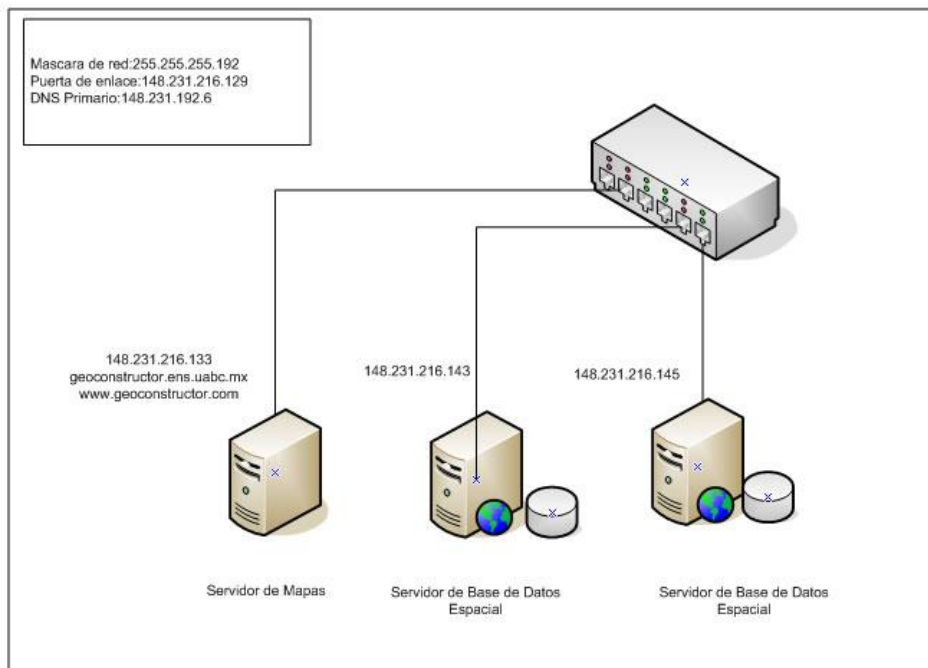
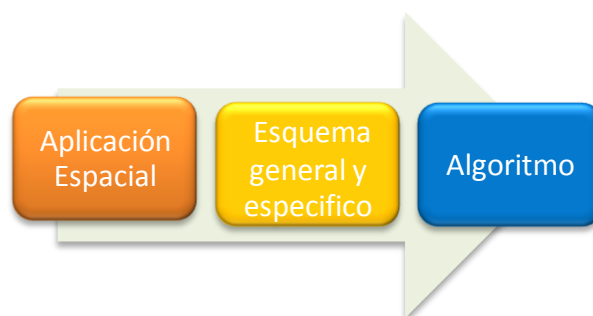


Figura 5.25. Configuración de la red de trabajo

Desarrollo

6

Este capítulo se divide en tres fases: 1.- diseño y desarrollo de aplicación espacial, 2.- diseño de esquema de selección de datos espaciales y 3.- desarrollo del algoritmo de selección de datos espaciales.



Fase 1

Diseño y desarrollo de
aplicación espacial

Fase 1: Diseño y desarrollo de aplicación espacial

La motivación de diseñar y desarrollar una aplicación espacial enfocada a la gestión de riesgos se debe a que en los últimos años en el planeta se han originado sismos y terremotos que han causado muertes y afectado la infraestructura de ciudades con derrames económicos cuantiosos. Lamentablemente hoy en día es difícil predecir con exactitud el lugar y tiempo en el que sucederá un sismo o terremoto; sin embargo es posible realizar una gestión de riesgos que permita prevenir y algunos casos reducir los efectos del fenómeno, en relación a los costos económicos en daños a infraestructura y la pérdida de vidas.

La región de Baja California, está contemplada como una zona con actividad sísmica moderada, esto se debe a que se encuentra rodeada por una cantidad considerable de fallas sísmicas (Ver Figura 6.1) según el boletín del grupo de Red Sísmica del Noroeste de México (RESNOM) el cual pertenece al Departamento de Sismología división ciencias de la tierra del Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) informó que tan solo durante el año 2005 se registraron 1430 sismos (Ver Figura 6.2) con magnitudes entre 0.7 y 5.2 [Grupo RESNOM, 2006].

A consecuencia de que la región de Baja California es una zona moderadamente sísmica, especialistas en la gestión de riesgos realizaron un estudio que determina el índice de vulnerabilidad de 29 puentes de la ciudad de Ensenada Baja California, México. La información recabada en este estudio marca la pauta y sirve de base para realizar estrategias y mitigaciones de riesgo en caso de un desastre; ya que antes no existían estudios tendientes a la prevención del desastre que pudiera ocasionar la falla de uno o varios de estos puentes debido a la ocurrencia de un terremoto [Juárez et. al., 2007].

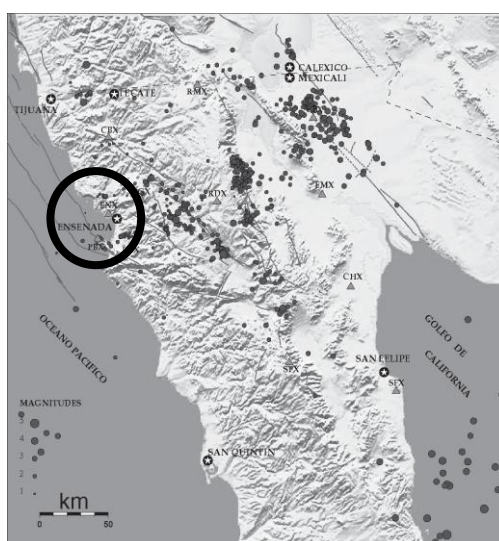


Figura 6.1 Principales fallas sísmicas de la región de Baja California, México.
Encerrado en un círculo la ciudad de Ensenada.
Fuente grupo RESNOM (2006).

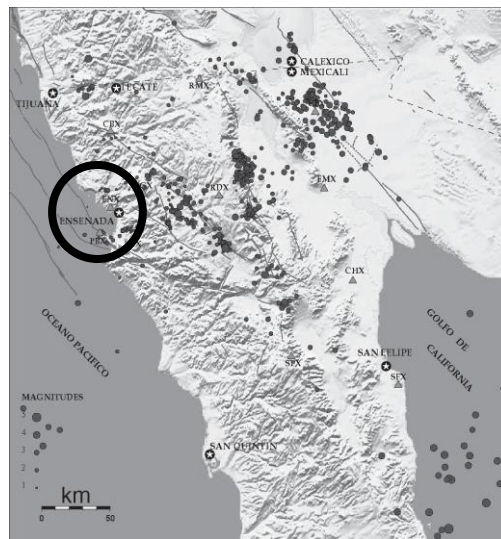


Figura 6.2 Epicentros de temblores de la región de Baja California, México registrados en el transcurso del año 2005. Encerrado en un círculo la ciudad de Ensenada.

Fuente grupo RESNOM (2006).

De acuerdo al estudio mencionado en el párrafo anterior se implementó una aplicación espacial vía web basado en tecnología de código abierto, que permite integrar y consultar la información de los puentes, anexando un catálogo de fotos que ayudan describir con mayor claridad el tipo de estructuras; esta aplicación ayudará a determinar planes de contingencia en caso de desastre y en consecuencia una toma de decisiones más efectiva.

6.1 Digitalización

A partir de la información del estudio de vulnerabilidad sísmica de 29 puentes urbanos de la ciudad de Ensenada (Ver Tabla 6.1), se crearon los datos espaciales con el software ArcView 3.2, cada uno de los datos espaciales fueron almacenados como shapefile (SHP), generándose automáticamente dos archivos con extensión: DBF y SHX; en el caso del archivo SHP se define como el archivo principal, este posee la geometría de los objetos: puntos, líneas o polígonos que fueron trazados; el archivo DBF es la base de datos, ésta almacena la información de los atributos de los objetos y finalmente el SHX funge como el enlace entre los archivos SHP y DBF almacenando sus índices respectivamente.

ID	Vialidad	Latitud(GPS)	Longitud(GPS)	Obstáculo	Largo	Ancho	Núm. Vanos.	Índice vulnerabilidad	Estado
1	Bvld. Lázaro Cárdenas	31°51'33.5"N	116°37'11.7"W	Arroyo Ensenada	61.50	2 secciones 15.50 m c/u	4	2.3	Poco Vulnerable
2	Calle Segunda	31°51'40.6"N	116°36'58.9"W	Arroyo Ensenada	54.70	15.5	3	3.4	Poco Vulnerable
3	Av. Juárez	31°51'51.7"N	116°36'58.9"W	Arroyo Ensenada	61.70	16.7	8	3.2	Poco Vulnerable
4	Calle Novena	31°52'7.30"N	116°36'54.3"W	Arroyo Ensenada	43.30	17.6	3	4.5	Moderadamente Vulnerable
5	Calle Once (Jorobado)	31°56'16.6"N	116°36'52.9"W	Arroyo Ensenada	50.00	20	2	4.1	Poco Vulnerable
6	Alisos	31°52'40.9"N	116°36'44.4"W	Arroyo Ensenada	43.00	9.5	3	2.8	Poco Vulnerable
7	Av. Ámbar	31°52'57.7"N	116°36'38"W	Arroyo Ensenada	18.00	21.00	1	3.1	Poco Vulnerable
8	Av. Reforma (Bonito)	31°53'0.3"N	116°36'37.8"W	Arroyo Ensenada	24.50	10	5	4.7	Moderadamente vulnerable
9	Av. Espinoza	31°52'9.3"N	116°36'47.2"W	Arroyo El Aguajito	20.60	10	1	3.5	Poco Vulnerable
10	Calle Once (Xochicalco)	31°52'8.1"N	116°36'36.1"W	Arroyo Ensenada	18.10	16.8	1	3.6	Poco Vulnerable
11	Av. Reforma (Misión)	31°52'19.30"N	116°36'29.4"W	Arroyo Ensenada	28.10	19.8	1	3.3	Poco Vulnerable
12	I. Allende	31°52'30.5"N	116°35'48.2"W	Arroyo Doña Petra	14.00	8.4	3	3.8	Poco Vulnerable
13	Calle Bronce	31°52'33.8"N	116°35'35.2"W	Arroyo Doña Petra	15.00	10.7	1	4.1	Moderadamente Vulnerable
14	Av. Riveroll	31°52'49.9"N	116°36'50.7"W	Arroyo Doña Petra	21.60	14.5	1	3.3	Poco Vulnerable
15	Calle Agaves	31°53'8.3"N	116°36'57.8"W	Arroyo El Gallo	8.30	8.7	1	3.3	Poco Vulnerable
16	Calle Puebla	31°53'16.9"N	116°37'1.1"W	Arroyo El Gallo	11.70	14.6	1	3.8	Poco Vulnerable
17	Puente Nuevo	31°50'49.7"N	116°34'21.5"W	Arroyo El Gallo	21.80	2 tramos 12.30 m c/u	1	2.8	Poco Vulnerable
18	Av. Pedro Loyola	31°50'46.00"N	116°36'28.2"W	Arroyo El Gallo	19.25	23.5	1	3.1	Poco Vulnerable
19	Av. Reforma (Libramiento)	31°50'45.8"N	116°36'10.8"W	Arroyo El Gallo	62.65	26	5	3.7	Poco Vulnerable
20	Av. México	31°50'43.20"N	116°35'24.3"W	Arroyo El Gallo	18.00	18.5	1	3.3	Poco Vulnerable
21	Acceso A Villa Bonita	31°50'49.2"N	116°34'21.8"W	Carretera Tijuana-Ensenada	32.00	10.6	1	2.7	Poco Vulnerable
22	Acceso a Ensenada por el Recinto Portuario	31°59'29.10"N	116°38'7.4"W	Carretera Tijuana-Ensenada	30.50	36	1	3.8	Poco Vulnerable
23	Marina Coral	31°52'0.2"N	116°38'7.4"W	Carretera Tijuana-Ensenada	30.00	14	1	3.8	Poco Vulnerable
24	Acceso a UABC	31°52'59.9"N	116°40'1.7"W	Carretera Tijuana-Ensenada	32.00	13	2	3.7	Poco Vulnerable
25	Acceso a Ensenada de Tecate	31°53'55.00"N	116°42'17.7"W	Carretera Tijuana-Ensenada	28.15	9	2	4	Moderadamente Vulnerable
26	Acceso a Cíbulas del Mar de Tijuana	31°54'13.13"N	116°43'21.00"W	Carretera Tijuana-Ensenada	25.70	22.6	3	2.8	Poco Vulnerable
27	Acceso a la carretera libre de Tijuana	31°54'13.00"N	116°43'39.8"W	Carretera Tijuana-Ensenada	25.80	9	2	3	Poco Vulnerable
28	San Miguel	-	-	No identificado	82.00	10	4	3.9	Poco Vulnerable
29	Carretera Transpeninsular (San Carlos)	31°46'16.5"N	116°35'17.00"W	No identificado	38.00	17.5	3	3.8	Poco Vulnerable

Tabla 6.1. Datos de los puentes urbanos de la ciudad de Ensenada y resultados del estudio de vulnerabilidad

Fuente: Espinoza et al. (2008)

El número total de capas de datos espaciales creadas fueron seis, tres fueron utilizadas para el mapa base: ciudad de Ensenada (Ver Figura 6.3), delegación municipal de El Sauzal (Ver Figura 6.4) y delegación municipal de Chapultepec (Ver Figura 6.5); el resto de las capas se destinaron para los puentes de cada una de las localidades mencionadas. Las capas de datos espaciales de puentes, se baso en la representación por puntos, agregando manualmente la información de la (Ver tabla 6.1) en la base de datos DBF de cada uno de los puntos representados, este procedimiento se realizó con la ayuda de la interfaz de edición de DBF que proporciona ArcView 3.2 (Ver Figura 6.6).

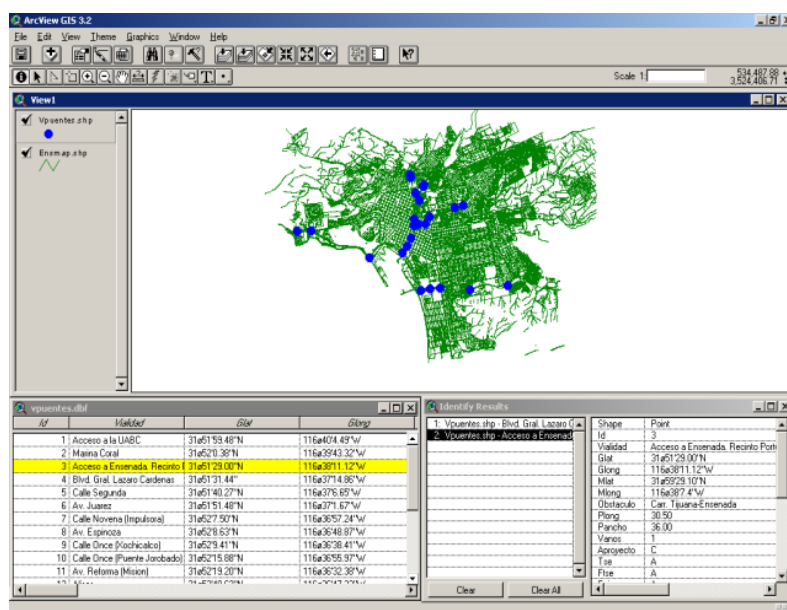


Figura 6.3 Integración de capas de la ciudad de Ensenada, BC y puentes.

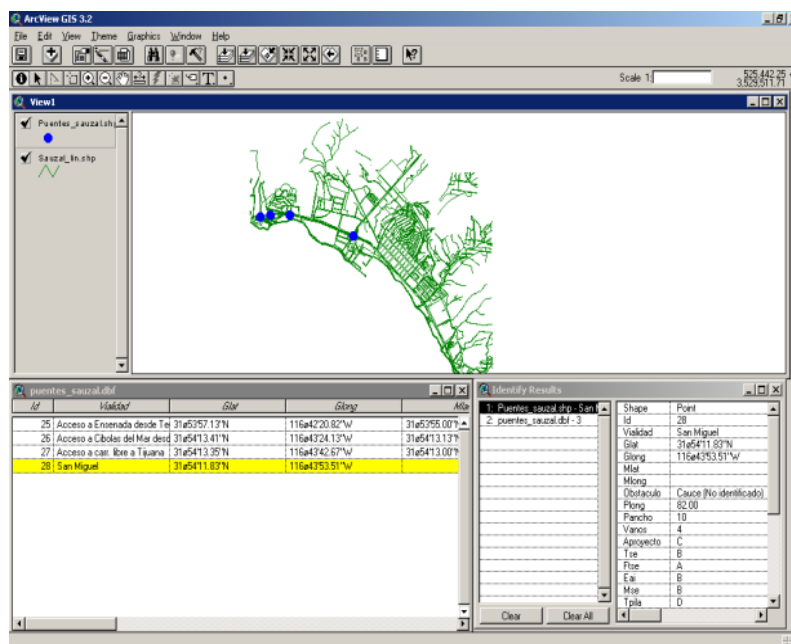


Figura 6.4 Integración de capas de la delegación municipal de El Sauzal en Ensenada, BC y puentes.

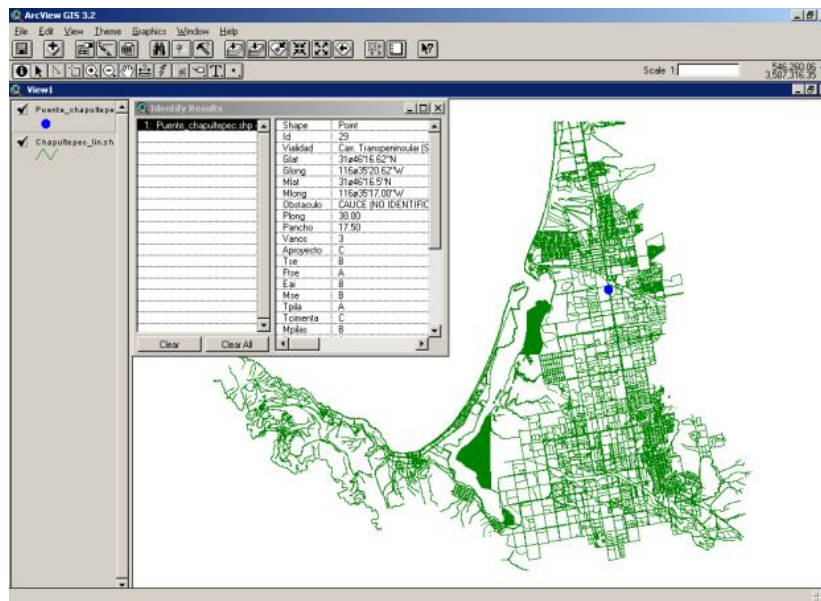


Figura 6.5 Integración de capas de la delegación municipal de Chapultepec en Ensenada, BC., y puente

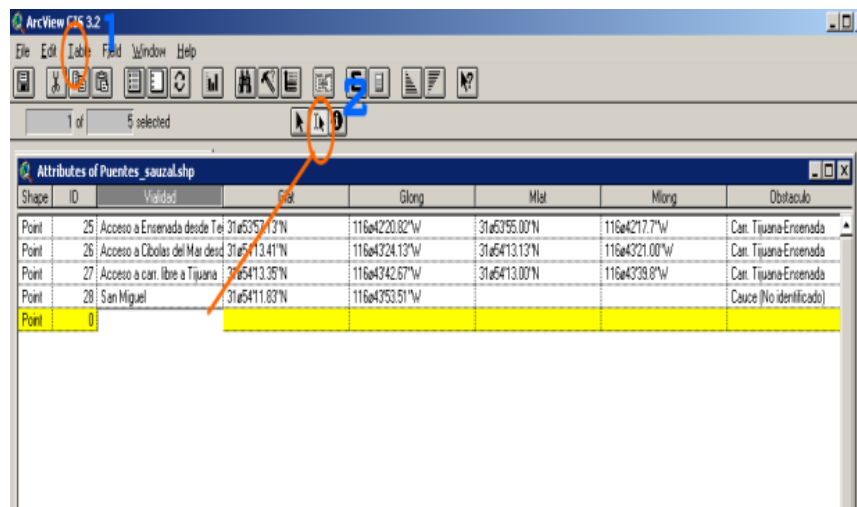


Figura 6.6 Interfaz de edición de tablas DBF. 1.- Habilitar opción de edición de tablas. 2.- Introducir datos en editor

6.2 Distribución del contenido

1. Herramientas. Existen 2 secciones para la selección de herramientas en la interfaz Web.
 - Parte superior del mapa-iconos. De izquierda a derecha: *tipo de medición*.- sirve cambiar el sistema de medición de distancias Imperial o Métrico decimal; *consulta*.-permite obtener información sobre las capas de datos; *mano*.-mueve el mapa; *lupa*.-acercamiento del mapa por zona seleccionada; *signo (+)*.- acercamiento de general del mapa; *signo (-)*.- alejar mapa; *mundo con lupa*.- restaura mapa inicial; *flecha izquierda*.- regresa al mapa visualizado previamente; *flecha derecha*.- permite visualizar de nueva cuenta un mapa; *regla*.-mide distancias entre dos puntos (Ver Figura 3.32).
 - Sobre el mapa. Con el botón derecho del ratón se despliega un menú el cual permite acceder de forma más rápida a las mismas herramientas que se hizo mención en el párrafo anterior (Ver Figura 3.32).
2. Selección de capas de datos espaciales. Se ubica en la parte superior izquierda un listado con las capas de datos espaciales con su nombre y una caja de verificación para habilitar o deshabitar las capas (Ver Figura 3.32).
3. Área de visualización de resultados.- Existen 2 secciones de visualización de resultados en la interfaz Web (Ver Figura 3.32).
 - Lado izquierdo. En la pestaña de *Resultados* se despliegan los resultados de la(s) consulta(s) realizadas a las capas de datos, por ejemplo: nombre de la vialidad en la que se sitúa el puente, imagen, coordenadas geográficas, ancho y largo, observaciones, entre otras. En el caso de la pestaña de *Mapa* se genera una imagen de referencia que permite orientar la ubicación actual en que se encuentra el mapa.
 - Centro. Se visualiza los mapas base y los puentes urbanos.

