

UNIVERSIDAD SAN FRANCISCO DE QUITO

Colegio de Postgrados

Diseño e implementación de Mapa Interactivo utilizando Web Mapping y Base de Datos Espacial: Ciudad de Quevedo

Orlando Ramiro Erazo Moreta

Tesis de grado presentada como requisito
para la obtención del título de
Maestría en Sistemas de Información Geográfica

Quito, julio de 2009

HOJA DE APROBACION

© Derechos de autor

Orlando Ramiro Erazo Moreta

2009

RESUMEN

En Ecuador, especialmente en los últimos años, el uso de Sistemas de Información Geográfica y la publicación de cartografía en Internet (Web Mapping) han tenido una mayor acogida, aunque en varias ciudades aún se desconoce o no se aprovechan completamente sus bondades, lo que ha llevado a la creación de un mapa interactivo de la ciudad de Quevedo.

El software desarrollado consiste en un sitio Web que permite visualizar un mapa del cantón Quevedo con su información básica, la misma que comprende: división político-administrativa, red de calles, recorridos de buses urbanos y lugares de interés (alojamientos, bancos, educativos, públicos, etc.); acompañado de la posibilidad de realizar búsquedas de sitios, intersecciones de calles y rutas más cortas (peatonal y vehicular) entre dos puntos, empleando para esto último el algoritmo de Dijkstra que se determinó es el más adecuado para la red utilizada. Cabe destacar que la aplicación ha sido implementada haciendo uso de herramientas Open Source.

Toda la información adquirida ha sido integrada en una base de datos espacial, diseñada pensando que en el futuro se incorporarán al mapa opciones como rutas más rápidas y áreas de captación, y además, para que se constituya en una fuente de datos para otros trabajos.

ABSTRACT

In Ecuador, during the last years specially, the use of the Geographic Information Systems and the Internet cartography publication (Web Mapping) have been considerably increasing, although in many cities their advantages or benefits are either unknown or not fully exploited, so these facts have lead to the creation of an interactive map of Quevedo city.

The developed software consists on a Web site that allows us to visualize a Quevevo map with its basic information: political-administrative divisions, streets maps, buses routes and public interest places (lodging, banks, educational institutions, etc.). The software also features the possibility to make searches of specific places, street intersections, the shortest routes (both pedestrian and vehicular) between 2 points, using for this the Dijkstra algorithm, which was determined to be the most useful for the employed network. It's important to remark that the application has been improved by the use of Open Source tools.

All the acquired information has been integrated into a spatial database, which was designed to include in the future more features such as faster routes and service areas, and also to become a data source for other works.

TABLA DE CONTENIDO

	Pág.
RESUMEN	iii
ABSTRACT	iv
TABLA DE CONTENIDO	v
LISTA DE TABLAS	vi
LISTA DE FIGURAS	vi
1. INTRODUCCION	1
1.1. PRESENTACION	1
1.2. OBJETIVOS	3
1.2.1. Objetivo general	3
1.2.2. Objetivos específicos	4
2. REVISIÓN DE LITERATURA	5
2.1. LOS SIG Y EL ANALISIS DE REDES	5
2.1.1. Sistemas de Información Geográfica	5
2.1.2. Análisis de redes	7
2.2. WEB MAPPING	9
2.2.1. Definición	10
2.2.2. Componentes de una aplicación Web Mapping	11
2.2.2.1. El cliente	13
2.2.2.2. Servidor Web	15
2.2.2.3. Servidor de Mapas	17
2.2.2.4. Los datos	21
2.3. BASES DE DATOS ESPACIALES	21

2.3.1. Bases de datos y DBMS	22
2.3.2. Base de datos y DBMS espaciales	23
2.3.3. PostgreSQL y PostGIS	24
3. MATERIALES	26
3.1. EQUIPOS	26
3.1.1. Computador	26
3.1.2. GPS	27
3.2. SOFTWARE	27
3.3. DATOS	28
4. METODOLOGIA	30
4.1. DESARROLLO DE SOFTWARE	30
4.1.1. Modelo de Proceso	31
4.1.2. Casos de uso	33
4.1.3. Lenguaje de modelado	34
4.2. ADQUISICION DE DATOS	35
5. IMPLEMENTACION	37
5.1. DOCUMENTO DE REQUERIMIENTOS	37
5.2. DIAGRAMA DE CASOS DE USO	39
5.3. CASO DE USO MOSTRAR MAPA INTERACTIVO DE QUEVEDO	40
5.4. CASO DE USO NAVEGAR EN EL MAPA	43
5.5. CASO DE USO MOSTRAR/OCULTAR CAPA	46
5.6. CASO DE USO BUSCAR RUTA	48
5.7. CASO DE USO UBICAR SITIO	51
5.8. CASO DE USO BUSCAR INTERSECCIÓN DE CALLES	54

5.9. CASO DE USO MOSTRAR RECORRIDO DE BUS	57
5.10. CASO DE USO MOSTRAR INFORMACIÓN DE SITIO	60
5.11. CASO DE USO IMPRIMIR MAPA	63
5.12. CASO DE USO RESTAURAR MAPA	66
5.13. DIAGRAMA DE CLASES	68
5.14. DISEÑO DE LA BASE DE DATOS	69
5.15. DIAGRAMA DE COMPONENTES	70
5.16. DIAGRAMA DE DESPLIEGUE	71
6. RESULTADOS	72
6.1. MOSTRAR MAPA INTERACTIVO DE QUEVEDO	74
6.2. NAVEGAR EN EL MAPA	75
6.3. MOSTRAR/OCULTAR CAPA	76
6.4. BUSCAR RUTA	77
6.5. UBICAR SITIO	81
6.6. BUSCAR INTERSECCIÓN DE CALLES	82
6.7. MOSTRAR RECORRIDO DE BUS	83
6.8. MOSTRAR INFORMACIÓN DE SITIO	84
6.9. IMPRIMIR MAPA	85
6.10. RESTAURAR MAPA	86
7. CONCLUSIONES	87
8. RECOMENDACIONES	90
BIBLIOGRAFIA	92
ANEXO A. MANUAL DE INSTALACION	96
A.1. REQUERIMIENTOS DEL SISTEMA	96

A.2. INSTALACIÓN	97
A.3. CREACIÓN DE LA BASE DE DATOS	99
ANEXO B. ARCHIVO DE MAPA miq.map	107
ANEXO C. CONTENIDO DEL CD	133