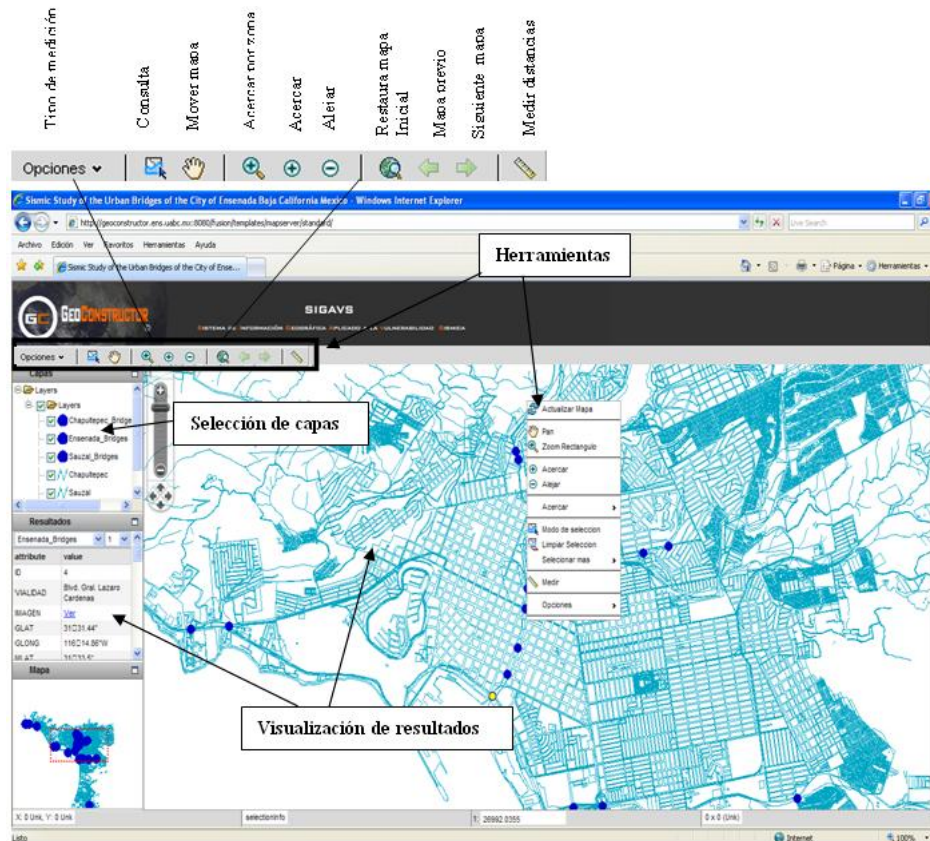


Figura 6.7. Distribución de las secciones de la aplicación espacial

6.3 Funcionamiento general para la consulta de datos

Al iniciar la aplicación espacial, carga de manera predeterminada las 6 capas de datos espaciales una vez cargadas las capas de datos espaciales en la aplicación espacial se puede obtener información de los puentes con la ayuda del icono consulta; el procedimiento es el siguiente: se presiona una vez sobre el icono consulta y con el cursor del ratón se realiza una selección múltiple o sencilla de los puntos representados sobre el mapa; automáticamente se generarán las consultas y será posible visualizar la información contenida en cada uno de los puentes urbanos en el lado izquierdo de la pantalla (Ver Figura 6.8).

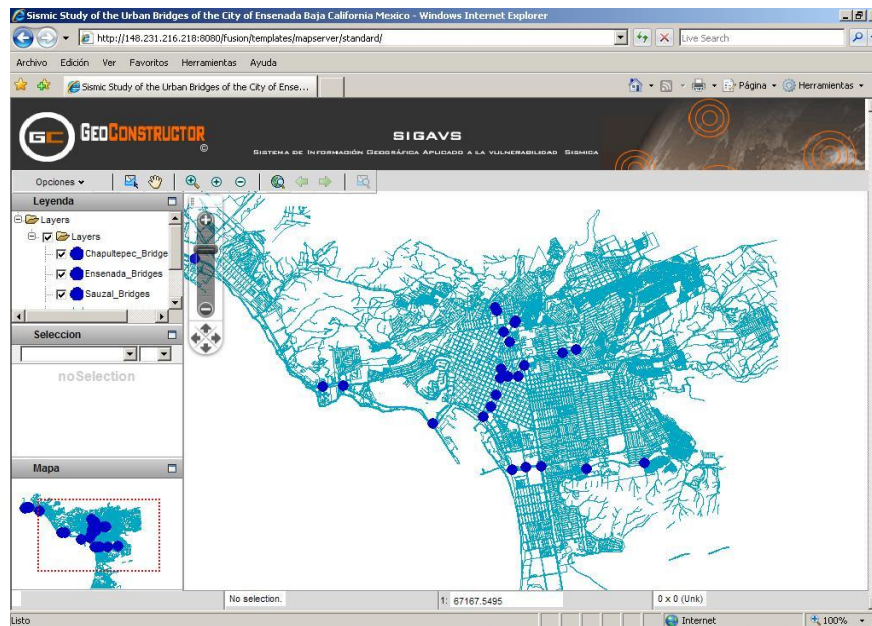


Figura 6.8. Resultado de la aplicación espacial

6.4 Catálogo de imágenes

Con el objetivo de entender con mayor claridad el tipo y el estado físico del puente urbano que se consulta en la aplicación cartográfica, se integró un catálogo de imágenes por cada puente. El catálogo es visualizado hasta que se consulta sobre el punto que representa el puente, al lado izquierdo de la pantalla aparece un enlace que permite acceder al catálogo. La implementación del catálogo solo permite visualizar imagen por imagen por lo tanto es necesario presionar sobre el botón que aparece al lado izquierdo y derecho de la imagen (Ver Figura 6.9).

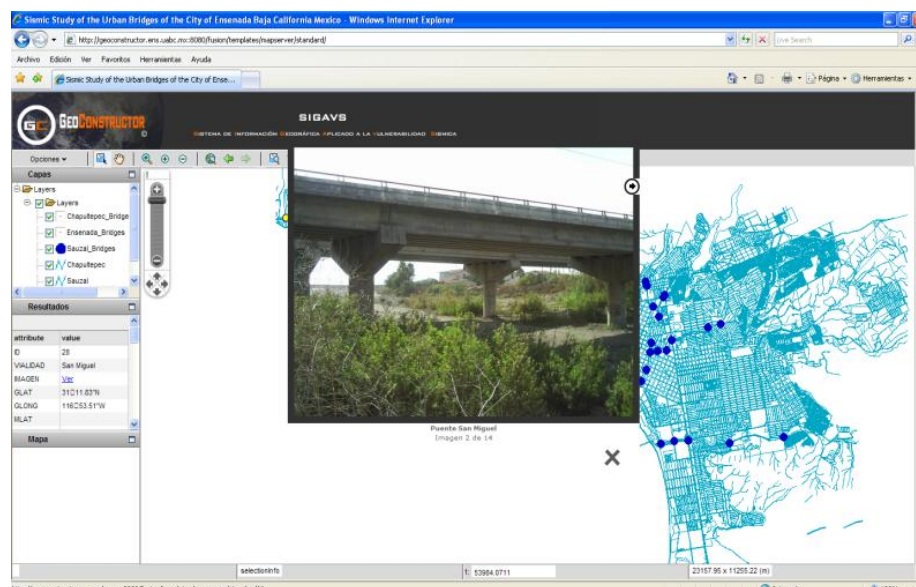


Figura 6.9. Catálogo de fotos en la aplicación espacial

Fase 2

Diseño de esquema de
selección de datos
espaciales

Fase 2.- Diseño de esquema de selección de datos espaciales

Con base en la problemática que enfrentan los servidores de mapas, objetivos planteados esta investigación, así como también características de los servidores de mapas se diseñó un esquema general que muestra el funcionamiento que debe tener el servidor de mapas UMN MapServer (Ver Figura 6.10).

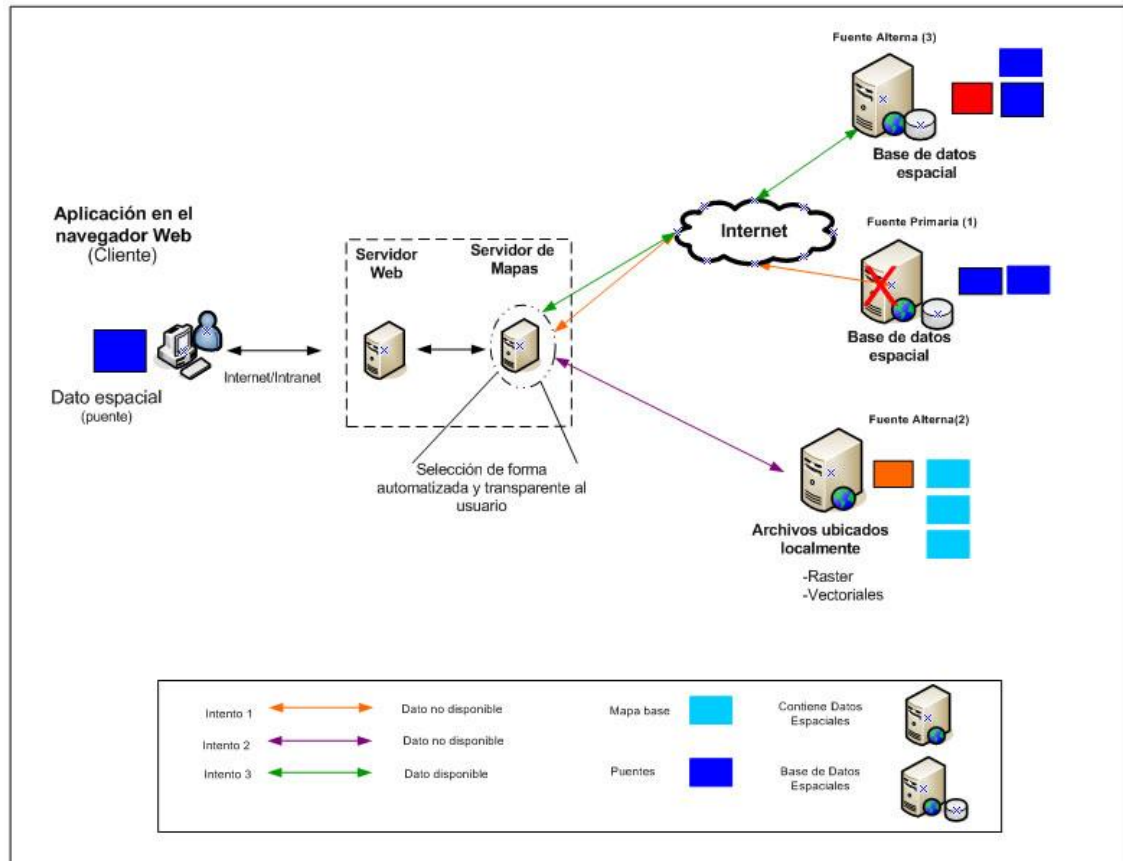


Figura 6.10 Esquema de selección de fuentes de datos espaciales

Para explicar el funcionamiento del esquema anterior se plantea un escenario de uso a continuación:

Un usuario elige el dato espacial puente desde su aplicación Web, el objetivo es solamente visualizar el dato espacial. La solicitud es enviada al servidor de mapas, el cual con base a las 3 fuentes de datos definidas en él, obtendrá el dato espacial.

Con la ayuda del método de OPTIMISTA definido en el archivo Mapfile (el método OPTIMISTA así como el PESIMISTA son métodos propuestos en los requerimientos).

El servidor de mapas selecciona la fuente primaria y solicita el dato espacial. En el primer intento, la fuente primaria no obtuvo un resultado favorable, debido a que la computadora que aloja la base de datos espacial no le sirve la fuente de poder.

Entonces se notifica en el servidor de mapas que no se pudo obtener el dato espacial y se realiza una nueva petición, pero esta vez se seleccionan de forma automática y transparente al usuario las fuentes alternas (el algoritmo de selección-método OPTIMISTA empezó su funcionamiento); en el primer intento en la fuente alterna el resultado no fue favorable, ya que el dato espacial puente fue borrado de la computadora. Por lo tanto, se notifica en el servidor de mapas que la fuente de datos alterna no está disponible.

Ahora el servidor de mapas selecciona la ultima fuente alterna definida , esta vez el dato espacial solicitado si está disponible en la base de datos espacial, entonces el servidor de mapas lo obtiene y envía una respuesta a la aplicación Web, visualizándose el puente. (Ver Figura 6.10).

6.5 ¿Dónde y cómo se adaptará el esquema de selección de fuentes de datos espaciales propuesto?

De acuerdo a la arquitectura que posee el servidor de mapas UMN MapServer, se consideró que una parte vital para adaptar el esquema de selección de fuentes de datos (Ver figura 6.10) es el archivo Mapfile, como comenta Jeff McKenna el Mapfile es el corazón del servidor de mapas UMN MapServer [Mckenna, 2009]. Una vez que se identificó de forma general la adaptación del esquema, fue necesario conocer de forma específica el lugar dentro del archivo Mapfile. Para llevar a cabo lo anterior se partió de dos requerimientos básicos que están intrínsecamente ligados al esquema:

- **Requerimiento 3: Fuentes alternas**
 1. El algoritmo debe permitir la declaración de varias fuentes alternas para obtener datos espaciales.
- **Requerimiento 4: Métodos**
 1. El algoritmo contará con dos métodos para obtener las fuentes de datos, el método OPTIMISTA y el método PESIMISTA.
 2. La persona encargada del servidor de mapas podrá elegir alguno de los métodos a usar.
 3. El método PESIMISTA debe contar con una opción para indicar el número de veces que tiene que recorrerá todas las fuentes.

A partir de los anteriores requerimientos se identifico a la etiqueta objeto LAYER como el lugar específico para la adaptación del esquema (Ver Figura 6.11). El motivo de implementar el esquema de selección de fuentes de datos en la etiqueta objeto LAYER, se debe a que dentro de esta etiqueta se definen las fuentes de datos, por lo tanto elegir otra ubicación sería incongruente a la estructura del Mapfile.

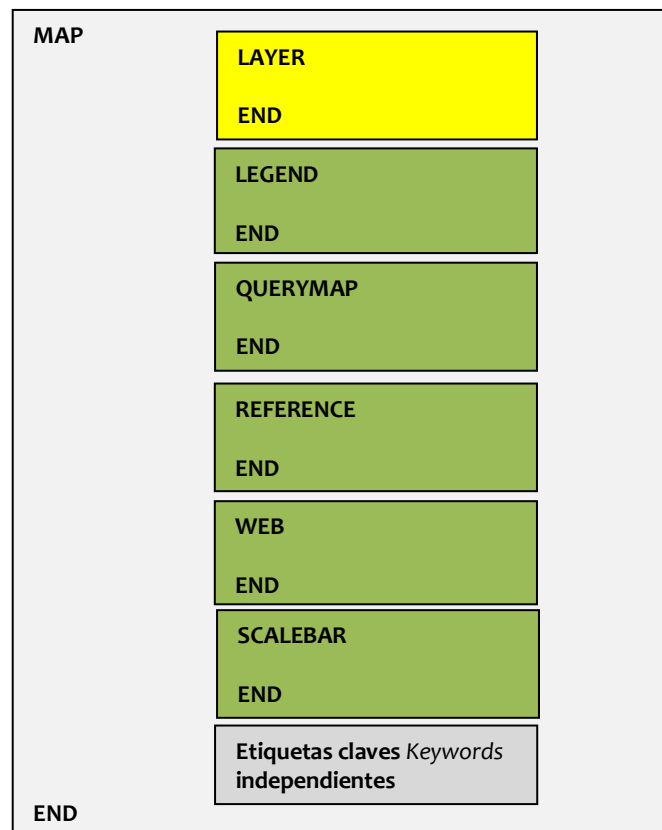


Figura 6.11. Identificación del lugar específico (cuadro amarillo) para implementar el esquema de selección de fuentes de datos espaciales

La actividad que prosiguió a la identificación general y específica del lugar de implementación del esquema, fue la interpretación de los requerimientos (mencionados anteriormente) al lenguaje del archivo Mapfile, a continuación se muestran las figuras que detallan solamente la traducción de las etiquetas propuestas:

6.6 LAYER: Nuevas etiquetas claves independientes

La etiqueta objeto LAYER se le asignarán dos etiquetas claves independientes nuevas: ACESSTYPE y ATTEMPT (Ver Figura 6.12).

1. Etiqueta clave independiente ACESSTYPE

En ACESSTYPE se define el tipo de acceso a las fuentes de datos alternas método: **OPTIMISTIC o PESSIMISTIC.**

OPTIMISTIC: Este método intentará obtener el dato solo una vez por cada una de las fuentes alternas, si no encuentra el dato en ninguna, notificara con mensaje de error.

Es decir si existen 3 fuentes alternas declaradas: intenta una vez en la fuente uno, una vez en la fuente dos y una vez en la fuente tres. Si en ninguna de esas fuentes obtuvo el dato, debe mostrar un mensaje de error en un archivo de texto.

PESSIMISTIC: Este método intentará obtener el dato de 1 a 3 veces (ATTEMPT) por fuente; es decir si existen 3 fuentes declaradas y ATTEMPT=1: Significa que si en el primer recorrido no encuentra el dato en ninguna de las tres fuentes, regresara a la primera fuente e intentara otra vez, si no encuentra el dato va a la segunda fuente y así sucesivamente. Si en ninguno de los recorridos de las fuentes alternas obtuvo el dato, debe mostrar un mensaje de error en un archivo de texto.

2. Etiqueta clave independiente ATTEMPT

La etiqueta ATTEMPT indica el número de recorridos que se harán en todas las fuentes alternas. Los valores que se pueden definir están en el rango de [1-3] y ATTEMPT solo aplica para el método PESSIMISTIC.

Ejemplo: Si ATTEMPT 2. Intentará 2 veces todas las fuentes, pero no de manera inmediata, si no que intentara una vez en la fuente uno y se pasa a la siguiente fuente, cuando termine de recorrer todas las fuentes, entonces regresara a la fuente uno e intentara otra vez, hasta que el contador de ATTEMPT llegue a dos.

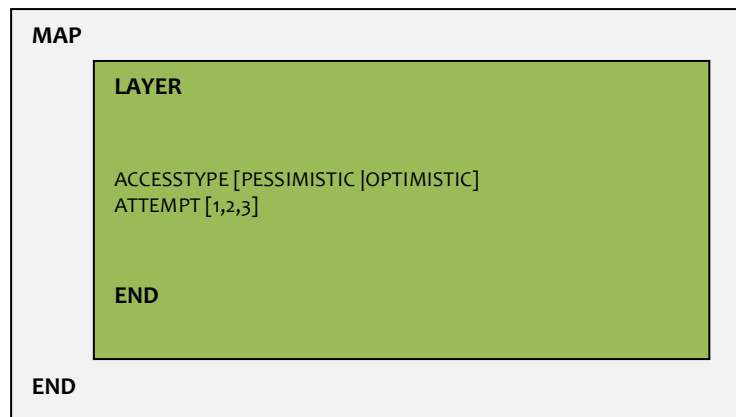


Figura 6.12. Creación de etiquetas claves independientes para la etiqueta objeto LAYER

6.7 Crear etiqueta objeto ALTERNATIVE

La etiqueta objeto ALTERNATIVE pertenecerá a la etiqueta objeto LAYER (Ver Figura 6.13) Permite identificar las fuentes alternas. Esta etiqueta clave engloba a las etiquetas claves independientes NAME, DATA, CONNECTION y CONNECTIONTYPE.

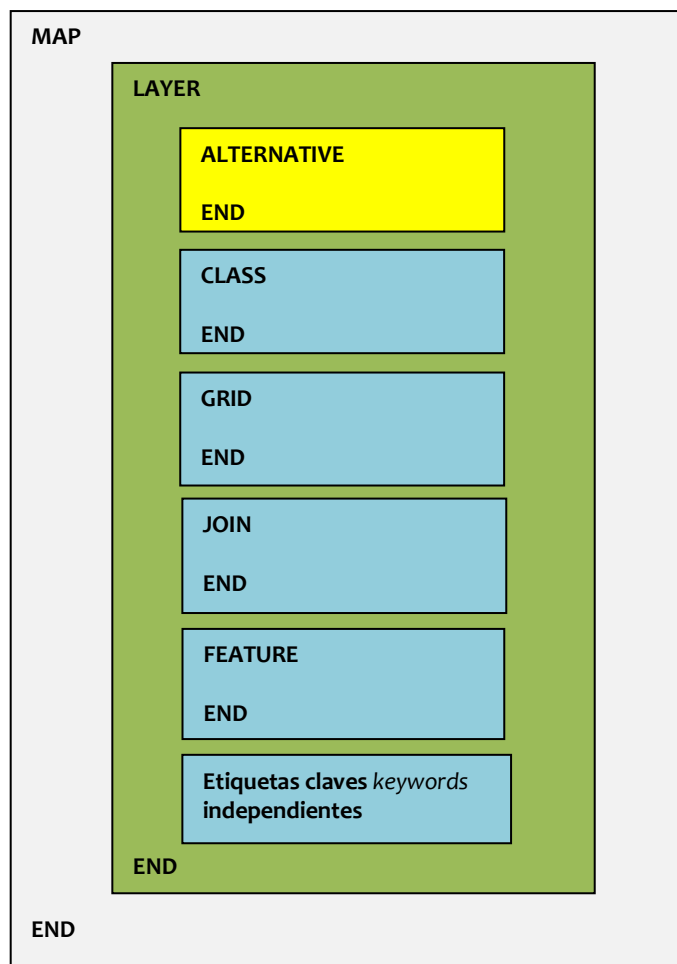


Figura 6.13. Creación de etiqueta objeto ALTERNATIVE

La etiqueta ALTERNATIVE permite declarar varias fuentes de datos alternas. Para llevar a cabo esto, es necesario escribir la palabra **ALTERNATIVE** y dentro de esta la fuente de datos remota o local, finalizando con un **END** por cada **ALTERNATIVE** declarado (Ver figura 6.14).

La etiqueta **ALTERNATIVE** contendrá a las etiquetas claves independientes: **NAME**, **DATA**, **CONNECTIONTYPE** y **CONNECTION**, estas cuatro etiquetas no son necesario crearlas, debido a que el servidor de mapas UMN MapServer ya las tiene implementadas, solamente se adaptaran al funcionamiento del esquema propuesto en este proyecto de investigación.

A continuación se realiza la descripción de las tres etiquetas claves independientes:

1. **NAME:** Indica el nombre de la fuente de datos alterna.
2. **DATA:** Indica la forma en que se consultará una fuente de datos. Contiene los siguientes parámetros: [nombre_del_dato_espacial], [tabla/columna postgis], los cuales dependen del CONNECTIONTYPE que se realice.

Si DATA utiliza solamente [nombre_del _dato_espacial]: indica que la fuente de datos es local, por lo tanto no es necesario indicar el CONNECTION Y CONNECTIONTYPE, ni tampoco la extensión del dato espacial (puente.SHP). Ejemplo:

DATA PUENTE

Sí DATA utiliza [tabla/columna POSTGIS]: indica que las fuentes de datos pueden ser remotas o locales, es necesario indicar el CONNECTION Y CONNECTIONTYPE. Ejemplo para una base de datos en POSTGIS:

DATA "geom from rio"

3. **CONNECTIONTYPE:** Indica el tipo de conexión que se realizara. Los parámetros que contiene son: [local|postgis]. Ejemplo:

CONNECTIONTYPE POSTGIS

4. **CONNECTION:** Indica la cadena de conexión para cada CONNECTIONTYPE que se elija. Por ejemplo para una base de datos POSTGIS:

CONNECTION "user=db dbname=database host=server"

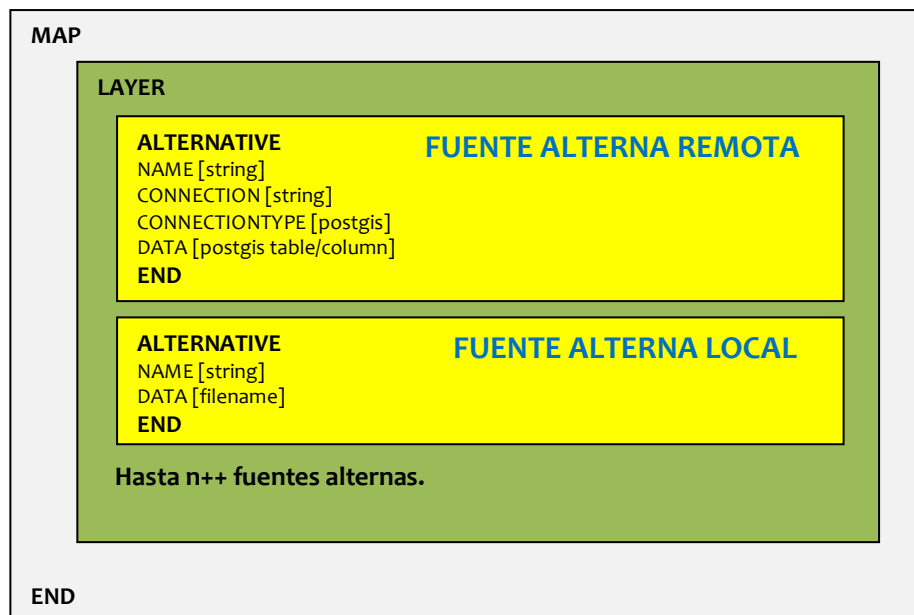


Figura 6.14. Declaración múltiple de fuentes alternas ALTERNATIVE

6.8 Diseño específico esquema de selección de datos espaciales: Resultado final de la interpretación de requerimientos

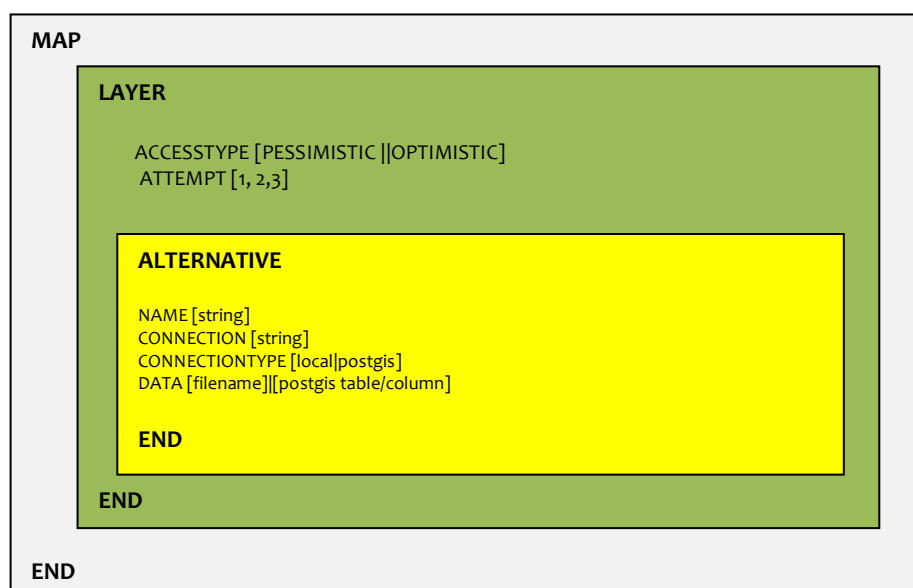


Figura 6.15. Esquema adaptado a la estructura interna del Mapfile

6.9 Identificación de fuente primaria y fuente alterna en el Mapfile

El servidor de mapas UMN MapServer permite declarar varios objetos LAYER y cada objeto LAYER tiene una sola fuente de datos, esta se conocerá para este proyecto de investigación como FUENTE PRIMARIA y la propuesta del objeto ALTERNATIVE como FUENTES ALTERNAS (Ver Figura 6.16).

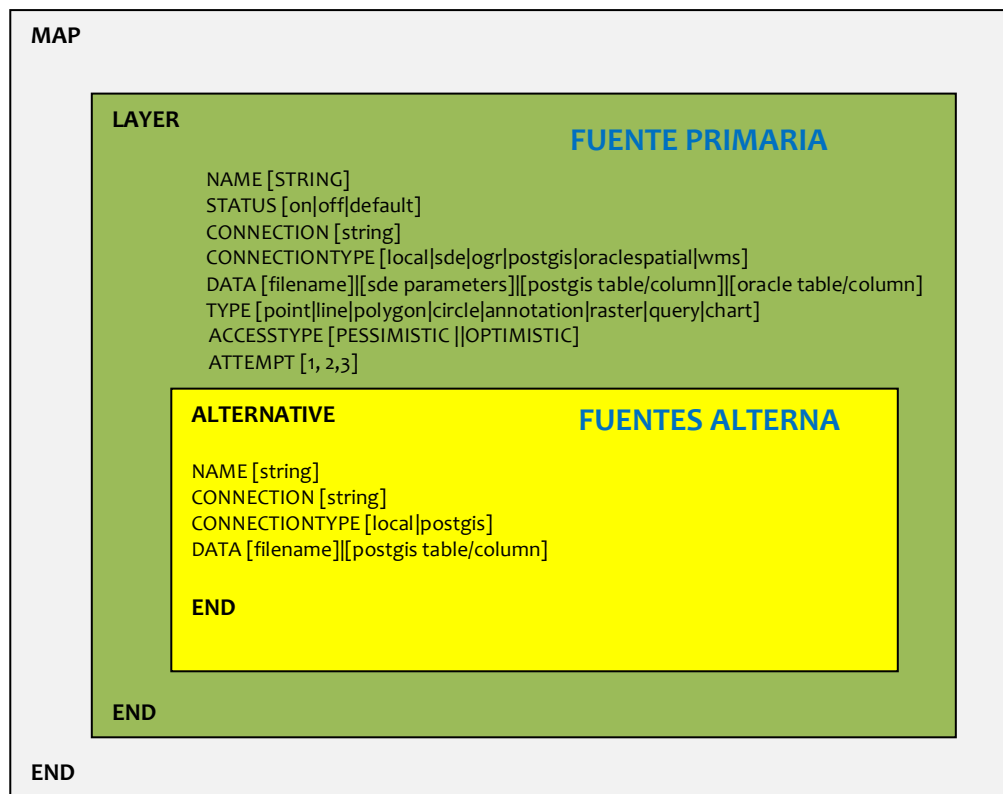


Figura 6.16. Identificación de fuente primaria y alterna en el Mapfile

Fase 3

Desarrollo de algoritmo
de selección de datos
espaciales

Fase 3: Desarrollo de algoritmo de selección de datos espaciales

Para llevar a cabo el proceso de implementación del esquema de selección de datos espaciales (Ver Figura 6.16) para el servidor de mapas UMN MapServer. Fue necesario efectuar tres actividades:

1. Realización de pruebas a diversos entornos de desarrollo (IDE) para seleccionar el entorno de desarrollo indicado para la programación, compilación y depuración de código fuente y para conocer el funcionamiento interno del servidor de mapas.
2. Interpretación del esquema general y específico a pseudocódigo
3. Identificación de los archivos fuentes del servidor de mapas a modificar y la forma en cómo se implementó el esquema de selección datos espaciales.

6.10 Pruebas a entornos de desarrollo (IDE)

Uno de los retos más importantes debido al tiempo dedicado y las adversidades enfrentadas en el presente proyecto de investigación fue encontrar un entorno de desarrollo (IDE) indicado para el desarrollo interno del servidor de mapas UMN MapServer. Esto se debe a dos factores principales:

1. La limitada y esporádica ayuda por parte de la comunidad de desarrollo de UMN MapServer (<http://mapserver.org/community/lists.html#mapserver-dev>) y foros relacionados con tecnología SIG en Internet.
2. La escasa y confusa documentación técnica encontrada en el sitio oficial del servidor de mapas UMN MapServer (<http://mapserver.org>), libros e internet.

La comunidad de desarrollo de UMN MapServer la conforma un equipo de trabajo no dedicado exclusivamente a este proyecto, por lo que resulta complicado obtener respuesta inmediata a dudas relacionadas con la implementación o funcionamiento de código en UMN MapServer. Por otro lado cabe señalar que si existió suficiente apoyo por parte de la comunidad de usuarios de UMN MapServer (<http://mapserver.org/community/lists.html#mapserver-users>) y foros relacionados con tecnología SIG, para resolver dudas relacionadas al uso de MapServer como: modo instalación, funcionamiento de las etiquetas objeto y etiquetas claves independientes del Mapfile (.MAP), creación de aplicaciones espaciales, entre otras.

FASE 3: DESARROLLO DE ALGORITMO DE SELECCIÓN DE DATOS ESPACIALES

La escasa documentación técnica del código fuente y de entornos de desarrollo para UMN MapServer, provocó que fuera una tarea ardua el descubrir cómo implementar el algoritmo de selección para el servidor de mapas UMN MapServer, debido al tiempo dedicado en investigación y pruebas a diferentes entornos de desarrollo para la programación, depuración y compilación del código fuente (Ver Tabla 6.2).

Entorno de desarrollo	Descripción
Visual Studio 2005	Permite la programación, depuración y compilación en un entorno gráfico. La compilación de código se realiza vía línea de comandos, con la emulación de librerías de Linux en Windows (Cygwin). La página oficial de UMN MapServer proporciona escasa información de cómo compilar MapServer en Windows y el modo de hacerlo es complejo, no se logro hacer funcionar Visual Studio 2005 para programar, depurar y compilar código de UMN MapServer. Lo anterior se debe a que no existe suficiente documentación que facilite su desarrollo o que ayude a responder a la pregunta ¿Como implementar código en UMN MapServer?
Netbeans 6.7	Permite la programación, depuración, y compilación en un entorno grafico, sin embargo para llevar a cabo estas actividades en Windows necesita emular librerías de Linux con Cygwin. No se logro hacer funcionar Netbeans 6.7 para programar, depurar y compilar código de UMN MapServer. Para el IDE Netbeans tanto en Linux como en Windows no existe suficiente documentación para responder a la pregunta ¿Como implementar código en UMN MapServer?
Dev C 5	Permite la programación, depuración y compilación en un entorno grafico, sin embargo para llevar a cabo estas actividades necesita emular librerías de Linux en Windows con Cygwin. No se logro hacer funcionar Dev c 5 para programar, depurar y compilar código de UMN MapServer. No existe suficiente documentación para responder a la pregunta ¿Como implementar código en UMN MapServer?
Turbo C	No se logro hacer funcionar Turbo C para programar, depurar y compilar código de UMN MapServer, debido a que Turbo C requiere de un depurador especial llamado GDB. Este depurador puede obtenerse a partir de las librerías Cygwin. Sin embargo no se logro hacer funcionar Turbo C para UMN MapServer. Además no se encontró documentación que ayudara a responder el cuestionamiento ¿Como implementar código en UMN MapServer?
KDbg2.0.2	Solo permite la depuración de código de UMN MapServer. Este programa solo es compatible en Linux
DDD 3.3.1	Solo permite la depuración de código de UMN MapServer. Este programa solo es compatible en Linux.

Tabla 6.2. Resultados de pruebas realizadas a entornos de desarrollo para la programación, depuración y compilación del código fuente de UMN Mapserver.

De acuerdo a la tabla mostrada anteriormente, se puede concluir que los entornos de desarrollo: Visual studio 2005, Netbeans 6.7, Dev C 5 y Turbo C, son complejos de adaptar para la programación,

depuración y compilación de código fuente de UMN MapServer, aunado a la escasa documentación técnica que existe de cada uno de estos entornos en relación al desarrollo de código para UMN MapServer, por lo tanto ninguno de los entornos antes mencionados fueron seleccionados.

A consecuencia de que ninguno de los entornos de desarrollo funcionó para la programación, compilación y depuración de código fuente, se decidió realizar pruebas independientes, es decir buscar programas capaces de realizar solo una actividad: solo la compilación, solo la programación y solo depuración.

Para el caso de la depuración de código fuente de UMN MapServer se realizaron pruebas con DDD 3.3.1 y KDbg 2.0.2, satisfactoriamente ambos pudieron depurar el código fuente, así que cualquiera de estos programas son una buena opción para depurar el código fuente de UMN MapServer, en el caso de este proyecto de investigación se selecciono KDbg 2.0.2. Tanto DDD 3.3.1 como KDbg 2.0.2 proporcionan una interfaz gráfica amigable para introducir valores de prueba, funciones para agregar puntos de ruptura (breakpoints) y observar los valores internos de las variables (watch), lo cual permite conocer el funcionamiento interno de UMN MapServer.

Para la compilación de código fuente se realizó por línea de comandos también conocida como intérprete de comandos, Shell, consola o terminal comúnmente utilizada para llevar a cabo la instalación y compilación de librerías de UMN MapServer en Linux.

Finalmente para la programación de código se utilizó el editor de textos NANO, este editor se invoca por línea de comandos y también es extensamente utilizado en Linux.

Para conocer más acerca del desarrollo independiente utilizando: el depurador KDbg 2.0.2, la línea de comandos y el editor NANO consultar el manual de desarrollo que se elaboró para el presente proyecto de investigación (Ver anexo Manual de desarrollo de UMN MapServer).

6.11 Interpretación del esquema general y específico a pseudocódigo

Realizar una interpretación del esquema general y específico a pseudocódigo tiene como finalidad contribuir en dos aspectos: 1.-Una fácil implementación del pseudocódigo a lenguaje formal y 2.- Comprender de forma clara la solución que se plantea en los esquemas antes mencionados.

```

If servidoresolicita then {
    Seleccionar fuente primaria y alternas
    if datoobtenido then
        Mostrar resultado en aplicación espacial
    else
        Contar número fuentes alternas
        If numerofuente==0 then
            Mostrar mensaje de error no hay fuentes alternas declaradas break;
        Else {
            Switch ACESSTYPE
            {Case OPTIMISTIC:
                contadorfuentes=0; i=0; obtenido=0;
                While contadorfuentes<=numerofuente {
                    Seleccionar fuentealterna [i]
                    if datoobtenido then {
                        Mostrar resultado en aplicación espacial
                        obtenido=1;
                    }
                    else {
                        i++;
                        contadorfuentes++;
                    }
                }
                If obtenido==0 then
                    Mostrar mensaje de error "ninguna de las fuentes alternas tiene el dato"
                    Exit(0)
                Break;
            Case PESSIMISTIC:
                ca=1; // ca= contador de intentos
                while ca<=ATTEMPT
                    contadorfuentes=0; i=0;
                    While contadorfuentes<numerofuente {
                        Seleccionar fuentealterna [i]
                        if datoobtenido then {
                            Mostrar resultado en aplicación espacial
                            obtenido=1;
                        }
                        Else {
                            i++;
                            contadorfuentes++;
                        }
                    }
                    ca++;
                }
                If obtenido==0&&ca==ATTEMPT then
                    Mostrar mensaje de error: ninguna de las fuentes alternas tiene el dato
                    Exit (0)
            }
        }
    }
    endswitch
}
endif
else
    No se ha solicitado ningún dato espacial

```

6.12 Identificación de los archivos fuentes del servidor de mapas a modificar y forma de implementar el esquema de selección datos espaciales.

De acuerdo a la revisión de los 149 archivos fuentes del servidor de mapas UMN MapServer (<http://download.osgeo.org/mapserver/mapserver-5.4.2.tar.gz>), se lograron identificar cinco archivos fundamentales para la implementación del esquema de selección de datos espaciales. La tabla que se muestra en la parte inferior, proporciona el nombre del archivo y el tipo de modificación que se realizó (Ver Tabla 6.3)

Archivos Fuentes	Tipo modificación
mapfile.h	Asignación de <i>tokens</i> (identificador) a etiqueta objeto y etiquetas claves independientes propuestas en este proyecto.
maplexer.l	Validación léxica de etiquetas objeto y etiquetas claves independientes propuestas en este proyecto.
mapserver.h	Declaración de estructuras y variables
mapfile.c	Desarrollo de funciones para el funcionamiento del esquema y validaciones.
mapdraw.c	Desarrollo de código para la selección automatizada y transparente de las fuentes de datos alternas

Tabla 6.3 Archivos fuentes modificados del servidor de mapas UMN MapServer

A continuación se detalla las modificaciones realizadas a cada uno de los archivos fuentes del servidor de mapas:

- **mapfile.h**

```
#define ALTERNATIVE 1230
#define ACCESTYPE 1231
#define ATTEMPT 1232
```

- **maplexer.l**

La forma de llevar a cabo la validación en el archivo maplexer.l es por medio de reglas, en este caso se utilizaron las siguientes: <INITIAL>, <URL_VARIABLE> y <URL_STRING>. Estas reglas indican que solamente se activara el funcionamiento y validación de las etiquetas si recibe una variable o cadena vía URL, en el caso de <INITIAL> sirve para empezar el funcionamiento o validación, es por eso que todas las etiquetas deben tener como mínimo esta regla.