LISTA DE TABLAS

	Pág.
Tabla 5.1. Documentos de Requerimientos para el Mapa Interactivo de la ciudad de Quevedo.	38
Tabla 6.1. Comparación de algoritmos para encontrar la ruta óptima para varios pares de puntos (lugares).	79

LISTA DE FIGURAS

	Pág.
Figura 2.1. Ruta más corta entre dos puntos de la ciudad de Quevedo.	8
Figura 2.2. Componentes de una aplicación Web Mapping.	12
Figura 2.3. Componentes de MapServer.	19
Figura 2.4. Ejemplo de estructura de un archivo de mapa.	20
Figura 4.1. Fases e hitos del modelo de proceso MSF.	32
Figura 5.1. Diagrama de casos de uso para el Mapa Interactivo de la ciudad de Quevedo.	39
Figura 5.2. Diagrama de secuencia mostrar mapa interactivo de Quevedo.	42
Figura 5.3. Herramientas para navegar en el mapa.	44
Figura 5.4. Diagrama de secuencia navegar en el mapa.	45
Figura 5.5. Lista de capas disponibles.	47
Figura 5.6. Diagrama de secuencia mostrar/ocultar capa.	47

Figura 5.7. Agregar localización de red y parámetros para buscar ruta.	50
Figura 5.8. Diagrama de secuencia buscar ruta.	50
Figura 5.9. Parámetros para ubicar sitio.	52
Figura 5.10. Diagrama de secuencia ubicar sitio.	53
Figura 5.11. Parámetros para buscar intersección de calles.	55
Figura 5.12. Diagrama de secuencia buscar intersección de calles.	56
Figura 5.13. Parámetros para mostrar recorrido de bus.	58
Figura 5.14. Diagrama de secuencia mostrar recorrido de bus.	59
Figura 5.15. Capa activa, identificar sitio y mostrar su información.	61
Figura 5.16. Diagrama de secuencia mostrar información de sitio.	62
Figura 5.17. Imprimir el mapa.	64
Figura 5.18. Diagrama de secuencia imprimir mapa.	65
Figura 5.19. Restaurar el mapa.	67
Figura 5.20. Diagrama de secuencia restaurar mapa.	67
Figura 5.21. Diagrama de clases para el Mapa Interactivo de la ciudad de Quevedo.	68
Figura 5.22. Diseño de la base de datos para el Mapa Interactivo de la ciudad de Quevedo.	69
Figura 5.23. Diagrama de componentes para el Mapa Interactivo de la ciudad de Quevedo.	70
Figura 5.24. Diagrama de despliegue para el Mapa Interactivo de la ciudad de Quevedo.	71
Figura 6.1. Página principal del Mapa Interactivo de la ciudad de Quevedo.	75

Figura 6.2. Centro de la ciudad de Quevedo visualizado usando la herramienta acercar con rectángulo.	76
Figura 6.3. Centro de la ciudad de Quevedo con las capas alojamiento y bancos visible y ocultando las parroquias.	77
Figura 6.4. Ruta más corta a pie de la Asociación de Discapacitados al IESS.	78
Figura 6.5. Ruta más corta en carro de la Asociación de Discapacitados al IESS.	78
Figura 6.6. Ruta más corta a pie de la Asociación de Discapacitados al IESS utilizando ArcGIS.	80
Figura 6.7. Ruta más corta en carro de la Asociación de Discapacitados al IESS utilizando ArcGIS.	81
Figura 6.8. Ubicación en el mapa del Terminal Terrestre de Quevedo.	82
Figura 6.9. Intersección de calles 7 de Octubre y Quinta.	83
Figura 6.10. Recorrido de buses urbanos (Línea 1 ejecutivo).	84
Figura 6.11. Visualización de información del Parque de la Confraternidad.	85
Figura 6.12. Vista preliminar del Parque Central y sus alrededores.	86

1. INTRODUCCION

1.1. PRESENTACION

Durante muchos años, los trabajos cartográficos han estado limitados a la generación de mapas en papel, que debido a su naturaleza estática, no brindaban las suficientes facilidades para la navegación y consultas. Con la aparición de los computadores personales, esta situación fue cambiando poco a poco; es así como a medida que estos evolucionaban y disponían de mayores capacidades de procesamiento, se desarrollaron programas que permitían utilizarlos tanto para el almacenamiento como para la visualización y reproducción en forma impresa o digital de la información geográfica.

Uno de estos tipos de programas son los Sistemas de Información Geográfica (SIG), que con sus bondades para la gestión y análisis de la información espacial, se han constituido en una herramienta de gran importancia, no sólo para cartógrafos y geógrafos, sino también para profesionales de otras áreas como ordenamiento territorial, gestión catastral, análisis de mercados, medio ambiente, turismo, entre otras.

Aunque inicialmente estuvieron limitados a aplicaciones de escritorio, con los avances tecnológicos en el campo de las redes de computadores, se ha logrado que los SIG puedan funcionar en forma distribuida. Esto ha permitido que los SIG y la publicación de mapas sean llevados al Internet, siendo cada vez más común escuchar los términos Web Mapping o SIG Web.

En paralelo a estos avances, y teniendo en cuenta el alto costo de los paquetes comerciales, surgió una interesante alternativa como lo es el Open Source (código fuente abierto, software libre), cuyo potencial se ha visto desarrollado especialmente en los últimos años.

Esta tendencia del Open Source ha sido acogida favorablemente por diversas instituciones, considerando notable mencionar el caso del gobierno ecuatoriano que, mediante decreto No. 1014 del 10 de abril del 2008, autorizó la utilización de Software Libre en las Entidades de la Administración Pública del país.

El concepto de Open Source ha sido aplicado en las distintas áreas del software, incluyendo la geoespacial, existiendo diversas herramientas, que utilizadas en forma independiente o conjunta, permiten generar atractivas soluciones. El GeoPortal del Instituto Geográfico Militar del Ecuador (<http://www.geoportaligm.gov.ec>), Open Street Map (<http://openstreetmap.com>), Ride the City (<http://www.ridethecity.com>), son sólo unos cuantos ejemplos de la gran cantidad de aplicaciones desarrolladas utilizando software libre, en la categoría Web Mapping.

Aprovechando las características de este tipo de software, se ha elaborado un mapa interactivo para la ciudad de Quevedo, el cual incluye, además de la información base del cantón, aquellos lugares que se consideran pueden ser de especial interés (bancos, hoteles, parques, etc.) para diferentes tipos de usuarios. Este mapa está acompañado de la respectiva red de calles del cantón, brindando además la posibilidad de realizar búsquedas, no sólo de los sitios incluidos, sino

también de la ruta más corta que se debe seguir para desplazarse desde un punto a otro.

Además, ha sido creada una base de datos espacial del cantón, incorporando en ella los datos tanto alfanuméricos como geográficos que fueron puestos a disposición y/o actualizados, y aquellos que fueron levantados por el autor, constituyéndose en una importante fuente de información para la elaboración de futuros proyectos similares en el área mencionada.

El acceso y consultas a este mapa se realizan a través del Internet, por lo cual, fue necesaria la combinación de herramientas como servidor de mapas, sistema de gestión de base de datos con capacidad para manejo de datos espaciales, lenguajes de programación.

1.2. OBJETIVOS

1.2.1. Objetivo general

Diseñar e implementar un mapa interactivo de la ciudad de Quevedo, que permita visualizar y consultar sitios de interés general y su red de calles, utilizando Web Mapping y base de datos espacial.

1.2.2. Objetivos específicos

- Desarrollar una aplicación Web para la visualización del mapa del cantón Quevedo, incluyendo las opciones necesarias para la navegación y búsquedas.
- Generar e implementar una base de datos espacial de los sitios considerados de interés general del cantón Quevedo, para utilizarla en la elaboración del mapa interactivo del cantón y como fuente de información para futuros trabajos en el área.
- Incorporar la red de calles de la ciudad en la base de datos generada, permitiendo realizar búsquedas de vías y rutas.
- Utilizar herramientas de tipo Open Source para la implementación del mapa interactivo de Quevedo.

2. REVISIÓN DE LITERATURA

2.1. LOS SIG Y EL ANALISIS DE REDES

Gracias a los avances tecnológicos logrados, especialmente en cuanto al hardware y software de computadores se refiere, los Sistemas de Información Geográfica (SIG) han experimentado un rápido desarrollo, ofreciendo métodos muy innovadores para la gestión y análisis de la información espacial.

Con el paso del tiempo, los SIG se han implementado en diversos tipos de organizaciones y continúan en expansión, siendo utilizados por profesionales de diversas áreas del conocimiento.

2.1.1. Sistemas de Información Geográfica

Un SIG es un sistema de información diseñado para trabajar con datos geográficos; es decir, la combinación de hardware, software, datos, personal y procedimientos para capturar, almacenar, manipular, analizar, modelar y presentar datos referenciados en el espacio para la solución de problemas¹.

Los SIG trabajan con dos modelos fundamentales de información geográfica: raster y vector. En el modelo vector la información se representa en forma de puntos, líneas y polígonos, como por ejemplo hoteles, calles, parroquias de una

¹ STROBL, J.; RESL, R. (2005): Orientación e Introducción en Sistemas de Información Geográfica. UNIGIS Professional para América Latina, pp. 67-70 y pp. 81-93.

ciudad, respectivamente. En el modelo raster, el espacio es considerado como un grid o matriz, compuesto de una serie de elementos discretos o celdas de igual tamaño, resultando más útil para la descripción de características de variación continua como podría ser los tipos o usos de suelos.

En forma general, algunas de las funciones de los SIG son: ingreso, transformación y gestión de datos; realización de consultas alfanuméricas y geográficas; análisis de superposición, proximidad, conectividad; visualización.

Tal como señalan Gutiérrez y Gould², los SIG son herramientas multipropósito, con diversos campos de aplicación, entre los que mencionan:

- Medio ambiente y recursos naturales: aplicaciones forestales, cambios en los usos del suelo, estudios de impacto ambiental, localización de vertederos.
- Catastro.
- Transporte: mantenimiento y conservación de infraestructuras de transporte, trazado de infraestructuras lineales, impacto territorial de las nuevas infraestructuras, sistemas de navegación para automóviles.
- Redes de infraestructuras básicas.
- Protección civil: riesgos, desastres, catástrofes.
- Análisis de mercados.
- Planificación urbana.

² GUTIERREZ, J.; GOULD, M. (2000): SIG: Sistemas de Información Geográfica. Madrid, Síntesis, pp. 225-234.

2.1.2. Análisis de redes

Una interesante utilidad que se le puede dar a los SIG está en el análisis de redes. “Una red es un sistema interconectado de elementos lineales, que forman una estructura espacial por la que pueden pasar flujos de algún tipo: personas, mercancías, energía, información”³. Redes de transporte, hidrográficas, eléctricas, de abastecimiento de agua, etc. son algunos ejemplos.

Las redes se representan mediante nodos y arcos enlazados, y se denomina grafo⁴. Los grafos son objeto de estudio de la teoría de grafos como parte de las Matemáticas Discretas. Las definiciones formales concernientes a la teoría de grafos pueden ser revisadas en textos de matemática discreta⁵.

Informalmente, los grafos son un conjunto de vértices o nodos unidos por aristas o arcos. En un grafo, los arcos pueden tener una dirección establecida, denominándose grafos dirigidos. Los arcos también pueden tener asociado un valor que representa un costo o peso, caso en el cual se hace referencia al término grafo ponderado. Si se necesita realizar búsquedas en el grafo se puede aplicar algún algoritmo de búsqueda.

Haciendo una analogía con la red de calles de una ciudad tenemos: Los arcos son la representación de las calles, que se pueden recorrer en uno o dos sentidos, y tienen asociado un peso o costo correspondiente a la distancia o el

³ BOSQUE, J. (1992): Sistemas de Información Geográfica, Primera edición. Madrid, p. 207.

⁴ STROBL, J.; RESL, R. (2000): Fundamentos: Modelos y Estructuras de Datos espaciales. UNIGIS Professional para América Latina, p. 63.

⁵ COMELLAS, F. et al. (2001): Matemática Discreta. Barcelona, Ediciones UPC, 336 p.

tiempo en atravesarlos; los nodos constituyen las intersecciones de las calles; finalmente, una persona puede desear conocer la mejor ruta (más corta o más eficiente) que debe seguir para ir de un lugar de la ciudad a otro, que puede obtenerse aplicando un algoritmo de caminos mínimos. La figura 2.1 muestra una parte de la red de calles de la ciudad de Quevedo, incluyendo también la ruta más corta entre dos puntos.