A continuación se muestra la forma en cómo se asignaron las validaciones para las etiquetas propuestas:

```
<INITIAL>accesstype {return (ACCESSTYPE);}
<INITIAL, URL_VARIABLE>alternative {return (ALTERNATIVE);}
<INITIAL>attempt {return (ATTEMPT);}
<INITIAL, URL_STRING>optimistic {return (MS_OPTIMISTIC);}
<INITIAL, URL_STRING>pessimistic {return (MS_PESSIMISTIC);}
```

- **mapserver.h**

```
#define MS_ALTERNATIVE_ALLOCSIZE 8
enum MS_ACCESSTYPE {MS_OPTIMISTIC,MS_PESSIMISTIC};
```

A continuación la definición de la estructura para la etiqueta objeto ALTERNATIVE y declaración de etiquetas claves independientes en la estructura de la etiqueta objeto LAYER.

```
typedef struct alternative_obj {
    char *name; //nombre único para identificar a fuente alterna
    struct layer_obj *layer;
    char *data; // variable para asignar un dato espacial
    int type; // tipo de dato espacial
    char *connection; // para asignar valores de conexión usuario, clave,etc
    enum MS_CONNECTION_TYPE connectiontype; //tipo de conexión local o remota
} alternativeObj;
```

```
typedef struct layer_obj {

    enum MS_ACCESSTYPE accesstype; //tipo método de acceso optimistic o pessimistic
    int attempt; // # de intentos
    int numAS; // # de fuentes alternas
    int maxAS; //# máximo de fuentes alternas
    alternativeObj **alternative;

    char *name;
    int status;
    char *data;
    char *connection;
    enum MS_CONNECTION_TYPE connectiontype;
    .
    .
    .
} layerObj;
```

- **mapfile.c**

Función	Descripción	Estado de la función
int initLayer(layerObj *layer, mapObj *map)	Inicializa valores de las etiquetas de LAYER por ejemplo: accesstype, attempt, numAS, maxAS y alternative	Modificada
int freeLayer(layerObj *layer)	Libera valores de las etiquetas de LAYER	Modificada
int freeAlternative(alternativeObj *alternative)	Libera valores de las etiquetas de ALTERNATIVE	Creada
int initAlternative (alternativeObj *alternative)	Inicializa valores de las etiquetas de ALTERNATIVE por ejemplo: name, data, connection, connectiontype	Creada
alternativeObj *msGrowLayerAlternatives(layerObj *layer)	• Controla el numero de fuentes alternas creadas • Asigna memoria dinámica	Creada
static int loadAlternative(alternativeObj *alternative, layerObj *layer)	• Función principal ALTERNATIVE, a partir de esta función se obtienen valores para: name,data, connection, connectiontype . • Manda llamar al analizador léxico para validar las etiquetas declaradas dentro de ALTERNATIVE	Creada
int loadLayer(layerObj *layer, mapObj *map)	• Función principal LAYER. A partir de esta función se manda llamar las siguientes funciones de ALTERNATIVE: int initAlternative (alternativeObj *alternative) static int loadAlternative(alternativeObj *alternative, layerObj *layer) • Se selecciona el tipo de método para obtener datos espaciales • Se obtiene el numero de intentos que se realizarán para las fuentes de datos alternas declaradas	Modificada
msDebug()	Permite generar un reporte de actividades en archivo de texto. Muestra el nombre de los datos espaciales, fuentes de datos alternas declaradas, el tipo de método de selección elegido por el administrador del servidor de mapas, reporte de errores, entre otros.	Modificada

Tabla 6.4 Funciones implementadas en el archivo mapfile.c

- mapdraw.c

Función	Descripción	Estado de la función
imageObj *msDrawMap(mapObj *map, int querymap)	Sirve para seleccionar las fuentes alternas en caso de que la fuente primaria no se encuentre disponible. Se encarga de generar el dato espacial que se visualizará en la aplicación espacial	Modificada

Tabla 6.5 Función implementa en el archivo mapdraw.c

Para llevar a cabo el desarrollo y modificación de cada una de las funciones y archivos mostrados anteriormente, se tuvieron que seguir ciertos pasos específicos que permitieron la programación, compilación y depuración del código (Ver anexo Manual de desarrollo de UMN MapServer).

Validación

7

Permite conocer si la aplicación espacial, esquemas y algoritmo de selección de datos espaciales desarrollados son correctos y cumplen con las expectativas del cliente, en el presente capítulo se llevan a cabo las pruebas de ACEPTACION y ALFA.

7.1 Validación: pruebas de ACEPTACIÓN y ALFA

Ian Sommerville comenta que una validación va más allá de comprobar si el sistema está acorde con las especificaciones planteadas por el cliente, debido a que las especificaciones pueden ser confusas o tener un resultado no esperado por el cliente [Sommerville,2002].

Para llevar a cabo la validación de especificaciones y expectativas del cliente se utilizaron dos tipos de pruebas: ACEPTACIÓN y ALFA. Las pruebas de ACEPTACIÓN permiten tener un control de validaciones respecto a la aprobación o rechazo de una especificación y/o expectativa del cliente durante el desarrollo del proyecto [Pressman, 2002].

Las pruebas ALFA valida el funcionamiento integral de la aplicación espacial y el algoritmo de selección de datos espaciales; en éstas pruebas existen dos tipos de usuarios: 1.-El cliente el cual utiliza el software o aplicación de forma natural en un entorno controlado 2.-El desarrollador actúa como observador, registrando errores y problemas de uso [Pressman, 2002].

7.2 Pruebas de ACEPTACIÓN

Especificación	¿Es acorde con lo que pide el cliente?	¿Cumple expectativas del cliente?
Aplicación Espacial		
Requerimiento 1: Formato datos espaciales	Sí	Sí
Requerimiento 2: Enfoque de los datos espaciales	Sí	Sí
Requerimiento 3: Número de capas de datos espaciales	Sí	Sí
Requerimiento 4: Almacenamiento de datos espaciales	Sí	Sí
Requerimiento 5: Funciones de interacción	Sí	Sí
Requerimiento 6: Tipo de software	Sí	Sí
Requerimiento 7. Interfaz	Sí	Sí
Requerimiento 8: Catalogo de fotos	Sí	Sí
Algoritmo		
Requerimiento 1: Análisis del funcionamiento interno del servidor de mapas	Sí	Sí
Requerimiento 2: Esquema	Sí	Sí
Requerimiento 3: Fuentes alternas	Sí	Sí
Requerimiento 4 : Métodos	Si	Si
Requerimiento 5: Informe de actividades del servidor de mapas	Sí	Sí
Usuarios		
Requerimiento 1: Proveedor de servicios (desarrollador)	Sí	Sí
Requerimiento 2: Usuarios (Clientes)	Sí	Sí
Adicionales		
Requerimiento 1: Computadoras	Sí	Sí
Requerimiento 2: Red	Sí	Sí
Requerimiento 3: Sitio Web	Sí	Sí

Tabla 7.1 Cumplimiento de expectativas y requerimientos

7.3 Pruebas ALFA

Para llevar a cabo las pruebas ALFA es necesario cumplir con la mayoría de los requerimientos planteados anteriormente (Ver Tabla 7.2), permitiendo con esto la realización de pruebas de forma integral a la aplicación espacial y el algoritmo de selección de datos espaciales.

A continuación se presentan las características bajo las cuales se realizaron las pruebas ALFA para comprobar el funcionamiento del algoritmo de selección de datos espaciales en el servidor de mapas UMN MapServer.

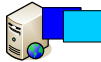
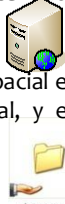
Características del entorno controlado	
Número de capas de datos espaciales a visualizar en la aplicación espacial	- Seis capas de datos espaciales - Mapa base: zona urbana de la ciudad de Ensenada, delegación municipal de El Sauzal y Chapultepec mas los puentes urbanos de cada zona respectivamente.
Los datos espaciales distribuyeron de la siguiente manera:	
Computadora 148.231.216.133 ✓ Funciona como SERVIDOR DE MAPAS (algoritmo) y fuente de datos. ✓ Sistema Operativo Linux CentOS 5	Contiene cinco datos espaciales: <u>tres capas</u> de datos espaciales del <u>mapa base</u>, <u>una</u> de los <u>puentes urbanos</u> de la delegación municipal de <u>El Sauzal</u> y un dato espacial <u>puentes urbanos</u> de la delegación de <u>Chapultepec</u> 
Computadora 148.231.216.143. ✓ Funciona como servidor de base de datos PostgreSQL/PostGIS y tiene compartido un dato espacial en red. ✓ Sistema Operativo Windows XP SP2	- Contiene dos datos espaciales, el primero es de los puentes urbanos de Ensenada el cual se encuentra almacenado en una base de datos PostgreSQL. - El segundo dato espacial es de la delegación municipal del Sauzal, y este se encuentra compartido en red. 
Computadora 148.231.216.145 ✓ Funciona como servidor de base de datos PostgreSQL/PostGIS y es una fuentes alterna. ✓ Sistema Operativo Linux CentOS5	Contiene cinco datos espaciales: las tres capas de datos espaciales de los puentes urbanos de Ensenada, Chapultepec y El Sauzal. Dos capas de datos espaciales del mapa base zona urbana de Ensenada y El Sauzal. 

Tabla 7.2 Características del entorno controlado

Respecto a la configuración del archivo MAPFILE que se utilizó es la siguiente:

MAP

```
DEBUG 5
CONFIG "MS_ERRORFILE" "/usr/local/apache2/htdocs/appgeo/tmp2/ms_error.txt"
IMAGETYPE PNG
EXTENT 529395.83783573 3519769.55896037 544885.86544958 3531024.78183222
SIZE 400 400
SHAPEPATH "datos"
IMAGECOLOR 255 255 255
```

WEB

```
TEMPLATE "/usr/local/apache2/htdocs/appgeo/index.html"
IMAGEPATH "/usr/local/apache2/htdocs/appgeo/tmp/"
IMAGEURL "/appgeo/tmp/"
LOG "/usr/local/apache2/htdocs/appgeo/tmp2/ms_status.txt"
```

END

Definición de símbolo para puntos de puentes

SYMBOL

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

END

#####Capas de datos espaciales#####

#MAPAS

#####INICIO LAYER 1#####

Dato espacial de la zona urbana de Ensenada

LAYER

NAME Ensenada

#dato espacial compartido en

#dato espacial no existe con el nombre urbanoensenadaA si no como urbanoensenada

DATA "//FAMILIA/dato_espacial/urbanoensenadaA"

STATUS ON

TYPE LINE

CLASS

NAME Ensenada

STYLE

COLOR 0 167 197

OUTLINECOLOR 218 218 218

END

END

#####Fuentes Alternas#####

ACCESSTYPE OPTIMISTIC

#dato espacial ubicado en un folder de la computadora que funge como servidor de mapas
#si existe el dato espacial

ALTERNATIVE

NAME FuenteAlterna1

DATA "/usr/datos_espaciales/ensenada"

END

#dato espacial "ubicado" en computadora 148.231.216.143

#No existe el dato espacial

ALTERNATIVE

NAME FuenteAlterna2

CONNECTIONTYPE POSTGIS

CONNECTION "host=148.231.216.143 port=5432 dbname=datos

user=postgres password=123456"

DATA "the_geom from ensenada"

END

END # Fin de layer

#####FIN LAYER 1#####

#####INICIO LAYER 2#####

Dato espacial de la delegación municipal de Chapultepec

LAYER

NAME Chapultepec

#dato espacial ubicado en un folder de la computadora que funge como servidor de mapas
#si existe el dato espacial

DATA "/usr/datos_espaciales/chapultepec"

STATUS ON

TYPE LINE

CLASS

NAME chapultepec

STYLE

COLOR 0 167 197

OUTLINECOLOR 218 218 218

END

END

#####Fuentes Alternas#####

#Esta vez para las fuentes alternas se selecciona otro método

ACCESSTYPE PESSIMISTIC

ATTEMPT 1

#dato espacial ubicado en computadora 148.231.216.145

#No existe el dato espacial con el nombre asignado en la consulta

```

ALTERNATIVE
    NAME FuenteAlterna1
    CONNECTIONTYPE POSTGIS
    CONNECTION "host=148.231.216.145 port=5432 dbname=datos
user=postgres password=123456"
    DATA "the_geom from chapul"
END

```

#dato espacial ubicado en computadora 148.231.216.145

#si existe el dato espacial con el nombre asignado en la consulta

```

ALTERNATIVE
    NAME FuenteAlterna2
    CONNECTIONTYPE POSTGIS
    CONNECTION "host=148.231.216.145 port=5432 dbname=datos
user=postgres password=123456"
    DATA "the_geom from chapultepec"
END

```

END # Fin de layer

#####FIN LAYER 2#####

#####INICIO LAYER 3#####

Dato espacial de la delegación municipal del Sauzal

#No se agrega ninguna fuente alterna, el dato si existe en la base de datos

```

LAYER
    NAME Sauzal
    CONNECTIONTYPE POSTGIS
    CONNECTION "host=148.231.216.145 port=5432 dbname=datos user=postgres
password=123456"
    DATA "the_geom from sauzal"
    STATUS ON
    TYPE LINE
    CLASS
        NAME chapultepec
    STYLE
        COLOR 0 167 197
        OUTLINECOLOR 218 218 218
    END
END
END # Fin de layer

```

#####FIN LAYER 3#####

#PUENTES URBANOS

#####INICIO LAYER 4#####

Puentes urbanos de Ensenada

#No se agrega ninguna fuente alterna, el dato si existe en la base de datos

LAYER

NAME Puentes_Ensenada

CONNECTIONTYPE POSTGIS

CONNECTION "host=148.231.216.143 port=5432 dbname=datos user=postgres
password=123456"

DATA "the_geom from puentes"

STATUS ON

TYPE POINT

CLASS

NAME pensenada

STYLE

COLOR 0 0 200

OUTLINECOLOR 0 0 200

SYMBOL 'circle'

SIZE 12

END**END**

END # Fin de layer

#####FIN LAYER 4#####

#####INICIO LAYER 5#####

Puentes urbanos de la delegación municipal del Sauzal

LAYER

NAME Puentes_ElSauzal

#dato espacial compartido en red

#No existe el dato espacial

DATA "//FAMILIA/dato_espacial/psauzal"

STATUS ON

TYPE point

CLASS

NAME psauzal

STYLE

COLOR 0 0 200

OUTLINECOLOR 0 0 200

SYMBOL 'circle'

SIZE 12

END**END**

#Se vuelve a seleccionar otro método diferente para las fuentes alternas

ACCESSTYPE OPTIMISTIC

#dato espacial ubicado en computadora 148.231.216.143 esta compartido en red
 #No existe ni el folder designado ni el dato espacial

ALTERNATIVE

NAME FuenteAlterna1
 DATA "//FAMILIA/dato_espacialES/SAUZAL"

END

#dato espacial ubicado en computadora 148.231.216.145
 #si existe el dato espacial, pero el nombre del usuario (user) no existe

ALTERNATIVE

NAME FuenteAlterna2
 CONNECTIONTYPE POSTGIS
 CONNECTION "host=148.231.216.145 port=5432 dbname=datos
 user=monge password=123456"
 DATA "the_geom from sauzalpuentes"

END

#dato espacial ubicado en computadora 148.231.216.145
 #si existe el dato espacial con el nombre asignado en la consulta

ALTERNATIVE

NAME FuenteAlterna3
 CONNECTIONTYPE POSTGIS
 CONNECTION "host=148.231.216.145 port=5432 dbname=datos
 user=postgres password=123456"
 DATA "the_geom from sauzalpuentes"

END**END # Fin de layer**

#####FIN LAYER 5#####

#####FIN LAYER 6#####

Puente urbano de Chapultepec

Caso: cuando ninguna fuente de datos tiene el dato espacial requerido en el primer intento

Uso del método pesimista

LAYER

NAME puente_chapultepec
 #dato espacial no existe en esa ruta
 DATA "/usr/local/apache2/puentechap"
 STATUS ON
 TYPE point
CLASS
 NAME pchapultepec
STYLE
 COLOR 0 0 200
 OUTLINECOLOR 0 0 200
 SYMBOL 'circle'
 SIZE 12

END
END

ACCESSTYPE PESSIMISTIC
ATTEMPT 2

#si existe el dato espacial, pero esta reiniciándose el servidor, posiblemente
Hasta la segunda vuelta con ATTEMPT obtenga el dato

ALTERNATIVE

NAME FuenteAlterna1
CONNECTIONTYPE POSTGIS
CONNECTION "host=148.231.216.145 port=5432 dbname=datos
user=postgres password=123456"
DATA "the_geom from chapultepec_puente"

END

#el nombre de la computadora no es correcto en lugar de FAM debe ir FAMILIA

ALTERNATIVE

NAME FuenteAlterna2
DATA "//FAM/datos_espaciales/pchapultepec"

END

END # Fin de layer

END # Fin del archivo map

A continuación se realiza una descripción de los resultados obtenidos con base en el entorno controlado implementado (Ver Figura 7.1)

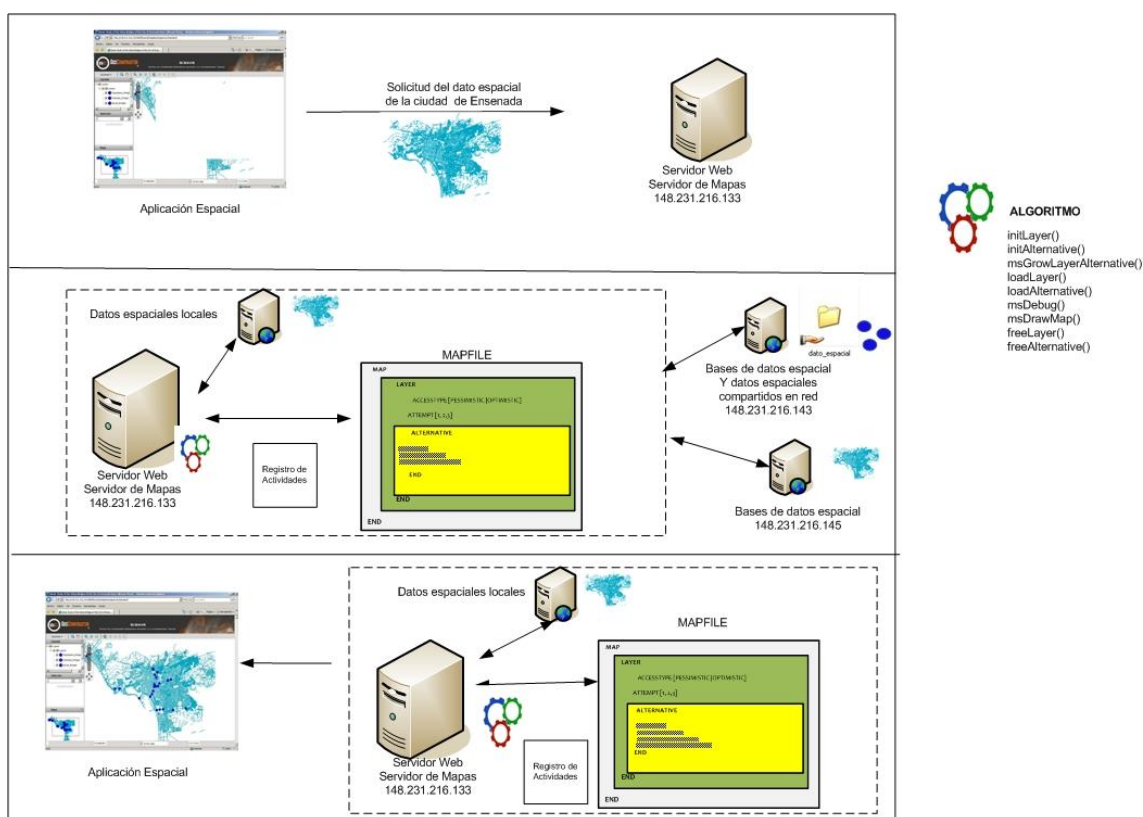


Figura 7.1. Entorno controlado para la realización de pruebas ALFA.