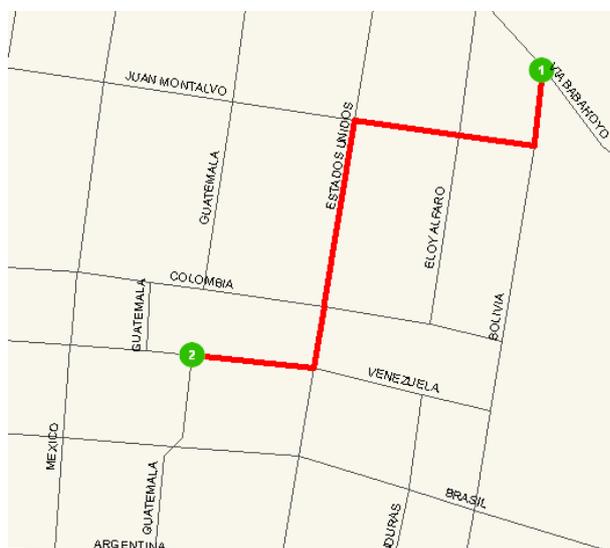


Figura 2.1. Ruta más corta entre dos puntos de la ciudad de Quevedo.

“Las rutas son objetos complejos formados por segmentos de líneas de un sistema de redes, en la mayoría de una red vial, como por ejemplo las rutas de buses”⁶.

El análisis de caminos mínimos (routing) junto con el análisis de áreas de influencia (allocate), constituyen las funcionalidades más populares dentro del análisis de redes. Una opción que puede incluir el primero es la de efectuar

⁶ STROBL, J.; RESL, R. (2000): Fundamentos: Modelos y Estructuras de Datos espaciales. UNIGIS Professional para América Latina, p. 73.

paradas intermedias, debiendo el sistema proporcionar el orden óptimo en que deben realizarse así como también la ruta⁷. Dos algoritmos muy comunes en el análisis de caminos mínimos son el de Dijkstra y A* (A star).

2.2. WEB MAPPING

En los últimos años, el Internet ha evolucionado notablemente llegando a convertirse en un medio de uso masivo a nivel mundial, gracias a las facilidades que ofrece para el intercambio de información.

Muchas de las aplicaciones que antes estaban limitadas a ser usadas en un computador en forma local han sido llevadas a esta red de redes, sin que las concernientes a la Cartografía y los Sistemas de Información Geográfica sean la excepción, pues desde 1993 en que Steve Putz publicó el primer mapa basado en el Web⁸, se siguieron generando una serie de técnicas y software que permiten visualizar, consultar y/o editar mapas en la red. Esto es lo que se conoce como Web Mapping, aunque también se suele hablar de Web GIS y Web Cartography.

El Web Mapping ha tenido una gran acogida, por lo que en la actualidad se puede encontrar una gran cantidad de sitios Web en los que se incluyen mapas de diversos tipos y lugares.

⁷ GUTIERREZ, J.; GOULD, M. (2000): SIG: Sistemas de Información Geográfica. Madrid, Síntesis, pp. 185-186.

⁸ LONGLEY, P. et al. (2001): Geographic Information Systems And Science. John Wiley & Sons, pp.13-17.

2.2.1. Definición

En forma general, Web Mapping es el término utilizado para referirse a la visualización de datos espaciales a través del Internet. En la Enciclopedia de SIG, Neumann⁹ escribe que el Web Mapping es “el proceso de diseñar, implementar, generar y entregar mapas en el World Wide Web”. Neumann continúa haciendo una distinción entre Web Mapping y Web Cartography, comentando que la segunda adicionalmente estudia aspectos teóricos relacionados con los mapas. Además, hace referencia al término Web GIS, que está relacionado con el Web Mapping, y aunque no significan lo mismo, muchas veces se utilizan como sinónimos (ver también ¹⁰ y ¹¹).

Las aplicaciones de Web Mapping permiten mostrar los mapas en Internet no sólo de forma estática (por ejemplo, como una carta topográfica escaneada) sino también hacerlos dinámicos, permitiendo así la interacción del usuario. Entre sus principales funcionalidades están:

- Generación mapas con todos los elementos requeridos incluyendo herramientas para la navegación (acercar, alejar, etc.).
- Superposición de capas de información.
- Despliegue de información descriptiva de los elementos del mapa.
- Ejecución de consultas de tipo espacial.

⁹ NEUMANN, A. (2008): Web Mapping and Web Cartography. En S. Shekar y H. Xiong (Eds.), Encyclopedia of GIS. Springer, pp. 1261-1262.

¹⁰ GREEN, D.; BOSSOMAIER, T. (2002): Online GIS and Spatial Metadata. Londres, Taylor & Francis, pp. 13-26.

¹¹ KRAAK, M.; BROWN, A. (2001): Web Cartography. Londres, Taylor & Francis, pp. 1-37.

- Interacción con bases de datos.

El Web Mapping brinda algunas ventajas como la posibilidad de compartir e intercambiar información con un público mucho más amplio, acceso a herramientas para análisis y toma de decisiones, actualización continua de información, facilidad para actualizar las aplicaciones, entre otras.

Pero el Web Mapping aún presenta ciertos problemas, como los mencionados por Neumann¹²: la realidad del Internet y la infraestructura de los servidores no es lo suficientemente buena todavía, la complejidad para desarrollar mapas basados en el Web, la limitación del espacio de la pantalla, entre otros. Una forma de contrarrestar estos inconvenientes, el primero principalmente, es mejorar u optimizar los recursos; para ello, en el artículo publicado por la empresa Geograma (<http://www.geograma.com/>), "Optimización de un servidor de mapas en Internet"¹³, se proponen algunas técnicas que se podrían aplicar.

2.2.2. Componentes de una aplicación Web Mapping

Para el desarrollo de una aplicación de Web Mapping se pueden recurrir a diversas alternativas, ya sea con software propietario o con software libre. Dependiendo de cuál sea la elección, el funcionamiento interno de su aplicación

¹² NEUMANN, A. (2008): Web Mapping and Web Cartography. En S. Shekar y H. Xiong (Eds.), Encyclopedia of GIS. Springer, pp. 1263-1264.

¹³ GEOGRAMA (2004): Optimización de un servidor de mapas en Internet. Disponible en: <http://www.cartesia.org/>.

puede variar, pero, tal como indica Mitchell¹⁴, existen algunos conceptos generales en común, los cuales se ilustran en la figura 2.2.

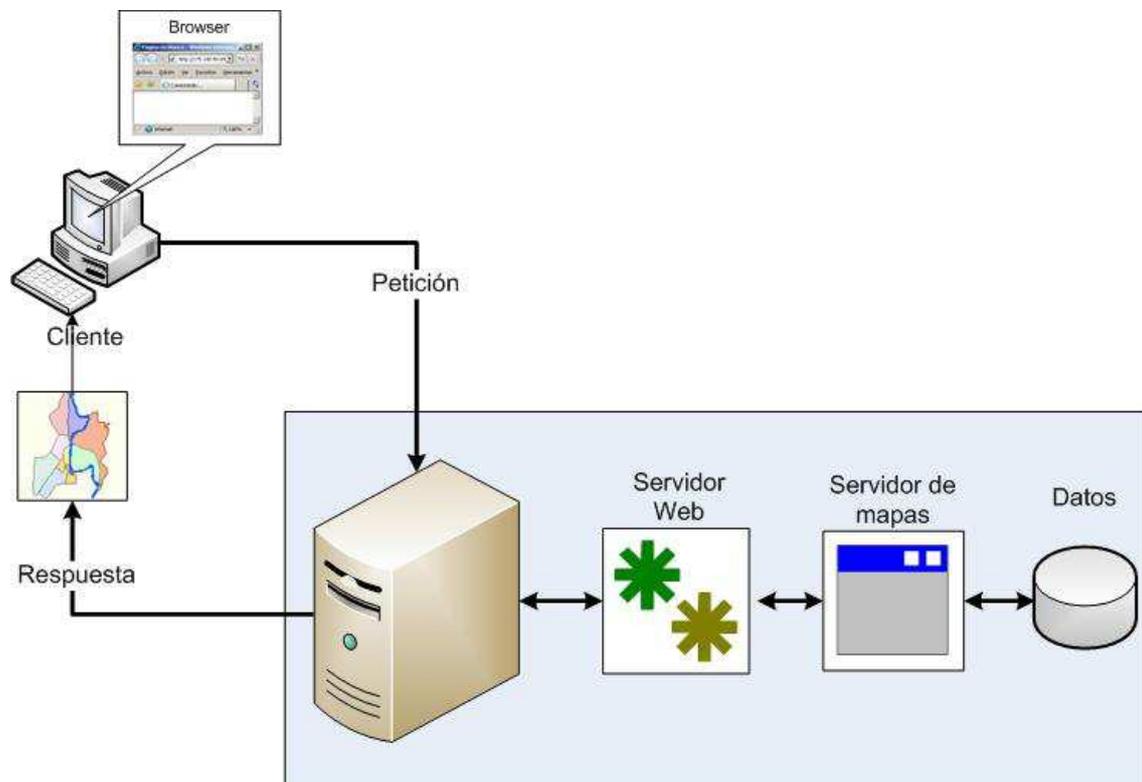


Figura 2.2. Componentes de una aplicación Web Mapping.

Primeramente, el usuario (cliente), desde su computador, a través de su programa de navegación en Internet (browser o navegador), escribe la dirección correspondiente al sitio Web en que reside la aplicación, se envía su petición y esta es recibida por el servidor Web, que generará una página Web como respuesta. Para ello, el servidor Web debe comunicarse con el servidor de mapas que será el encargado de acceder a los datos espaciales y a partir de ellos generar el texto o las imágenes correspondientes al mapa del área solicitada y que se incluirán en la respuesta.

¹⁴ MITCHELL, T. (2005): Web Mapping Illustrated. O'Reilly, pp. 5-11.

Teniendo en cuenta esto, los componentes que generalmente se encuentran en una aplicación Web Mapping son (remitirse también a ¹⁵, ¹⁶, ¹⁷ y ¹⁸):

- El cliente.
- Un servidor Web.
- Un servidor de mapas.
- Los datos.

2.2.2.1. El cliente

Por lo general, el cliente es un browser; es decir, el programa que se utiliza para visualizar las páginas Web. El browser debe interpretar la información que recibe de un servidor Web y presentarla gráficamente al usuario, debiendo para ello interpretar código HTML (HyperText Markup Language).

El Lenguaje de Marcación de Hipertexto, es el principal lenguaje de descripción de páginas Web. Básicamente se trata de un conjunto de etiquetas que sirven para definir la forma en la cual presentar el texto y otros elementos de la página.

Para el aprendizaje del lenguaje HTML se sugiere ¹⁹ y ²⁰.

¹⁵ NEUMANN, A. (2008): Web Mapping and Web Cartography. En S. Shekar y H. Xiong (Eds.), Encyclopedia of GIS. Springer, pp. 1266-1267.

¹⁶ ESRI (2004): ArcIMS 9 Architecture and Functionality. ESRI White Paper. Disponible en: <http://www.esri.com>.

¹⁷ KRAAK, M.; BROWN, A. (2001): Web Cartography. Londres, Taylor & Francis, pp. 73-85.

¹⁸ GREEN, D.; BOSSOMAIER, T. (2002): Online GIS and Spatial Metadata. Londres, Taylor & Francis, pp. 27-44.

¹⁹ TITTEL, E.; BURMEISTER, M. (2005): HTML 4 For Dummies, Quinta edición. Wiley Publishing, 432 p.

²⁰ TITTEL, E.; NOBLE, J. (2008): HTML, XHTML & CSS For Dummies, Sexta edición. Wiley Publishing, 384 p.

Cuando se carga o visualiza una página en el browser, este también debe permitir la interacción con el usuario, sin limitarse a los enlaces entre páginas o hipervínculos, lo que llevó a la necesidad de mejorar el HTML originando el HTML Dinámico (DHTML). Esta mejora del HTML se basó en dos estándares, las Hojas de Estilo en Cascada (CSS, Cascading Style Sheets) que permiten separar la estructura de la presentación, y fundamentalmente en el Modelo de Objetos de Documento (DOM, Document Object Model) que puede entenderse como la forma en que los navegadores interpretan una página desprovista de comportamientos programables, transformando sus elementos en objetos que poseen propiedades, métodos y eventos, manejados mediante un lenguaje de programación (lenguaje de script), como por ejemplo, JavaScript²¹.

JavaScript es un lenguaje de programación que permite realizar acciones en el ámbito de una página Web, siendo el browser el encargado de interpretar las instrucciones. La Biblia de JavaScript²² es una interesante obra para iniciar y profundizar en el estudio de JavaScript.