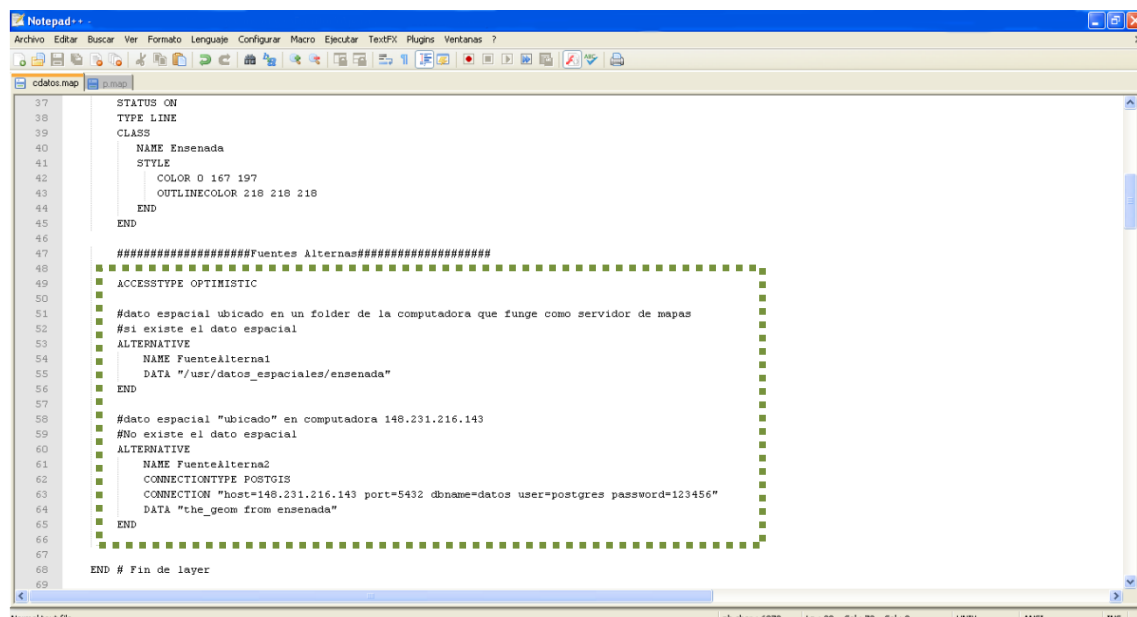
(1) Interacción aplicación espacial-- Servidor de mapas

- ✓ La aplicación espacial cumple con los requerimientos solicitados y no se encontró ningún problema en su funcionamiento: permite visualizar e interactuar con el servidor de mapas UMN MapServer y las seis capas de datos espaciales en formato vectorial. La aplicación espacial contiene herramientas para consultar la información geográfica relacionada a los datos espaciales, realizar acercamiento y alejamiento del dato espacial, activar y desactivar capas de datos espaciales.

(2) Servidor de mapas – Fuentes de datos

¿Cómo es la interacción entre el servidor de mapas que contiene el algoritmo de selección y el esquema de selección de datos espaciales que se implementó en el MAPFILE?

- ✓ El esquema designado en el MAPFILE es reconocido por el algoritmo de selección que se desarrollo en el servidor de mapas; en específico reconoce las etiquetas claves independientes ATTEMPT y ACESSTYPE, así como también la etiqueta objeto ALTERNATIVE y las etiquetas claves independientes que contiene: NAME, DATA, CONNECTION y CONNECTIONTYPE (Ver Figura 7.2). Se obtienen satisfactoriamente los valores que son asignados a las etiquetas claves independientes mediante el archivo MAPFILE. Lo anterior es posible comprobarlo mediante el archivo de texto que se genera con la ayuda de las etiquetas DEBUG 5 y CONFIG "MS_ERRORFILE" "ruta donde se desea guardar el archivo" con las cuales se lleva un registro de actividades del servidor de mapas (Ver Figura 7.3).

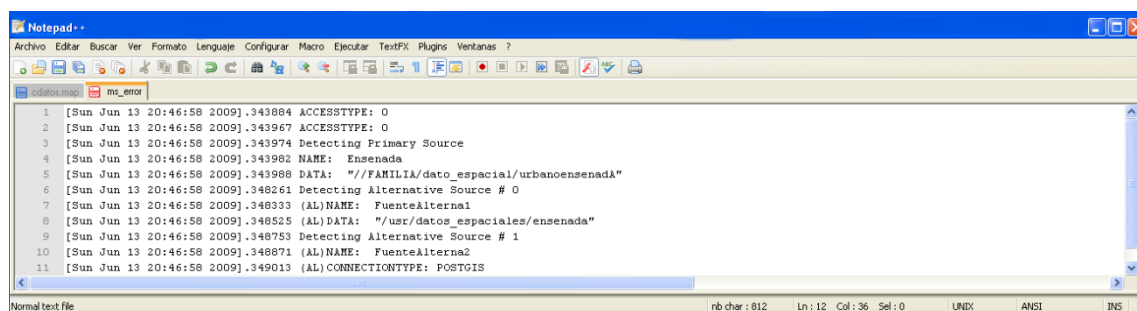


```

37 STATUS ON
38 TYPE LINE
39 CLASS
40   NAME Ensenada
41   STYLE
42     COLOR 0 167 197
43     OUTLINECOLOR 218 218 218
44   END
45 END
46
47 #####Fuentes Alternas#####
48 ACESSTYPE OPTIMISTIC
49
50 #dato espacial ubicado en un folder de la computadora que funge como servidor de mapas
51 #si existe el dato espacial
52 ALTERNATIVE
53   NAME FuenteAltern1
54   DATA "/usr/datos_espaciales/ensenada"
55 END
56
57 #dato espacial "ubicado" en computadora 148.231.216.143
58 #No existe el dato espacial
59 ALTERNATIVE
60   NAME FuenteAltern2
61   CONNECTIONTYPE POSTGIS
62   CONNECTION "host=148.231.216.143 port=5432 dbname=datos user=postgres password=123456"
63   DATA "the_geom from ensenada"
64 END
65
66 #####
67
68 END # Fin de layer
69

```

Figura 7.2 El servidor de mapas UMN MapServer detecta el esquema específico propuesto



```

1 [Sun Jun 13 20:46:58 2009].343884 ACESSTYPE: 0
2 [Sun Jun 13 20:46:58 2009].343967 ACESSTYPE: 0
3 [Sun Jun 13 20:46:58 2009].343974 Detecting Primary Source
4 [Sun Jun 13 20:46:58 2009].343982 NAME: Ensenada
5 [Sun Jun 13 20:46:58 2009].343988 DATA: "/FAMILIA/dato_espacial/urbanoensenada"
6 [Sun Jun 13 20:46:58 2009].348261 Detecting Alternative Source # 0
7 [Sun Jun 13 20:46:58 2009].348333 (A)NAME: FuenteAltern1
8 [Sun Jun 13 20:46:58 2009].348525 (A)DATA: "/usr/datos_espaciales/ensenada"
9 [Sun Jun 13 20:46:58 2009].348753 Detecting Alternative Source # 1
10 [Sun Jun 13 20:46:58 2009].348871 (A)NAME: FuenteAltern2
11 [Sun Jun 13 20:46:58 2009].349013 (A)CONNECTIONTYPE: POSTGIS

```

Figura 7.3 Registro de actividades: detección de fuentes

- ✓ Permite declarar un número ilimitado de fuentes alternas, debido a que se asigna dinámicamente memoria para la creación de fuentes de datos alternativas. Sin embargo para este proyecto de investigación se probaron con hasta 10 fuentes alternas de datos por LAYER.
- ✓ Están validadas las etiquetas claves independientes ACESSTYPE y ATTEMPT. ACESSTYPE debe tener como parámetro: OPTIMISTIC o PESSIMISTIC. En el caso de ATTEMPT debe tener como parámetro un entero (1, 2, 3 hasta N); si reciben cualquier otro parámetro o no se agrega uno, marca error.

Tipo de error	Descripción
[Mon Jun 15 18:26:19 2009].356494 loadLayer(): General error message. You need to choose PESSIMISTIC method for ATTEMPT keyword:(line x)	Si se elige en ACESSTYPE el método OPTIMISTIC y se agrega ATTEMPT, se mostrara este error. OPTIMISTIC no funciona con ATTEMPT, solamente con el método PESSIMISTIC.
[Mon Jun 15 18:26:19 2009].3565905 loadLayer(): General error message. You need to choose any method (OPTIMISTIC or PESSIMISTIC):(line x)	Cuando se declaran fuentes alternas y no se elige ningún método
[Mon Jun 15 18:26:19 2009].356515 loadLayer(): General error message. You need to add the keyword (ATTEMPT) in your mapfile:(line x)	Si se declara el método PESSIMISTIC es necesario agregar ATTEMPT
[Mon Jun 15 18:28:23 2009].356672 getSymbol():Symbol definition error. Parsing error near (independent key label or label object):(line x)	Si ACESSTYPE o ATTEMPT no contiene algún parámetro. Ej. ACESSTYPE _____

Tabla 7.3 Validación de etiquetas y mensajes de error

- ✓ En caso de que se presente el escenario de que ninguna fuente de datos (primaria y alternas) obtengan el dato espacial requerido. Se muestra el siguiente error:

Tipo de error	Descripción
[Mon Jun 15 18:40:37 2009].356588 msDrawMap():The spatial data does not exist in any data source. Please check the data sources or add more alternative sources:(line x)	Si la fuente primaria y fuentes alternas no contienen el dato espacial requerido en una capa LAYER. Se notifica con el siguiente mensaje de error en la aplicación espacial y en el archivo de texto que registra las actividades del servidor de mapas.

Tabla 7.4 Validación fuentes primaria y alterna no están disponibles

- ✓ Están validadas la etiqueta objeto ALTERNATIVE y las etiquetas claves independientes.

Tipo de error	Descripción
[Mon Jun 15 18:55:21 2009].356641 getString():Symbol definition error. Parsing error near (independent key label or label object):(line x)	Si las etiquetas claves independientes NAME, DATA y CONNECTION no contienen algún parámetro. Ej. DATA _____
[Mon Jun 15 18:56:17 2009].356689 msDrawMap():Image handling error. Failed to draw layer named:(line x) msShapefileOpen(): Unable access. (filename)	Si el dato espacial asignado a la etiqueta clave independiente DATA no existe o tiene un nombre diferente en la fuente de datos
[Mon Jun 15 22:56:09 2009].356823 msLoadMap(): Unknown identifier. Parsing error near (ALTERNATIVE):(line 90)	Si la etiqueta objeto ALTERNATIVE no es cerrada por la etiqueta objeto END.

Tabla 7.5 Validación de etiquetas y mensajes de error (ALTERNATIVE)

- ✓ El servidor de mapas genera un informe de actividades que contiene: El numero de fuente alternas declaradas y los parámetros que contienen, el tipo de método seleccionado para obtener datos (OPTIMISTIC o PESSIMISTIC), el numero de intentos que se realizaran en las fuente de datos alternas (ATTEMPT), los datos espaciales que son generados y errores en declaración de las etiquetas claves independientes y objeto, lleva un control de actividades con fecha y hora (Ver Figura 7.4).

Nota: Para habilitar la opción de generar informe se tiene que declarar las siguientes etiquetas claves independientes (Ver Figura 7.5).

Ejemplo:

DEBUG 5

CONFIG "MS_ERRORFILE" "ruta donde se desea guardar el archivo"

```

1 [Wed Aug 12 22:56:09 2009].265057 meLoadMap(): Unknown identifier. Parsing error near (ALTERNATIVE):(line 90)
2 [Wed Aug 12 22:56:09 2009].265172 meFreeMap(): freeing map at 0x9a1f968.
3 [Wed Aug 12 22:56:53 2009].428726 meLoadMap(): Unknown identifier. Parsing error near (ALTERNATIVE):(line 91)
4 [Wed Aug 12 22:56:53 2009].428794 meFreeMap(): freeing map at 0x97a0968.
5 [Wed Aug 12 22:57:02 2009].170905 meLoadMap(): Unknown identifier. Parsing error near (ALTERNATIVE):(line 91)
6 [Wed Aug 12 22:57:02 2009].171001 meFreeMap(): freeing map at 0x8583968.
7 [Wed Aug 12 22:57:29 2009].238942 ACCESSSTYPE: -1
8 [Wed Aug 12 22:57:29 2009].239070 loadLayer(): General error message. Debes declarar algun metodo de seleccion(OPTIMISTIC o PESSIMISTIC):(line 88)
9 [Wed Aug 12 22:57:29 2009].239079 meFreeMap(): freeing map at 0x8d1c968.
10 [Wed Aug 12 22:57:52 2009].299886 ACCESSSTYPE: 0
11 [Wed Aug 12 22:57:52 2009].299962 ACCESSSTYPE: 0
12 [Wed Aug 12 22:57:52 2009].299969 Detecting Alternative Source # 0
13 [Wed Aug 12 22:57:52 2009].461898 meDrawMap(): Layer 0 (Ensenada), 0.117s
14 [Wed Aug 12 22:57:52 2009].461941 meDrawMap(): Layer 1 (chapultepec), 0.000s
15 [Wed Aug 12 22:57:52 2009].461953 meDrawMap(): Drawing Label Cache, 0.000s
16 [Wed Aug 12 22:57:52 2009].461961 meDrawMap() total time: 0.124s
17 [Wed Aug 12 22:57:52 2009].495836 meSaveImage() total time: 0.034s
18 [Wed Aug 12 22:57:52 2009].496631 meSaveImage() total time: 0.000s
19 [Wed Aug 12 22:57:52 2009].506511 meSaveImage() total time: 0.010s
20 [Wed Aug 12 22:57:52 2009].512111 mapserv request processing time (loadmap not incl.): 0.175s
21 [Wed Aug 12 22:57:52 2009].512126 meFreeMap(): freeing map at 0x882e968.
22 [Wed Aug 12 23:20:49 2009].380232 ACCESSSTYPE: 0
23 [Wed Aug 12 23:20:49 2009].380336 ACCESSSTYPE: 0
24 [Wed Aug 12 23:20:49 2009].380342 Detecting Alternative Source # 0
25 [Wed Aug 12 23:20:49 2009].380350 (AL)NAME: FuenteAlternat
26 [Wed Aug 12 23:20:49 2009].380356 (AL)DATA: Chapultepec_lin
27 [Wed Aug 12 23:20:49 2009].380361 ACCESSSTYPE: 0
28 [Wed Aug 12 23:20:49 2009].380366 Detecting Alternative Source # 1
29 [Wed Aug 12 23:20:49 2009].380370 (AL)NAME: FuenteAlternat2
30 [Wed Aug 12 23:20:49 2009].380400 getString(): Symbol definition error. Parsing error near (END):(line 96)
31 [Wed Aug 12 23:20:49 2009].380408 meFreeMap(): freeing map at 0x978c968.
32 [Wed Aug 12 23:20:55 2009].987975 ACCESSSTYPE: 0
33 [Wed Aug 12 23:20:55 2009].988069 ACCESSSTYPE: 0

```

Figura 7.4 Visualización del informe de actividades

```

1 MAP
2
3
4 CONFIG "MS_ERRORFILE" "/usr/local/apache2/htdocs/appgeo/tmp2/ms_error.txt"
5
6 EXTENT 529395.83783573 3519769.55896037 544885.86544958 3531024.78183222
7
8 SIZE 400 400
9
10 SHAPEPATH "datos"
11
12 IMAGECOLOR 255 255 255
13
14
15 WEB
16
17 TEMPLATE "/usr/local/apache2/htdocs/appgeo/index.html"
18
19 IMAGEPATH "/usr/local/apache2/htdocs/appgeo/tmp/"
20
21 IMAGEURL "/appgeo/tmp/"
22
23 LOG "/usr/local/apache2/htdocs/appgeo/tmp2/ms_status.txt"
24
25 END
26
27 # Definicion de simbolo para puntos de puentes
28
29 SYMBOL
30
31 NAME 'circle'
32
33 TYPE ELLIPSE
34
35 POINTS 1 1 END
36
37 FILLED TRUE
38
39 END
40
41 #####Capas de datos espaciales#####
42
43 #MAPAS
44
45 #####INICIO LAYER 1#####
46
47 # Dato espacial de la zona urbana de Ensenada
48
49 LAYER
50
51 NAME Ensenada

```

Figura 7.5 Declarar CONFIG a DEBUG para generar informe de actividades

- ✓ Con base en el archivo de configuración MAPFILE mostrado anteriormente para la realización de las pruebas ALFA, se pudo comprobar el funcionamiento correcto de los métodos PESSIMISTIC y OPTISMISTIC.

Ambos métodos pueden utilizarse al mismo tiempo en el archivo MAPFILE, solo condicional de que exista un método definido por cada LAYER. Por ejemplo en el archivo de configuración

MAPFILE se utiliza el método OPTIMISTIC para las fuentes alternas del dato espacial de la zona urbana de Ensenada y para el dato espacial de la delegación municipal de Chapultepec se utiliza el método PESSIMISTIC. Por lo tanto puede existir una combinación de métodos para la selección de fuentes alternas o bien la utilización de un solo método para todas las fuentes alternas; cabe señalar que debe ser agregado el método en cada una de las capas (LAYER).

Conclusiones

8

Este capítulo contiene un resumen resultados obtenidos en el proyecto de investigación, conclusiones, aportaciones y trabajos a futuro.

8.1 Resultados Finales

Durante las 3 etapas de desarrollo que se designaron para este proyecto de investigación se mostraron los resultados finales, es por ello que en este apartado se presenta un resumen de resultados obtenidos en esta investigación:

1. **Aplicación espacial:** Se diseñó una aplicación espacial para un entorno web que permite interactuar con el servidor de mapas UMN MapServer y con los datos espaciales almacenados en fuentes remotas y locales (Ver Figura 8.1). Además la aplicación fue una parte fundamental en este proyecto de investigación ya que permitió comprobar el funcionamiento del algoritmo de selección de datos espaciales desarrollado a partir del esquema propuesto.

Por otro lado cabe señalar que existió una colaboración con especialistas en la gestión de riesgos que aportaron información valiosa para la generación de datos espaciales a partir de un estudio que determina el índice de vulnerabilidad sísmica de los puentes urbanos de Ensenada Baja California México.

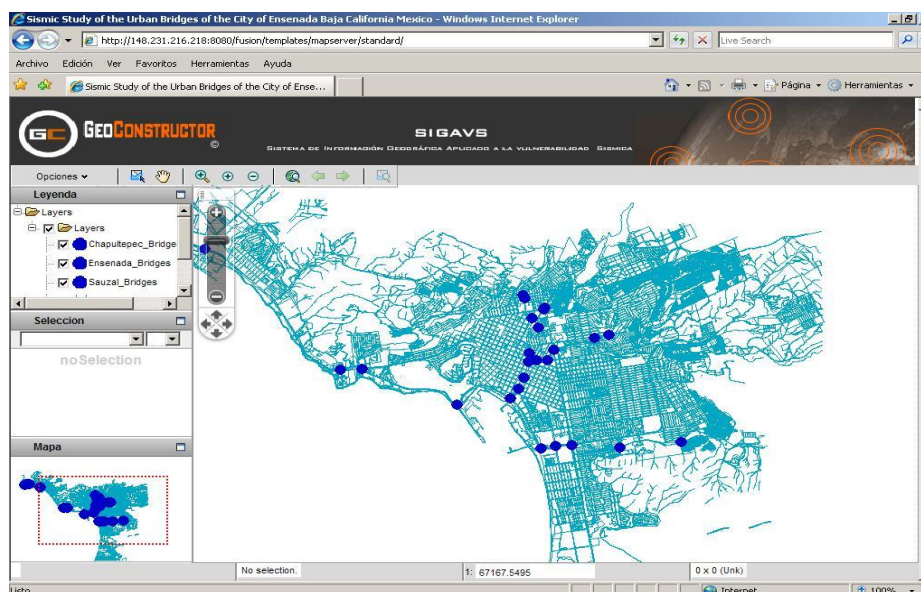


Figura 8.1 Aplicación espacial

2. **Esquema de selección de datos espaciales:** Se diseñó un esquema general y un esquema específico. El esquema general propone la adopción de métodos en servidores de mapas que permitan obtener datos espaciales: de fuentes alternas, de manera automatizada y transparente al usuario, cuando el dato espacial no esté disponible en la fuente primaria (Ver Figura 8.2).

El esquema específico es la implementación del esquema general para el servidor de mapas UMN MapServer; el esquema específico define los métodos: OPTIMISTIC y PESSIMISTIC que ayudan a seleccionar y obtener datos espaciales de fuentes alternas, la forma en cómo se declara una fuente alterna y los parámetros que debe contener (Figura 8.3).

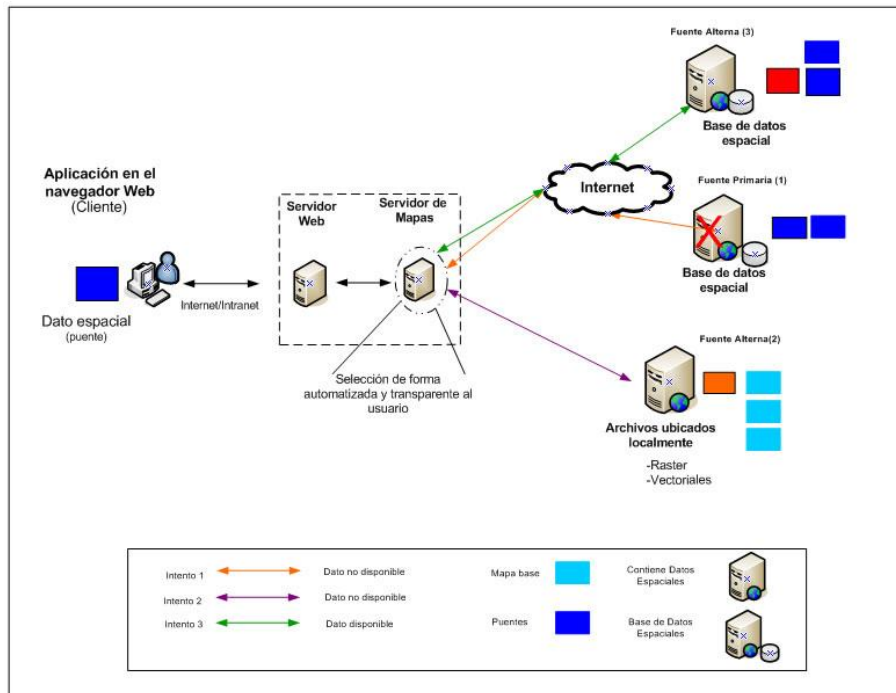


Figura 8.2 Esquema general de selección de datos espaciales

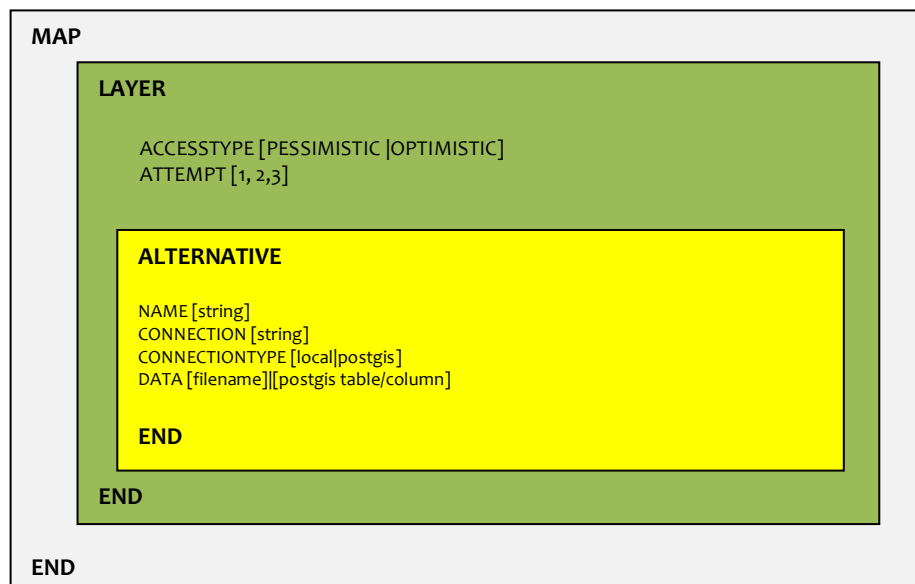


Figura 8.3 Esquema específico de selección de datos espaciales

3. **Algoritmo de selección de datos espaciales:** Se desarrollaron diversas funciones (Ver descripción de funciones en capítulo 6.12) en el código fuente del servidor de mapas UMN MapServer que permitieron implementar el esquema específico propuesto (Ver Figura 8.3).

Nota: El código de las funciones desarrolladas puede descargarse desde el sitio web <http://geoconstructor.ens.uabc.mx>, <http://www.geoconstructor.com>.

Con base en las diversas pruebas que se realizaron, el algoritmo fue capaz de obtener datos espaciales de diversas fuentes de datos tanto remotas en bases de datos PostgreSQL como locales directamente de archivos vectoriales (.SHP), así como también bases de datos PostgreSQL creadas localmente.

El algoritmo funcionó correctamente y pudo obtener datos espaciales de forma automatizada, transparente al usuario y de fuentes alternas cuando la fuente de datos primaria no estuvo disponible.

4. **Sitio Web (Complementario):** Se implementó un sitio web denominado GEOCONSTRUCTOR que permite integrar: la aplicación espacial y publicaciones de artículos, carteles y ponencias que se realizaron durante el proyecto de investigación (Ver Figura 8.4).



Figura 8.4. Sitio web GeoConstructor

8.2 Conclusiones

El análisis realizado a diversos servidores de mapas comerciales, de código abierto y servicios gratuitos, así como también aplicaciones espaciales y mecanismos que buscan garantizar la disponibilidad de los datos espaciales, contribuyeron a plantear las necesidades, áreas de oportunidad y correcto entendimiento de la problemática relacionada con la disponibilidad de datos espaciales en servidores de mapas.

El diseño y desarrollo de la aplicación espacial enfocada al índice de vulnerabilidad de los puentes urbanos de Ensenada Baja California México, permite tener un registro de las características de los puentes urbanos de Ensenada, ayudará a los especialistas en la gestión de riesgos interactuar de forma amigable con los datos espaciales, servirá de base para realizar estrategias y mitigaciones de riesgo facilitando la toma de decisiones en caso de un sismo o la posible falla de un puente y finalmente el catálogo de fotos implementado en la aplicación espacial ayudará a describir con mayor facilidad el tipo de puentes urbanos.

El diseño del esquema general es una propuesta para la adopción de métodos en los servidores de mapas, que permitan obtener datos espaciales: de fuentes alternas, de manera automatizada y transparente al usuario, cuando el dato espacial no esté disponible en la fuente primaria.

El diseño e implementación del esquema específico que da origen al algoritmo de selección de datos espaciales, contribuye al mejoramiento del servidor de mapas UMN MapServer. Respecto al algoritmo de selección ayuda mejorar la disponibilidad de los datos espaciales en el servidor de mapas; ya que cuando el servidor de mapas detecta la falla de la fuente primaria, el algoritmo selecciona las fuentes alternas y obtiene el mismo dato espacial que fue requerido en la fuente primaria, realizando el anterior proceso de forma automatizada y transparente al usuario con la ayuda de los métodos OPTIMISTIC y PESSIMISTIC. El algoritmo es capaz de obtener datos espaciales tanto de fuentes remotas como locales en bases de datos PostgreSQL y como archivos vectoriales (.SHP).

Con base en los anteriores hechos se puede concluir que los objetivos planteados para el proyecto de investigación se cumplieron satisfactoriamente.

8.3 Aportaciones

Las aportaciones que brinda el presente proyecto de investigación son los siguientes:

- La propuesta del esquema general marca la pauta y sirve de base para que los diversos servidores de mapas comerciales, código abierto o hasta los de servicios gratuitos puedan implementar métodos que permitan obtener datos espaciales de fuentes alternas en caso de que falle la fuente primaria. La propuesta del esquema específico y su implementación en el servidor de mapas UMN MapServer, mejora la disponibilidad de los datos espaciales en el servidor de mapas, tiene cierto control sobre las fuentes de datos, permite un número ilimitado de fuentes alternas y obtiene de forma automatizada y transparente al usuario datos espaciales. Además se contribuye al mejoramiento del servidor de mapas UMN MapServer.
- El beneficio de desarrollar un algoritmo de selección de datos espaciales en un servidor de mapas; permitirá a particulares, empresas, organizaciones o dependencias proveedoras de servicios geográficos mejorar la disponibilidad de sus datos espaciales, aumentando la confiabilidad de los usuarios respecto a la proveedora de servicios en correlación con la disminución en pérdidas económicas debido a la satisfacción del usuario por la disponibilidad de los datos..

Cabe mencionar que el grado de impacto negativo o repercusión que se tenga al no estar disponible un dato espacial cuando este sea requerido, dependerá del enfoque y relevancia de la aplicación espacial. Pudiendo afectar de diversas formas a la organización, empresa o dependencia que preste servicios a través de servidores de mapas y a los mismos usuarios. Por ejemplo existen aplicaciones espaciales que manejan un control en tiempo real de incendios (<http://www.geomac.gov/viewer/viewer.htm>) o tráfico vehicular <http://traffic.houstontranstar.org/layers/>, donde la información que se maneja es de alta prioridad, debido a los objetivos de las anteriores aplicaciones como: combate a incendios y posibles afectaciones a flora, fauna y sectores habitacionales, gestión de emergencias y advertencias para disminuir el número de muertes, lesiones, daños en propiedades ocasionados por inundaciones, entre otras.

De acuerdo a la alta prioridad en la información que manejan estas aplicaciones es necesario considerar un algoritmo que les permita mejorar la disponibilidad de sus datos. Sin embargo cabe señalar que el algoritmo que se desarrollo funciona independientemente del tipo de aplicación espacial y prioridad de la información que se maneje, implicando que la mejora en la disponibilidad de los datos espaciales sería el mismo en aplicaciones espaciales que manejen información tanto alta prioridad como de baja prioridad.

8.4 Trabajo futuro

El presente proyecto es una versión susceptible a mejoras como las que se muestran a continuación:

- Sería conveniente incrementar el soporte a diversas bases de datos; ya que solamente el algoritmo es capaz de obtener datos espaciales de bases de datos PostgreSQL.
- Es necesario realizar pruebas con diversos datos espaciales en formatos raster y vectoriales, debido a que este estudio sólo se realizaron pruebas con datos espaciales en formato vectorial (.SHP)
- Realizar una aplicación que muestre de forma más amigable el registro de actividades del servidor de mapas, ya que actualmente te muestra la información en un archivo de texto (.txt).
- Método de integridad de datos espaciales:
Es necesario implementar un método que garantice que los datos espaciales que se obtengan de las fuentes de datos realmente sean los que se solicitaron; ya que normalmente la verificación de integridad de datos espaciales la lleva a cabo la persona que administra el servidor de mapas, algunas veces obtiene datos espaciales fuentes que están fuera de su control y que en consecuencia no puede verificar si los datos espaciales son íntegros o no.
- Método de verificación de disponibilidad de datos espaciales de lado del cliente:
Que permita a los clientes analizar desde su aplicación espacial el archivo Mapfile y detecte la disponibilidad de las fuentes de datos, permitiendo a los clientes reportar la falla de una fuente de datos al administrador del servidor de mapas.
- Nuevos métodos para obtener datos espaciales de fuentes alternas, de forma automatizada y transparente al usuario cuando la fuente primaria no se encuentre disponible(Monge,2008) por ejemplo :
 1. Método con base a prioridades: Las fuentes alternas se les designará una prioridad: alta, media y baja. Siendo la prioridad alta la fuente que primero se consultará y la prioridad baja la última en consultarse.

2. Método distancia geográfica: Las fuentes alternas se ordenarían dependiendo de su localización geográfica; de tal manera que la fuente más cercana geográficamente al servidor de mapas, será la que se consultará primero.
3. Método ruta más corta de la red: Para seleccionar una fuente alterna se evaluaría el número de saltos y el tiempo de respuesta, con la finalidad de identificar la fuente alterna que responda más rápido poder para consultarse.
4. Método dinámico: Que permita obtener un dato espacial de fuentes alternas que no necesariamente estén definidas en el MAPFILE, que sea capaz de buscar y obtener un dato espacial en un ambiente distribuido.

La razón de proponer nuevos métodos que permitan obtener datos espaciales de forma automatizada y transparente al usuario, es principalmente por el interés de realizar un análisis comparativo en cuanto al rendimiento y rapidez de los nuevos métodos propuestos y los métodos desarrollados en este proyecto de investigación.

- Respecto a la aplicación espacial se tiene considerado incluir en la aplicación espacial más capas de datos espaciales, referentes a investigaciones de vulnerabilidad de sísmica en diversas estructuras de Ensenada Baja California, para generar una aplicación más integral. Además se agregarán videos que permitan a expertos en el área de gestión de riesgos compartir: sus opiniones, análisis, experiencias, prácticas educativas en la recolección de información de campo, entre otros.

Bibliografías

BIBLIOGRAFÍAS

- [amanda,2009] Software para el respaldo de información en Linux AMANDA. [Consulta: 26-05-09]. Disponible en: <http://www.amanda.org/>
- [ArcIMS,2009] Servidor de mapas comercial ArcIMS [Consulta: 8-01-2009]. Disponible en: <http://www.esri.com/software/arcgis/arcims/index.html>
- [ArcGIS Server,2009] Servidor de mapas comercial ArcGIS [Consulta: 8-01-2009]. Disponible en: <http://www.esri.com/software/arcgis/arcgisserver/index.html>
- [API GoogleMaps,2009] API del Servidor de mapas de servicios gratuitos Google Maps [Consulta: 2-05-2009]. Disponible en: <http://code.google.com/intl/es-ES/apis/maps/>
- [API Map24,2009] API del Servidor de mapas de servicios gratuitos Map24 [Consulta: 2-05-2009]. Disponible en: <http://developer.navteq.com/site/global/zones/ms/index.jsp>
- [API MapQuest,2009] API del Servidor de mapas de servicios gratuitos MapQuest [Consulta: 2-05-2009]. Disponible en: http://developer.mapquest.com/Library/SDK_Documentation/
- [API LiveSearchMap,2009] API del Servidor de mapas de servicios gratuitos Live Search Map [Consulta: 2-05-2009]. Disponible en: <http://maps.live.com/>
- [API YahoMaps,2009] API del Servidor de mapas de servicios gratuitos Yahoo Maps [Consulta: 2-05-2009]. Disponible en: <http://developer.yahoo.com/maps/>
- [CALCOM,2009] Costo de batería UPS [Consulta: 18-03-2009]. Disponible en: <http://www.calcom.com>
- [Deegree,2009] Servidor de mapas de código abierto Deegree [Consulta: 8-01-2009]. Disponible en: <http://www.deegree.org/>
- [dev.mysql,2009] Extensión espacial Spatial Extensions [Consulta: 20-01-2009]. Disponible en: <http://dev.mysql.com/doc/refman/5.0/en/spatial-extensions.html>
- [dlink,2009] Costos de componentes de red [Consulta: 18-03-2009]. Disponible en: <http://www.dlink.com/>
- [ESRI, 1998] ShapeFile Technical Description, an ESRI white paper. California, USA. Editora ESRI [Consulta: 18-11-2008]. Disponible en: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>
- [GDAL, 2009] Geospatial Data Abstraction Library [Consulta: 20-01-2009]. Disponible en: <http://www.gdal.org/>
- [GDB, 2009] GNU Project Debugger [Consulta: 7-03-2009]. Disponible en: <http://www.gnu.org/software/gdb/>
- [GeomediaWeb,2009] Servidor de mapas comercial GeoMedia Web [Consulta: 8-01-2009]. Disponible en: <http://www.intergraph.com/sgi/products/>
- [GeoServer,2009] Servidor de mapas de código abierto GeoServer [Consulta: 8-01-2009]. Disponible en: <http://geoserver.org>
- [Gilroy,2001] Gilroy, John [2001]: Neurología. MacGraw Hill Interamericana Tercera Edición. ISBN 970-10-2894-5 pp. 428

- [GOOGLE,2009] Condiciones de uso del servidor de mapas de servicios gratuitos Google Maps. [Consulta: 2-05-2009] http://maps.google.com/help/terms_maps.html
- [GoogleMaps,2009] Servidor de mapas de servicios gratuitos Google Maps [Consulta: 2-05-2009]. Disponible en: <http://maps.google.es/>
- [gopower,2009] Costo de generador eléctrico [Consulta: 18-03-2009]. Disponible en: <http://www.gopower.com/>
- [Grupo RESNOM, 2006] Grupo RESNOM [2006]. *Boletín de la red sísmica del noroeste de México (periodo enero a diciembre de 2005)*. Boletín Informativo de la Unión Geofísica Mexicana, GEOS, Vol. 25, No. 3, México.
- [handybackup,2009] Software para el respaldo de información Handy Backup [Consulta: 26-05-09]. Disponible en: <http://www.handybackup.com/>
- [Harder, 1998] Harder, Christian [1998] *Serving maps on the Internet*. ESRI Press. pp. 16
- [Heather, 2001] Heather, Kennedy [2001]. *Dictionary of GIS Terminology*. ESRI Press 2001 ISBN 1879102781
- [Horak,2008] Horak, Ray [2008] *Webster's New World Telecom Dictionary*. Editorial Wiley. ISBN 978-0-471-77457-0. Pp 36
- [IntelCompras, 2009] Costo servidor IBM Bladecenter. [Consulta: 18-03-2009]. Disponible en: <http://www.intelcompras.com/blade-center-first-blade-paquete-empresarial-pyme-p-14794.html>
- [Juárez et. al., 2007] Juárez García, Jesús; Espinoza Barreras, Fortunato; Torres Herrera, Pablo; Huerta López, Carlos; Maldonado Rondón, Esperanza; Monge De La Cruz, Luis; De Jesús Ricalde y Lugo, Felipe. [2007]. "Sistema de Información Geográfica Puentes Urbanos de Ensenada". XVI Congreso Nacional de Ingeniería Sísmica, Ixtapa Zihuatanejo, México.
- [Kennedy, 2000] Kennedy, Heather [2000] *Dictionary of GIS Terminology*. ESRI Press ISBN 1879102781
- [linuxvirtualserver,2009] Balanceo de cargas por software. [Consulta: 18-03-2009]. Disponible en: <http://www.linuxvirtualserver.org/>
- [Linux-es, 2009] Sistemas informáticos redundantes. [Consulta: 18-03-2009]. Disponible en: <http://www.linux-es.org/node/211>
- [LiveSearchMaps,2009] Servidor de mapas de servicios gratuitos Live Search Maps [Consulta: 2-05-2009]. Disponible en: <http://maps.live.com/>
- [Longley et al, 2005] Longley, Paul A.; Goodchild, Michael F; Maguire, David J.; Rhind, David W. [2005]: *Geographic Information systems and Science*. Second Edition. Editorial John Wiley & Sons y ESRI Press. ISBN: 978-0-470-87001-3
- [Map24,2009] Servidor de mapas de servicios gratuitos Map24 [Consulta: 2-05-2009]. Disponible en: <http://www.mx.map24.com/>
- [MapGuide,2009] Servidor de mapas comercial Autodesk MapGuide Enterprise [Consulta: 8-01-2009]. Disponible en: <http://usa.autodesk.com/adsk/servlet/index?id=6546938&siteID=123112>
- [MapGuideOS,2009] Servidor de mapas de código abierto MapGuide OS [Consulta: 8-01-2009]. Disponible en: <http://mapguide.osgeo.org/>

- [mapinfo,2009] Extensión espacial Spatialware [Consulta: 20-01-2009]. Disponible en: <http://www.mapinfo.com/products/developer-tools/enterprise-software/spatialware>
- [Manifold,2009] Servidor de mapas comercial Manifold [Consulta: 8-01-2009]. Disponible en: <http://www.manifold.net>
- [Mapnik,2009] Servidor de mapas de código abierto Mapnik [Consulta: 8-01-2009]. Disponible en: <http://www.mapnik.org/>
- [MapQuest,2009] Servidor de mapas de servicios gratuitos MapQuest [Consulta: 2-05-2009]. Disponible en: <http://www.mapquest.com/>
- [Mapserver,2009] Servidor de mapas de código abierto UMN MapServer [Consulta: 8-01-2009]. Disponible en: <http://mapserver.org>
- [MapXtreme,2009] Servidor de mapas comercial MapInfo MapXtreme [Consulta: 8-01-2009]. Disponible en: <http://www.mapinfo.com/products/developer-tools/desktop%2c-mobile-%26-internet-offering/mapxtreme-java>
- [microsoft,2009] Respaldo de datos con Windows XP Backup [Consulta: 20-01-2009]. Disponible en: http://www.microsoft.com/windowsxp/using/setup/learnmore/bott_03july14.mspx
- [McKenna, 2009] McKenna, Jeff. Anatomía del servidor de mapas UMN MapServer [Consulta: 20-04-2009]. Disponible en: <http://mapserver.org/introduction.html#introduction>
- [Mitchell, 2005] Mitchell, Tyler [2005]: Web Mapping Illustrated. Editorial O'Really. ISBN 0-596-00865-1
- [Monge,2008] Monge De La Cruz, Luis Angel; Torres Herrera , Juan Pablo; Navarro Cota , Christian Xavier; López Chico, Luz Evelia [2008] Análisis y esquema de selección-recuperación de datos espaciales en un servidor de mapas. Memorias CD Congreso Vértice
- [Neumann,2008] Neumann, Andreas (2008): Encyclopedia of GIS -Web Mapping and web cartography. ISBN 978-0-387-35975-5. Editorial Springer pp.1263
- [OGC, 2009] Open Geospatial Consortium [Consulta: 20-04-2009]. Disponible en: <http://www.opengeospatial.org/>
- [oracle,2009] Extensión espacial Oracle Spatial [Consulta: 20-01-2009]. Disponible en: <http://www.oracle.com/technology/products/spatial/index.html>
- [osgeo, 2009] Organización que promueve el desarrollo colaborativo de tecnologías abiertas OSGEO [Consulta: 20-01-2009]. Disponible en: <http://www.osgeo.org/>
- [openstreetmap,2009] OpenStreetMap [Consulta: 20-01-2009]. Disponible en: <http://www.openstreetmap.org/>
- [Padrón et al, 2003] Padrón Diego, Prado Elena, Chuvieco Emilio [2003]: Empleo de servidores cartográficos en internet para la gestión y manejo de desastres. Foro Euro Mediterráneo sobre prevención de catástrofes. Recurso digital: http://www.proteccioncivil.org/es/DGPCE/Informacion_y_documentacion/catalogo/carpeta04/foroeuromediterraneo2003/cd/docs/seminario/sesion2/S2-4.pdf
- [phpmyadmin,2009] Administrador de bases de datos PHPMyAdmin. [Consulta: 26-05-2009]. Disponible en: <http://www.phpmyadmin.net/>

- [phppgadmin,2009] Administrador de bases de datos PHPpgAdmin. [Consulta: 26-05-2009]. Disponible en: <http://phppgadmin.sourceforge.net/>
- [postgis,2009] Extensión espacial PostGIS [Consulta: 20-01-2009]. Disponible en: <http://postgis.refractory.net/>
- [Pressman, 2002] Pressman, Roger. [2002] Ingeniería del software, un enfoque practico. Editorial McGraw-Hill 5ta Edición. ISBN 0-07-709677-0. Pp. 316
- [Ramsey, 2006] Ramsey, Paul, Mashing up the Enterprise [Consulta: 20-01-2009]. Disponible en: <http://www.refractory.net/expertise/whitepapers/mashups/mashups/>
- [Serra, 2002] Serra del Pozo, Pau. [2002]: Cinco servidores de mapas. Revista Internacional de Ciencias de la Tierra Mapping Interactivo. [Consulta: 18-11-2008]. Disponible en: http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=179 ISSN 1131-9100, N°81, pp. 6-12
- [Sommerville, 2002] Sommerville, Ian [2002]: Ingeniería de Software 6ta edición .Addison Wesley. ISBN 970-26-0206-8
- [symantec,2009] Software para el respaldo de información Norton Ghost [Consulta: 26-05-09]. Disponible en: <http://www.symantec.com/es/mx/norton/ghost>
- [W3C, 2009]. World Wide Web Consortium [Consulta: 23-02-09]. Disponible en: <http://www.w3.org/CGI/>
- [YahooMaps,2009] Servidor de mapas de servicios gratuitos Yahoo Maps [Consulta: 2-05-2009]. Disponible en: <http://espanol.maps.yahoo.com/>

Anexo

Manual de desarrollo para el servidor de mapas UMN MapServer

Los cuestionamientos más recurrentes durante el proyecto de investigación fueron: ¿Qué entorno de desarrollo es el indicado para UMN MapServer?, ¿Cómo se implementa código en UMN MapServer?.