La interactividad que brinda JavaScript se ha popularizado mucho en los últimos años, al punto que se pueden crear aplicaciones Web sin necesidad de recargas constantes de las páginas, ya que la comunicación con el servidor se produce en segundo plano. Esto se consigue utilizando la técnica denominada Ajax (Asynchronous JavaScript And XML). Garrett, quien acuñó el término por primera vez, lo define así: "Ajax no es una tecnología en sí mismo. En realidad, se trata de

²¹ POSADAS, M. (2000): HTML dinámico, modelos de objetos y JavaScript. Grupo EIDOS, pp. 9-14.

²² GOODMAN, D.; MORRISON, M. (2004): JavaScript Bible, Quinta edición. Wiley Publishing, 1272 p.

la unión de varias tecnologías, que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes²³. En Profesional Ajax²⁴ se pueden encontrar los fundamentos de Ajax así como también algunos ejemplos de su uso, como en el caso de Gmail y Google Maps.

2.2.2.2. Servidor Web

El servidor Web es el programa que hace posible la visualización de contenidos en el Internet, entregando documentos a los clientes cuando estos lo solicitan por su nombre, debiendo permanecer continuamente en ejecución, esperando peticiones. Un servidor Web debe implementar el protocolo HTTP (Hypertext Transfer Protocol) para permitir la transferencia de las páginas Web.

Entre los servidores Web más populares encontramos a Internet Information Server (IIS) de Microsoft y Apache, siendo el segundo el más utilizado a nivel mundial desde abril de 1996²⁵.

Apache es un proyecto Open Source, desarrollado por la Apache Software Foundation (<http://www.apache.org/>), que trabaja tanto en entornos Linux/Unix como Windows²⁶. Apache tiene una arquitectura modular, permitiendo agregar o eliminar módulos para obtener un servidor a la medida de las necesidades que se tengan.

²³ GARRETT, J. (2005, Febrero): Ajax: A New Approach to Web Applications. Disponible en: <http://www.adaptivepath.com/>.

²⁴ ZAKAS, N.; MCPEAK, J.; FAWCETT, J. (2006): Profesional Ajax. Anaya Multimedia, pp 23-74.

²⁵ NETCRAFT (2008, Octubre): October 2008 Web Server Survey. Disponible en: <http://news.netcraft.com/>.

²⁶ APACHE (2008): Apache HTTP Server Project. Disponible en: <http://httpd.apache.org/>.

Una vez instalado Apache, para que su computador empiece a funcionar como servidor Web, se debe iniciar el servicio. En Linux se lo puede hacer, según la distribución utilizada, con el comando `service httpd start` (para detenerlo, `service httpd stop`). En Windows, primero será necesario abrir la ventana de *Servicios* desde *Herramientas administrativas* en el *Panel de Control*. El archivo de configuración predeterminado (`/etc/httpd/conf/httpd.conf`, en Linux; `C:\Archivos de programa\Apache Group\Apache2\conf\httpd.conf`, en Windows) instalado en el Servidor Apache HTTP funciona en la mayor parte de los casos, pero de ser necesario puede ser abierto con un editor de texto y configurado manualmente. Si este es el caso, se recomienda primero realizar una revisión de la documentación (puede consultarse ²⁷).

Muchas de las peticiones que recibe un servidor Web requieren el uso de otros recursos; por ejemplo, en el caso de una aplicación Web Mapping es necesario acceder a los datos espaciales. Para que esto sea posible, el servidor Web debe complementarse con otro software llamado Interfaz de Gateway Común (CGI, Common Gateway Interface) o interpretar algún lenguaje de programación de script como PHP, Perl, Java, ASP, etc.

En el primer caso, el servidor Web invoca al programa CGI que aparece en la solicitud del cliente para que la ejecute y le remita los resultados en formato HTML. Los lenguajes de programación que corren del lado del servidor permiten generar páginas HTML de forma dinámica, pudiendo ir embebidos en ellas o utilizarse solos, como código ejecutable (lo mismo que un CGI).

²⁷ KABIR, M. (2003): LA BIBLIA DE SERVIDOR APACHE 2. Madrid, Anaya Multimedia, 862 p.

Los dos servidores antes mencionados pueden operar en cualquiera de estas formas. Una combinación que se ha vuelto muy usual es la de Apache con PHP. En Nexen²⁸ se pueden encontrar estadísticas que demuestran la gran acogida que tiene PHP.

“PHP (acrónimo de Hypertext Preprocessor) es un lenguaje de código abierto interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor”²⁹. PHP es un lenguaje libre, puede trabajar en múltiples plataformas, con gran capacidad de expansión y de conexión a la mayor parte de manejadores de bases datos, soporte para Programación Orientada a Objetos, amplia documentación, y muchas otras características.

La meta de este lenguaje es permitir a los creadores de páginas Web escribir páginas dinámicas, de una manera rápida y sencilla, aunque se puede hacer mucho más con PHP, como por ejemplo utilizarlo para trabajar con servidores de mapas.

2.2.2.3. Servidor de Mapas

Un servidor de mapas es un software con capacidad para interactuar con las fuentes de datos geográficas y presentarlos en Internet con la ayuda de un servidor Web³⁰.

²⁸ SEGUY, D. (2008, Noviembre): PHP statistics for October 2008. Disponible en: <http://www.nexen.net/>.

²⁹ ACHOUR, M. et al. (2008): PHP Manual. Philip Olson (Ed.), PHP Documentation Group.

³⁰ COLL, E. (2005): Introducción a la publicación de cartografía en Internet. Universidad Politécnica de Valencia, pp. 12-14.

En la actualidad existen varias opciones en cuanto a servidores de mapas, algunas de las cuales requieren una considerable inversión de dinero, como en el caso de ArcIMS de ESRI, mientras que otras no, entre las que figuran MapServer y GeoServer, que son de tipo OpenSource. Si se opta por el segundo enfoque, la elección de uno de ellos no es tan simple pues no existe una fórmula bien definida para hacerlo, siendo mejor analizar las características (³¹ y ³²) de cada cual según el tipo de proyecto a desarrollar.

MapServer, desarrollado por la Universidad de Minnesota, es un programa de renderizado de mapas, que trabaja en un entorno web, como un CGI (Common Gateway Interfaz) o como una aplicación autosuficiente a través de una API (Application Programming Interface) accesible desde varios lenguajes de programación como PHP, Perl, etc.³³.