Por medio de la realización de pruebas a diferentes entornos de desarrollo y depuradores, se encontró respuesta a los cuestionamientos antes mencionados, dando origen a este manual, el cual es una propuesta de cómo se puede programar, compilar y depurar código fuente para UMN MapServer. Cabe señalar que este manual no tiene el propósito de enmarcarse como la mejor o la única forma de llevar a cabo la programación, compilación y depuración de código fuente UMN MapServer, sin embargo si da solución de forma clara a los cuestionamientos de qué entornos utilizar y cómo hacerlo.

A continuación se detallan los pasos necesarios para desarrollar en UMN MapServer:

Paso 1: Detalles a considerar

A partir ingeniería inversa fue posible conocer el funcionamiento interno del servidor de mapas UMN MapServer. De acuerdo a Ian Sommerville la ingeniería inversa se utiliza para recuperar el diseño o la especificación del sistema a partir de su código fuente [Sommerville, 2002]. Para llevar a cabo este procedimiento se utilizó el depurador GDB y un entorno gráfico llamado KDbg. El depurador GDB permite observar lo que sucede dentro de un programa mientras este se ejecuta [GDB, 2009], en el caso del entorno gráfico KDBG es para facilitar la interacción con el depurador.

Paso 2: Instalación de GDB y KDbg en CentOS 5

El sistema operativo Linux distribución CentOS 5 posee un administrador para la instalación de software, por medio de este instala KDbg y GDB. Se debe habilitar la caja de verificación que aparece al lado izquierdo de cada software y posteriormente presionar el botón aplicar (Ver Figura 9.1). Normalmente este administrador indica las librerías o software que se necesitan para el funcionamiento de KDBG y GDB, solo es cuestión de aceptar la indicación para que las instale.

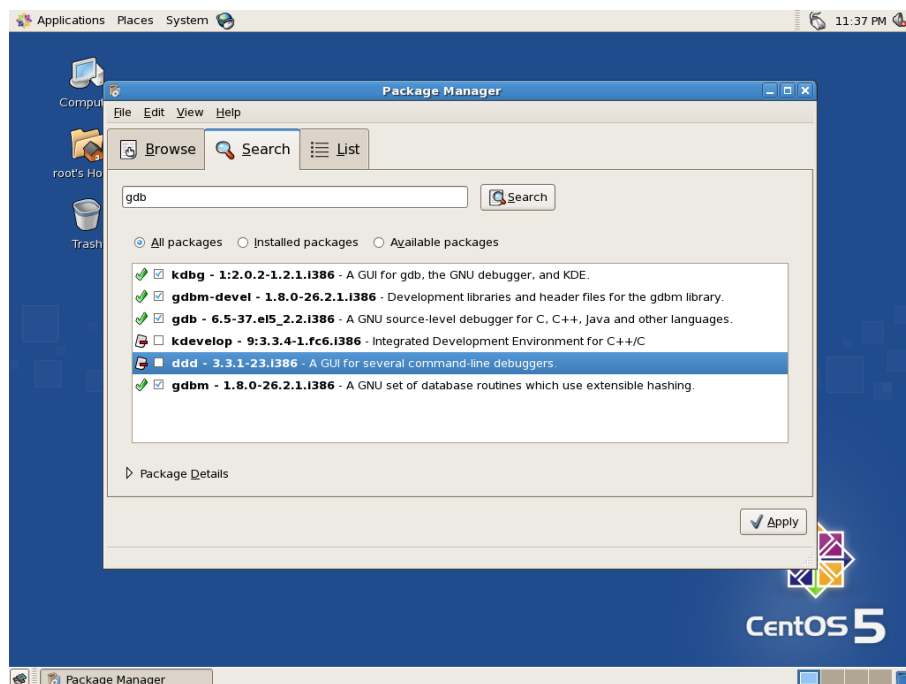


Figura 9.1 . Instalación del depurador y entorno de grafico

Paso 3: Abrir KDBG

Para abrir el entorno gráfico KDBG, ir al menú del sistema operativo, parte superior de la pantalla en Applications→Programming→KDBG (Ver Figura 9.2).

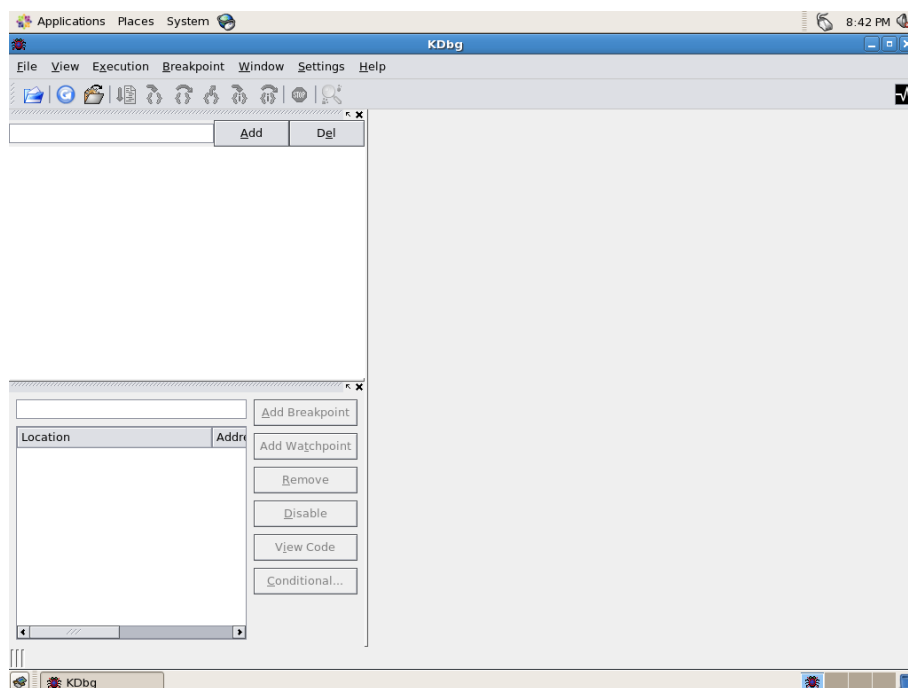


Figura 9.2. Abrir entorno gráfico KDBG

Paso 4: Abrir ejecutable de UMN MapServer (mapserv)

En este paso se considera que el servidor de mapas fue compilado previamente vía línea de comandos con las librerías requiere. Por lo tanto, el ejecutable de UMN MapServer debió ser generado y solamente se adjuntará con el entorno gráfico KDbg (Ver Figura 9.3).

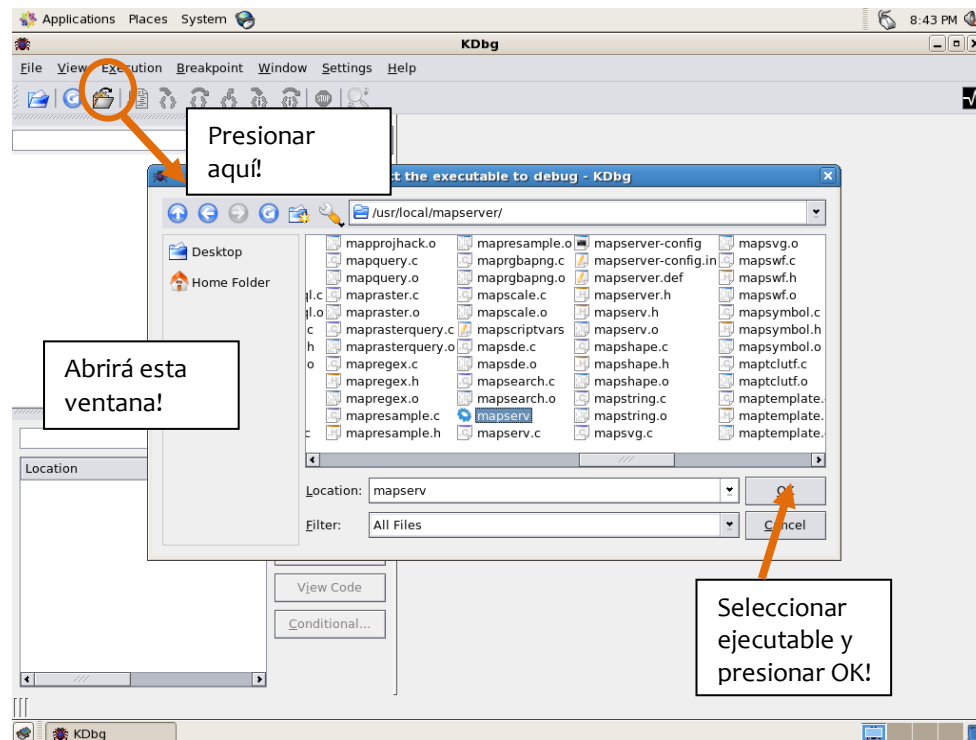


Figura 9.3. Adjuntar ejecutable mapserv desde KDBG

Al presionar el botón OK, te abrirá el archivo mapserv.c, el cual es el archivo principal del servidor de mapas (Ver Figura 9.4).

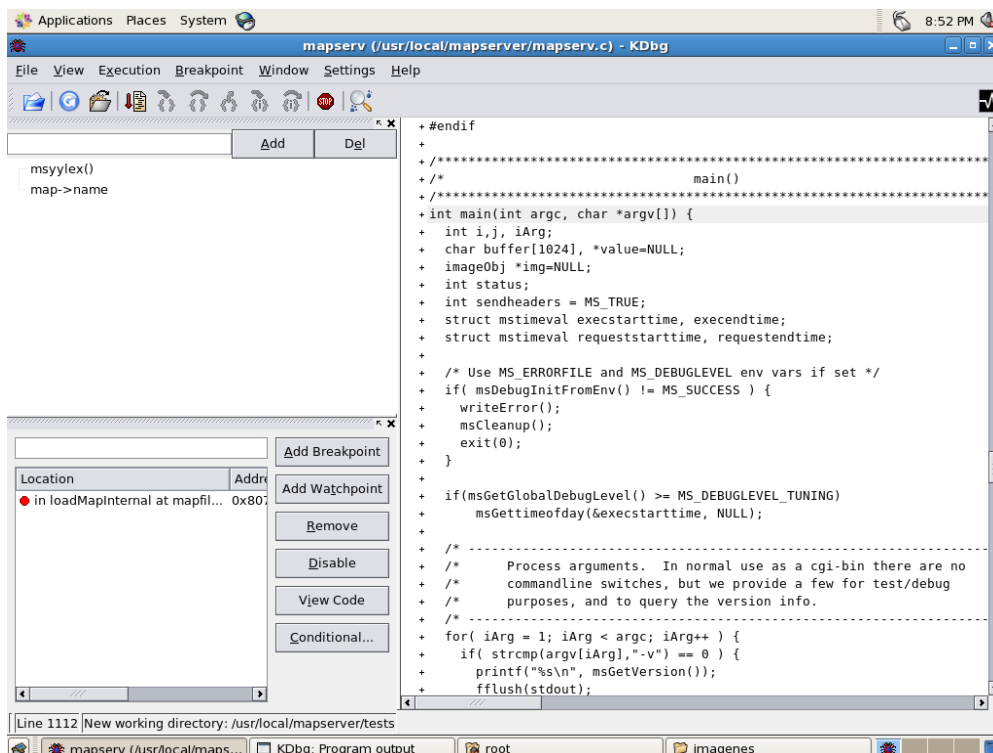


Figura 9.4. Resultado obtenido a partir del ejecutable mapserv

Paso 5: Adjuntar código fuente

Se adjuntará el archivo que se desea conocer su funcionamiento, en este caso se adjunta el archivo mapfile.c (Ver Figura 9.5)

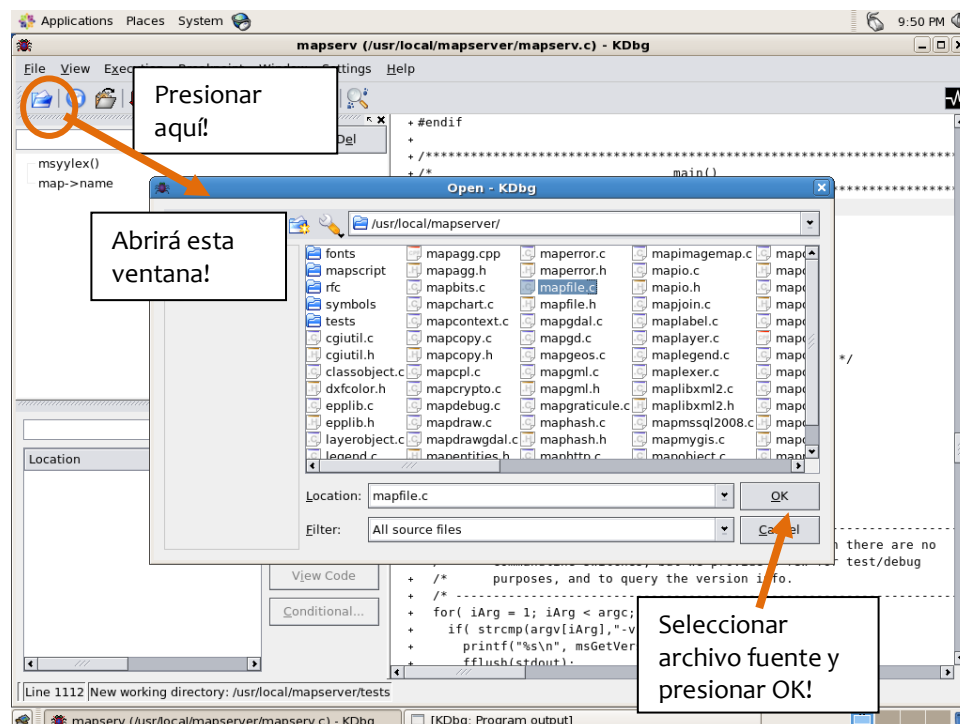


Figura 9.5. Adjuntar archivo fuente desde KDBG

Al presionar el botón OK, te abrirá el archivo con código fuente mapfile.c, el cual es el archivo que se desea conocer su funcionamiento (Ver Figura 9.6).

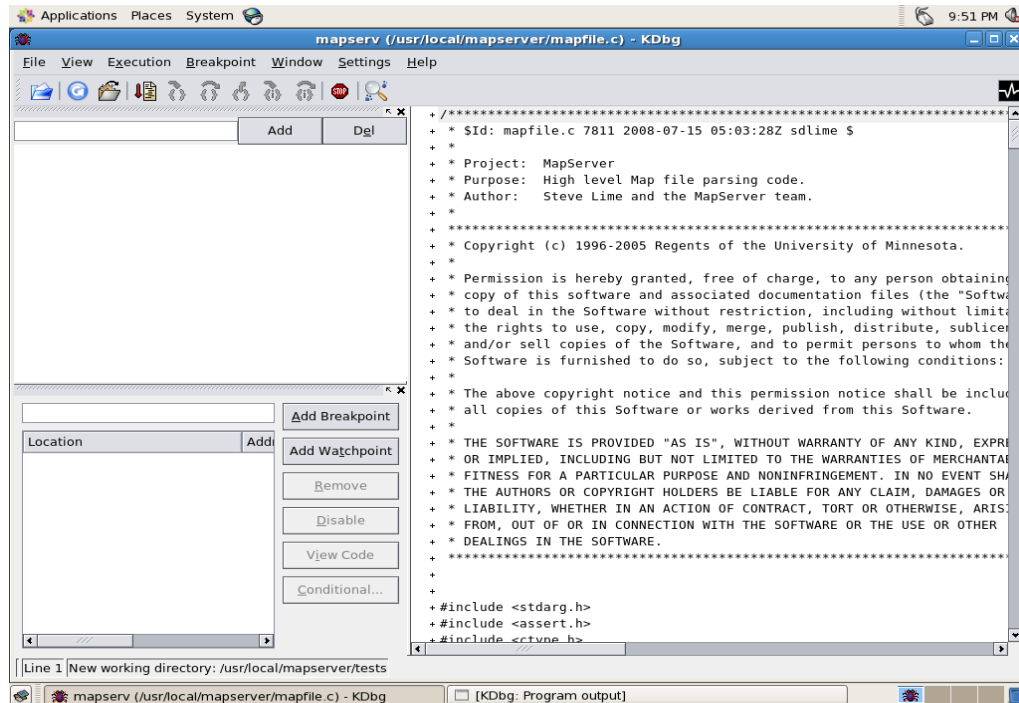


Figura 9.6. Resultado obtenido al adjuntar un archivo fuente.

Paso 6: Introducir valores de prueba.

La única forma de introducir valores de prueba en el entorno grafico KDbg, es por medio de una cadena de texto y la ruta del directorio de trabajo (Ver Figura 9.7).