Usualmente, MapServer trabaja detrás de un servidor Web, que como ya se dijo antes, es quién recibe los requerimientos de mapas y los pasa a MapServer para que los cree. MapServer genera la imagen del mapa requerido y la entrega al servidor Web (Apache), el cual la transmite de vuelta al usuario. Esto es lo que se muestra en la figura 2.3.

³¹ MAPSERVER (2008). Welcome to MapServer. Disponible en: <http://mapserver.gis.umn.edu/>.

³² GEOSERVER: GeoServer Features. Disponible en: <http://geoserver.org/>.

³³ KROPLA, B.(2005): Beginning MapServer: Open Source GIS Development. New York, Apress, 418 p.

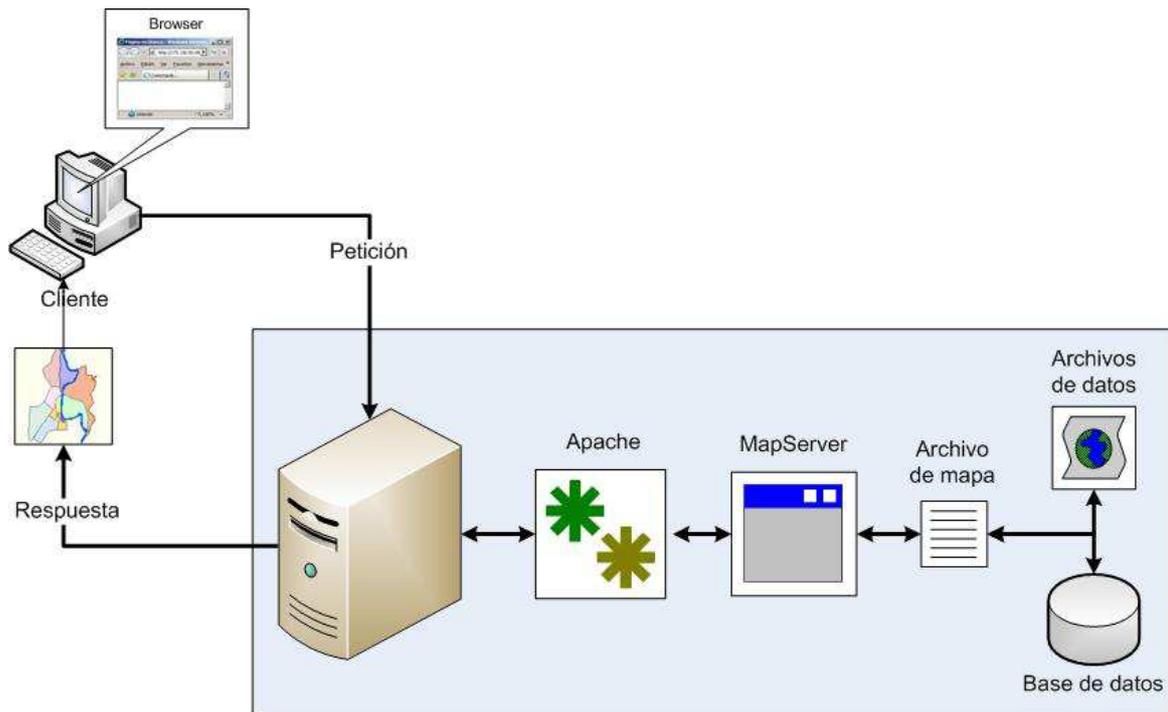


Figura 2.3. Componentes de MapServer.

La principal función de MapServer es leer los datos de varios orígenes y juntar estas capas dentro de un archivo de imagen (GIF, JPGE, PNG).

Este proceso de dibujado (renderizado) ocurre cada vez que se hace un nuevo requerimiento a MapServer, por ejemplo, cuando un usuario amplía o acerca la vista del mapa.

MapServer es como un motor que requiere combustible para funcionar y un sistema de entrega de ese combustible a tal motor. El programa MapServer necesita conocer qué capas del mapa y cómo se dibujarán, y dónde están localizados los orígenes de datos. Los datos son el combustible, y el archivo de mapa (archivo .map) el sistema que lo proporciona. El archivo de mapa es un archivo de texto que contiene las configuraciones necesarias para dibujar e

interactuar con el mapa. Incluye información sobre qué capas serán dibujadas, dónde está el enfoque geográfico del mapa, qué sistema de proyecciones se están usando, el formato de la imagen generada, parámetros para indicar si se incluirán leyendas y escala, etc. Algunos de estos elementos se pueden apreciar en la figura 2.4.

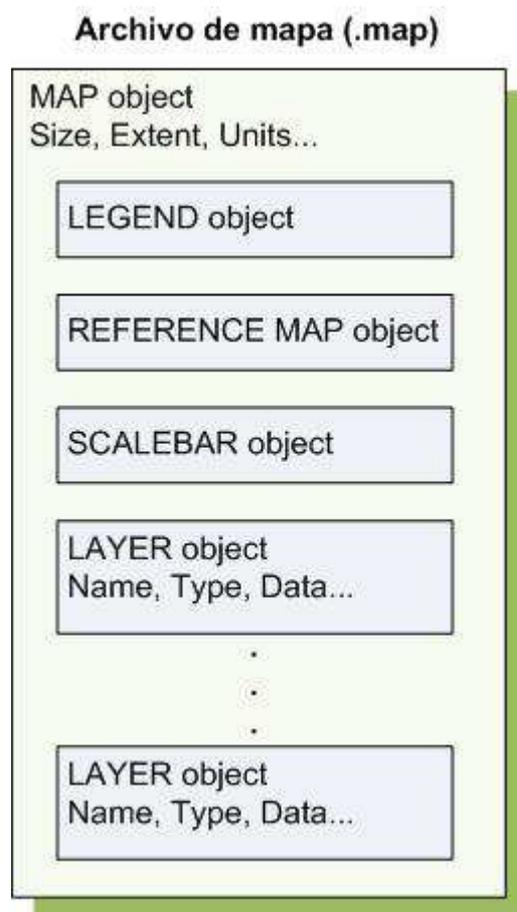


Figura 2.4. Ejemplo de estructura de un archivo de mapa.

También es posible crear aplicaciones con MapServer sin utilizar un archivo de mapa, pero en este caso será necesario utilizar el MapScript. El MapScript es una API que permite crear aplicaciones desde las cuales se invocan las funciones de MapServer. Estas aplicaciones son escritas en un lenguaje de programación,

como PHP. Si este es el caso, el módulo PHP MapScript, desarrollado por DM Solutions Group (<http://www.dmsolutions.ca>), debe ser cargado.