Cadena de texto:

“QUERY_STRING=map=/usr/local/mapserver/tests/test.map&mode=map”

Ruta del directorio de trabajo:

/usr/local/mapserver/tests

La anterior cadena de texto indica la ubicación del archivo fuente que contiene los valores de prueba (Ver Figura 9.8), en este caso son los valores asignados a las etiquetas objeto y las etiquetas claves independientes del archivo Mapfile las que se introducen como valores de prueba (Ver Figura 9.9)

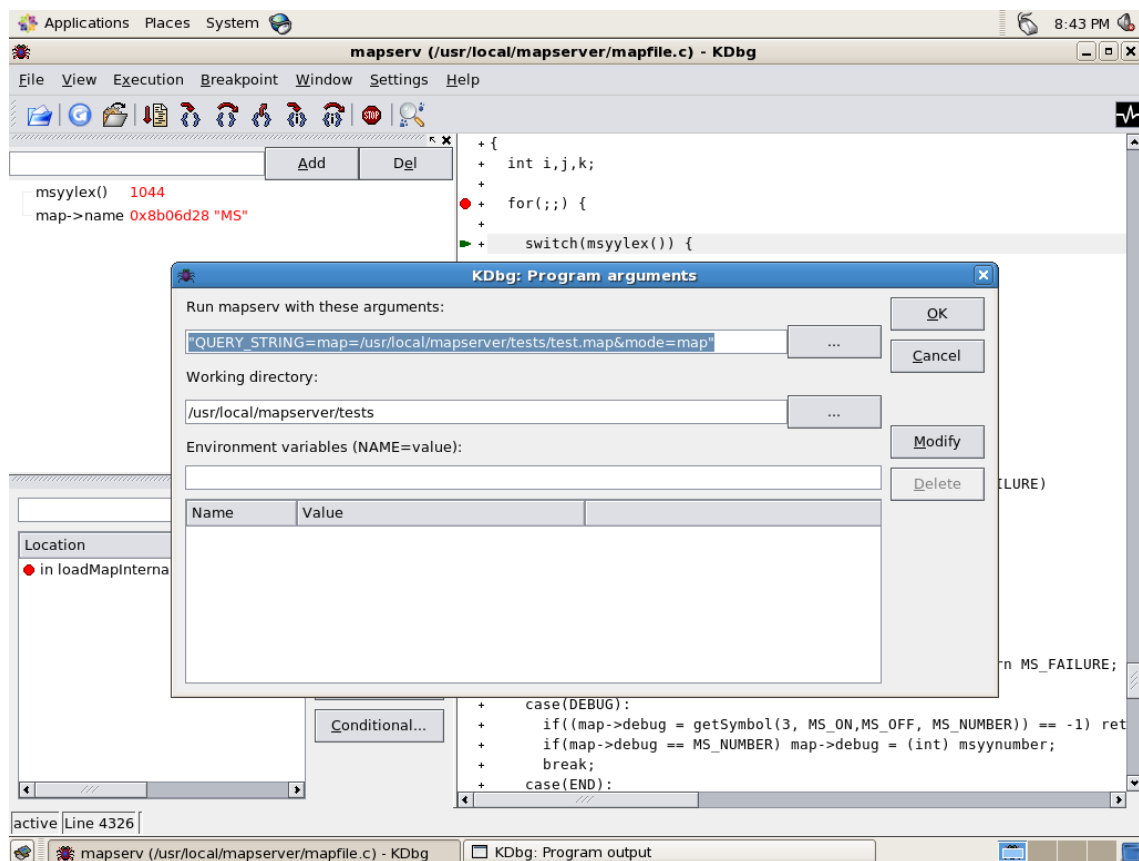


Figura 9.7. Forma de introducir valores de prueba en KDbg

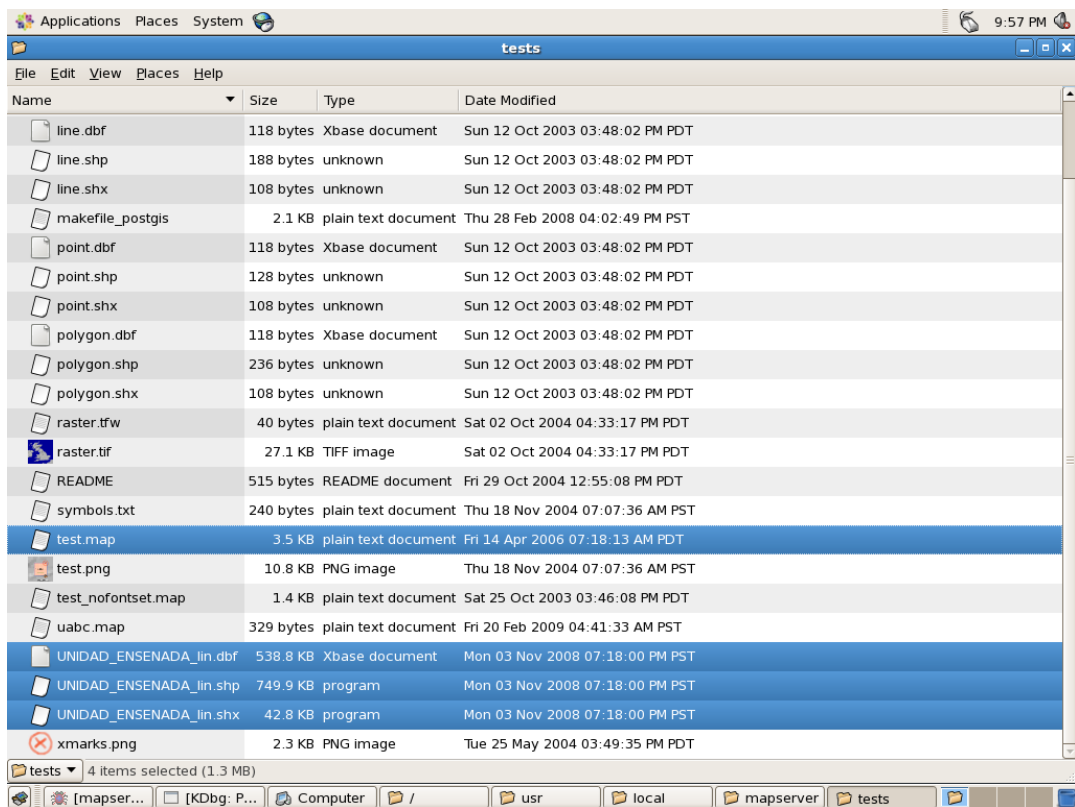


Figura 9.8. Ubicación de los valores de prueba

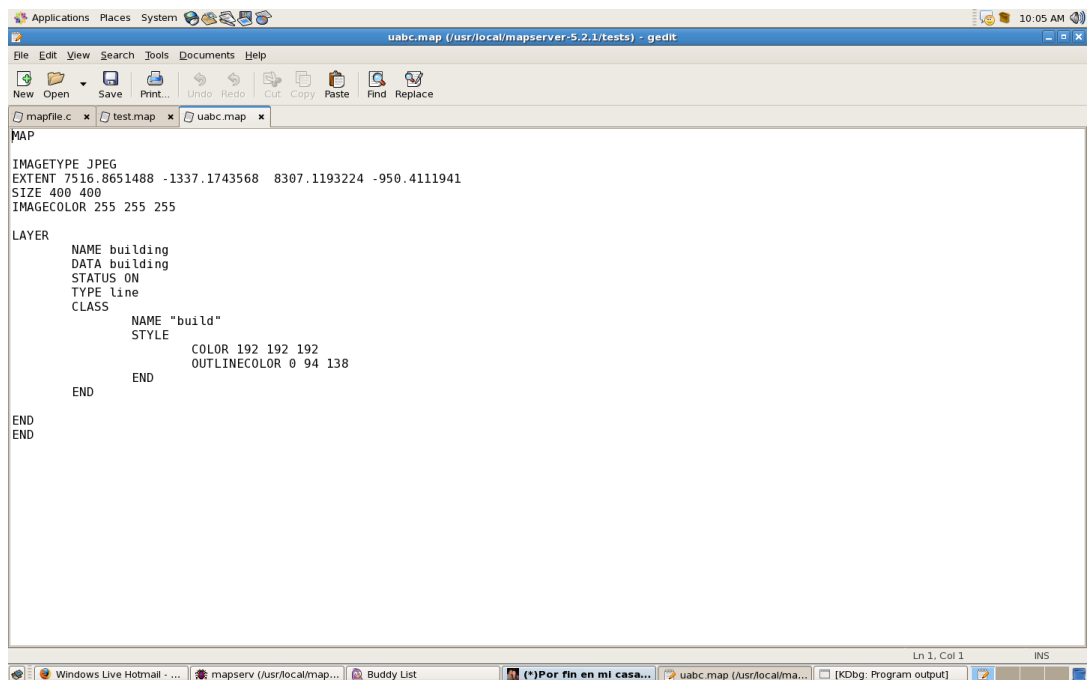


Figura 9.9. Archivo Mapfile contiene los valores de prueba

Paso 7: Watch, breakpoint y depuración

Una parte fundamental para comprender el funcionamiento y el flujo de datos interno del servidor de mapas UMN MapServer en el entorno gráfico KDbg, es por medio de Watch y breakpoint (Ver Figura 9.10).

- Watch: sirve para observar los valores que obtienen las variables en tiempo de ejecución.

Pasos:

1. Introducir el nombre de la variable
2. Presionar el botón Add
3. Visualizar el contenido de la variable

- Breakpoint: detiene la ejecución en cierta línea de código con la finalidad de revisar paso a paso su funcionamiento.

Pasos:

1. Posicionar el cursor sobre la línea de código que deseas detener la ejecución
2. Presionar el botón stop o el botón Add Breakpoint
3. Visualizar o ubicar rápidamente un breakpoint

Para llevar a cabo la depuración de un código puede hacerse en dos modos, presionando F5 o un icono con una hoja y una flecha roja apuntando hacia abajo que se encuentra en la barra de controles (parte superior de la pantalla). Si la depuración del programa se detuvo debido a un breakpoint y se desea continuar depuración paso a paso debe presionarse los iconos ubicados también en la barra de controles.

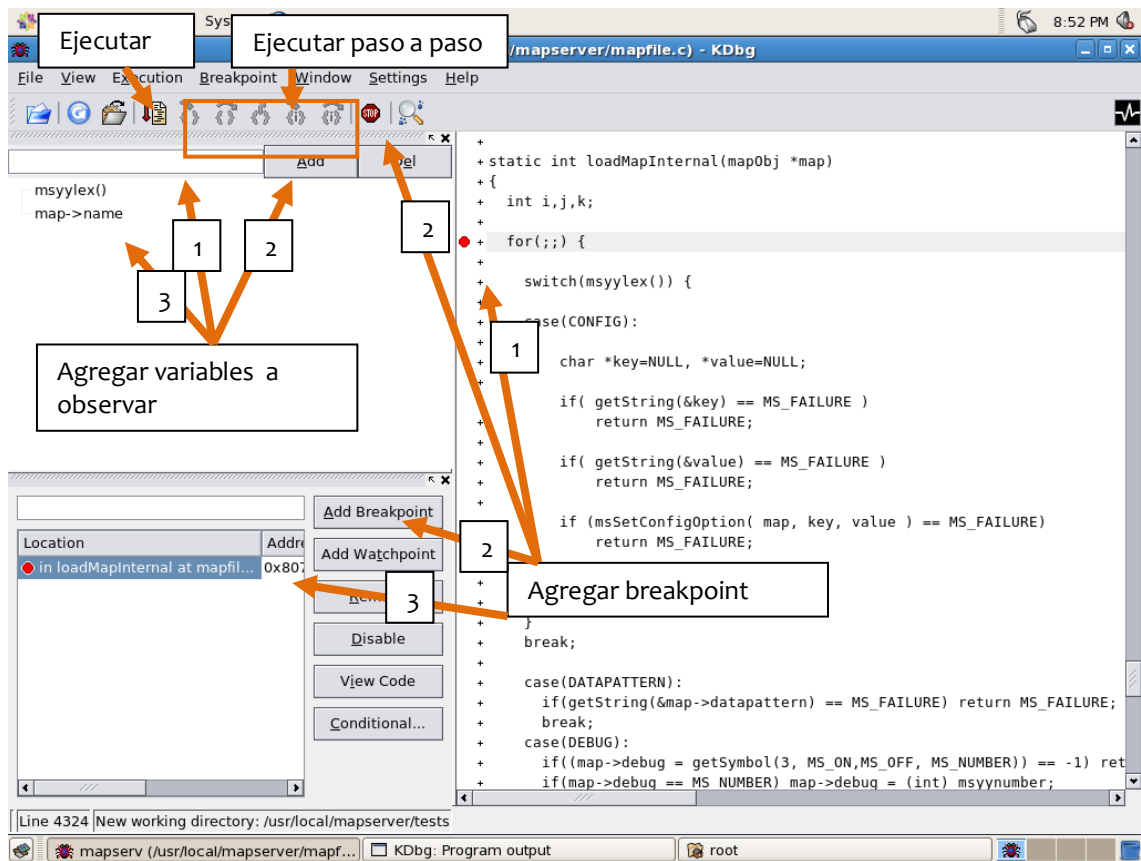


Figura 9.10. Agregar watch y breakpoint en KDbg

Paso 8: Programación

Lamentablemente la única forma de programar fue a partir de un editor de textos llamado nano; este es un editor que se utiliza línea de comandos o terminal (Ver Figura 9.11). El modo de uso es el siguiente:

```
nano + nombre del archivo  
nano mapfile.c
```

Nota: Cabe señalar que se utilizar el gedit el cual es un editor en un entorno gráfico

Una vez realizadas la modificaciones en el archivo mapfile.c debe presionarse CTRL + O para guardar cambios.



```
Applications Places System root@localhost:/usr/local/mapserver
File Edit View Terminal Tabs Help
GNU nano 1.3.12 File: mapfile.c

[
msSetError(MS_SYMERR, "Parsing error near (%s):(line %d)", "getDouble()", msyytext, msyylineno);
return(-1);
}

/*
** Load a integer from the map file. (see lexer.l)
*/
int getInteger(int *i) {
    if(msyylex() == MS_NUMBER) {
        *i = (int)msyynumber;
        return(0); /* success */
    }

    msSetError(MS_SYMERR, "Parsing error near (%s):(line %d)", "getInteger()", msyytext, msyylineno);
    return(-1);
}

int getCharacter(char *c) {
    if(msyylex() == MS_STRING) {
        *c = msyytext[0];
        return(0);
    }

    msSetError(MS_SYMERR, "Parsing error near (%s):(line %d)", "getCharacter()", msyytext, msyylineno);
    return(-1);
}

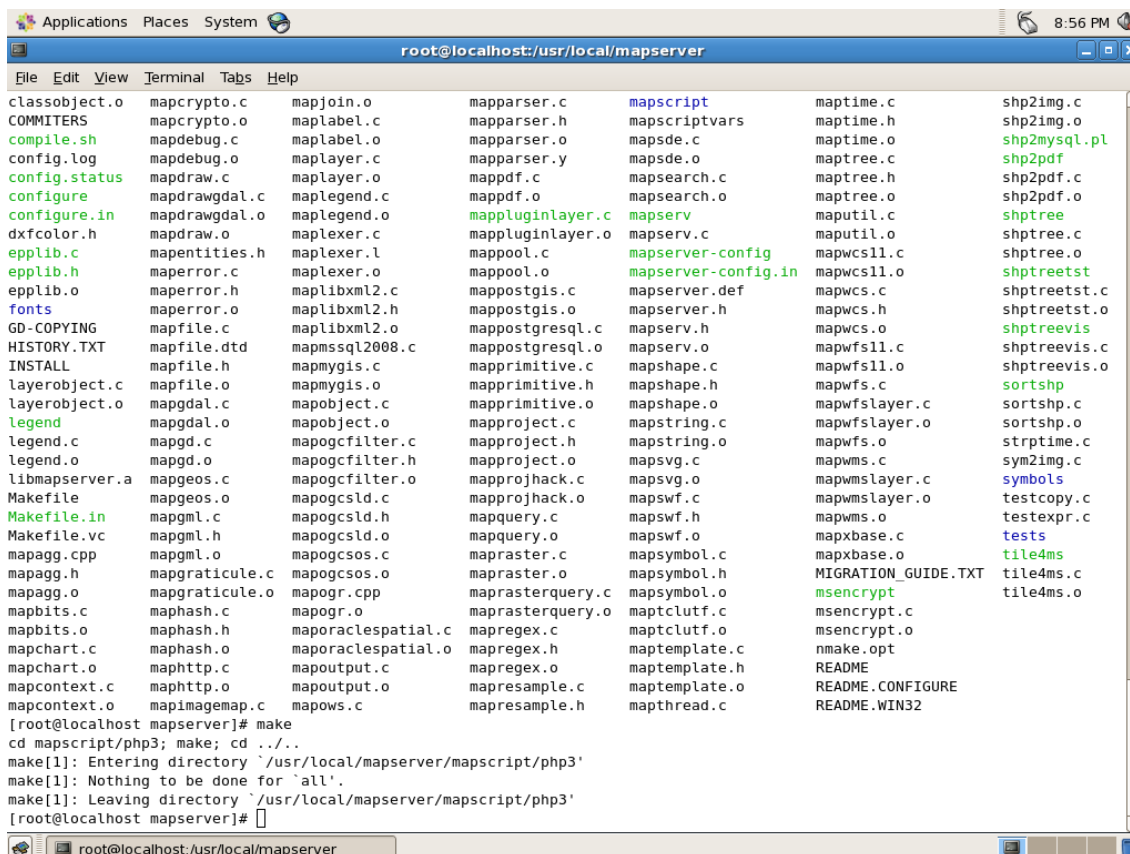
/*
** Try to load as an integer, then try as a named symbol.
** Part of work on bug 490.
*/
int getIntegerOrSymbol(int *i, int n, ...)
{
    ^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
    ^X Exit          ^J Justify       ^W Where Is      ^V Next Page     ^U UnCut Text    ^T To Spell

[mapserv (/usr/local/mapserver/mapfile.c) - KDbg] root@localhost:/usr/local/mapserver
```

Figura 9.11. Línea de comandos de CentOS 5

Paso 9: Compilación

La compilación se lleva a cabo por línea de comandos o terminal, este proceso sirve para generar de nueva cuenta el ejecutable mapserv, con las últimas modificaciones que se realizaron en los archivos de UMN MapServer. El modo de compilarlo es por medio del comando MAKE (Ver Figura 9.12)



```
root@localhost:usr/local/mapserver
File Edit View Terminal Tabs Help

classobject.o  mapcrypto.c  mapjoin.o      mapparser.c   mapsript      mvertime.c    shp2img.c
COMMITTERS    mapcrypto.o  maplabel.c    mapparser.h   mapsriptvars  mvertime.h    shp2img.o
compile.sh    mapdebug.c  maplabel.o    mapparser.o   mapsde.c      mvertime.o    shp2mysql.pl
config.log    mapdebug.o  maplayer.c    mapparser.y   mapsde.o      mptree.c      shp2pdf
config.status mapdraw.c   maplayer.o    mappdf.c      mapsearch.c   mptree.h      shp2pdf.c
configure     mapdrawdal.c maplegend.c   mappdf.o      mapsearch.o   mptree.o      shp2pdf.o
configure.in  mapdrawgdal.o maplegend.o   mappluginlayer.c mapserv      maputil.c     shptree
dxfcolor.h    mapdraw.o   maplexer.c    mappluginlayer.o mapserv.c     maputil.o     shptree.c
epplib.c      mapentities.h maplexer.l    mappool.c     mapserver-config mapwcs11.c    shptree.o
epplib.h      maperror.c  maplexer.o    mappool.o     mapserver-config.in mapwcs11.o    shptreetst
epplib.o      maperror.h  maplibxml2.c  mappostgis.c  mapserver.def  mapwcs.c      shptreetst.c
fonts         maperror.o  maplibxml2.h  mappostgis.o  mapserver.h    mapwcs.h      shptreetst.o
GD - COPYING  mapfile.c   maplibxml2.o  mappostgresql.c mapserv.h     mapwcs.o      shptreetvis
HISTORY.TXT   mapfile.dtd mapmssql2008.c mappostgresql.o mapserv.o     mapwfs11.c    shptreevis.c
INSTALL       mapfile.h   mapmygis.c    mapprimitive.c mapshape.c    mapwfs11.o    sym2img.c
layerobject.c mapfile.o   mapmygis.o    mapprimitive.h mapshape.h    mapwfs.c      sortshp
layerobject.o mapgdal.c   mapobject.c   mapprimitive.o mapshape.o    mapwfs11.c    sortshp.c
legend        mapgdal.o   mapobject.o   mapproject.c  mapstring.c   mapwfs11.o    sortshp.o
legend.c      mapgd.c     mapogcfilter.c mapproject.h  mapstring.o   mapwfs.o      strptime.c
legend.o      mapgd.o     mapogcfilter.h mapproject.o  mapstring.o   mapwms.c      sym2img.c
libmapserver.a mapgeos.c   mapogcfilter.o mapprojhack.c mapsvg.c      mapwms11.c    symbols
Makefile      mapgeos.o   mapogcsld.c  mapprojhack.o mapsvg.o      mapwms11.o    testcopy.c
Makefile.in   mapgml.c    mapogcsld.h  mapquery.c    mapswf.c     mapwms.o      testexpr.c
Makefile.vc   mapgml.h    mapogcsld.o  mapquery.o    mapswf.h     mapwms.o      tests
mapagg.cpp    mapgml.o    mapogcsos.c  mapraster.c   mapswf.o     mapxbase.c    tile4ms
mapagg.h      mapgraticule.c mapogcsos.o  mapraster.o   mapsymbol.c  mapxbase.o    tile4ms.c
mapagg.o      mapgraticule.o mapogr.cpp    maprasterquery.c mapsymbol.h   MIGRATION_GUIDE.TXT tile4ms.o
mapbits.c     maphash.c   mapogr.o     maprasterquery.o mapsymbol.o   msencrypt     tile4ms.o
mapbits.o     maphash.h   maporaclespatial.c mapregex.c   msencrypt.c  msencrypt.o
mapchart.c    maphash.o   maporaclespatial.o mapregex.h   msencrypt.o  nmake.opt
mapchart.o    maphttp.c   mapoutput.c   mapregex.o   msencrypt.o  README
mapcontext.c  maphttp.o   mapoutput.o   mapresample.c maptemplate.h msencrypt.o   README
mapcontext.o  mapimagemap.c mapows.c     mapresample.h mapthread.c  msencrypt.o   README.WIN32

[root@localhost mapserver]# make
cd mapsript/php3; make; cd ../..
make[1]: Entering directory `/usr/local/mapserver/mapsript/php3'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/usr/local/mapserver/mapsript/php3'
[root@localhost mapserver]#
```

Figura 9.12. Compilación de UMN MapServer por línea de comandos

Resumen

El proceso de desarrollo para el servidor de mapas UMN MapServer , se resume con el siguiente diagrama:



Figura 9.13. Diagrama para el desarrollo de código en el servidor de mapas UMN MapServer