Para una completa referencia sobre MapServer, archivo de mapa, PHP MapScript, y más, se puede consultar la documentación online del proyecto MapServer³⁴.

2.2.2.4. Los datos

Como se indicaba en el apartado anterior, los datos juegan un papel fundamental en una aplicación Web Mapping. Los datos con los que se requiera interactuar corresponden a archivos en algún formato específico o pueden residir en una base de datos, teniendo cuidado que el servidor de mapas seleccionado pueda acceder a ellos.

En el caso de MapServer, puede usar una gran variedad de orígenes de datos para crear los mapas, entre los que se encuentran los archivos shapefile, base de datos como PostGreSQL, entre otros.

2.3. BASES DE DATOS ESPACIALES

En el mundo actual existe una demanda de datos cada vez mayor, llegando las bases de datos a desempeñar un papel crucial en las distintas áreas, incluyendo la geográfica.

³⁴ MAPSERVER: Documentation. Disponible en: <http://mapserver.gis.umn.edu/docs>.

Las bases de datos generalmente contienen datos alfanuméricos, lo que limita sus capacidades ante la necesidad de almacenar datos espaciales. La solución está en mejorar el software existente, o desarrollar uno nuevo, para permitir también la adecuada gestión de estos datos espaciales.

2.3.1. Bases de datos y DBMS

Una base de datos es una colección de datos relacionados, almacenados de forma organizada, con el fin de poder gestionarlos fácilmente.

Las bases de datos no necesariamente deben estar en formato digital, pero gracias a los avances en el campo de la Informática, la mayoría de ellas lo están. Para el efecto, existe un tipo de software llamado Sistema de Gestión de Bases de Datos o DBMS.

Un DBMS (DataBase Management System) es un programa o conjunto de programas que permiten crear y mantener las bases de datos, asegurando su integridad, seguridad y confidencialidad. Sirve de interfaz entre la base de datos y los usuarios que la utilizan. Entre los sistemas de gestión de bases de datos más usados en el medio encontramos Microsoft SQL Server, Oracle, PostgreSQL, MySQL, Microsoft Access.

Se pueden encontrar varias clases de DBMS, siendo el más habitual el relacional (RDBMS). Los RDBMS almacenan y administran los datos en forma de tablas y

para su manipulación se provee de un álgebra relacional y uno o más lenguajes como el SQL (Structured Query Language).

2.3.2. Base de datos y DBMS espaciales

Las bases de datos pueden ser optimizadas para permitir guardar y consultar datos relacionados con objetos en el espacio, incluyendo puntos, líneas y polígonos, dando origen a las bases de datos espaciales.

Una base de datos espacial es un tipo de base de datos, que almacena representaciones de fenómenos del mundo real para ser usados en un SIG³⁵.

Para implementar una base de datos espacial se debe disponer de un Sistema de Gestión de Base de Datos Espacial (remitirse también a ³⁶).

Los DBMS espaciales se caracterizan porque permiten representar tipos de datos espaciales, eficiente acceso a los datos permitiendo aplicar índices a los datos espaciales, incorporan un lenguaje de consulta con operaciones aplicables a los objetos geométricos, entre otras³⁷.

Aprovechando las capacidades de los DBMS para operar en entornos distribuidos, las bases de datos espaciales pueden ser utilizadas como fuentes de

³⁵ BY, R.A. de (2001): Principles of Geographic Information Systems: An Introductory Textbook. International Institute for Aerospace Survey and Earth Sciences (ITC), p. 52.

³⁶ COMPTE, M; STROBL, J.; RESL, R. (2007): Bases de datos espaciales. UNIGIS Professional para América Latina, pp. 111-113.

³⁷ RIGAUX, P.; SCHOLL, M.; VOISARD, A. (2001): Spatial Databases: With Application to GIS, Segunda edición. Morgan Kauffman, pp. 25-26.

datos accesibles desde la Web, como en el caso de las aplicaciones Web Mapping. En este sentido, el DBMS PostgreSQL con su extensión espacial PostGIS, de tipo Open Source, se ha convertido en una herramienta de gran importancia para la elaboración y uso de bases de datos espaciales en trabajos como: sistema móvil de información turística³⁸, SIG para la gestión de suelo³⁹, colaboración en la planeación de expediciones⁴⁰, colaboración en la toma de decisiones⁴¹, búsqueda de rutas⁴².

2.3.3. PostgreSQL y PostGIS

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacional (ORDBMS), desarrollado en la Universidad Berkeley de California. PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo⁴³.

Algunas de las características a destacar de PostgreSQL son:

- Puede correr en casi todos los sistemas operativos (Linux, Windows, etc).
- Soporta todas las características de una base de datos profesional.

³⁸ COMPTE, M.; MOLINA, J.; TURBAU, E.: d-Ruta, un sistema móvil de información turística. En I Jornadas de SIG Libre. Disponible en: <http://www.sigte.udg.es/>.

³⁹ MIRA, J.; NAVARRO, J.; RAMON, A.: SIGUA: SIG libre para la gestión del suelo de la Universidad de Alicante. En I Jornadas de SIG Libre. Disponible en: <http://www.sigte.udg.es/>.

⁴⁰ GORNI, D. et al.: Sistema de Información Geográfica para Web con informaciones espaciales (geográficas) de output y input para Expediciones. En I Jornadas de SIG Libre. Disponible en: <http://www.sigte.udg.es/>.

⁴¹ HALL, G.; ALPERIN, J.; LEON, S. (2007, Mayo): El uso de Internet con software libre y fuentes abiertas espaciales para colaborar en la toma de decisiones. En XI Conferencia Iberoamericana en Sistemas de Información Geográfica. Buenos Aires, Argentina.

⁴² ANDERSON, J.; KUNGYS, V.; STEINBAUER, J.: Ride the City: Disponible en: <http://www.ridethecity.com>.

⁴³ PostgreSQL: About. Disponible en <http://www.postgresql.org>.

- Posee amplia documentación de libre acceso.
- Tiene soporte nativo para lenguajes populares (PHP, C++, Perl, etc.).
- Incluye utilidades para limpieza y respaldo de la de la base de datos y para el análisis de consultas.
- Puede ser extendido por los usuarios de varias formas, añadiendo nuevos tipos de datos, funciones, etc.

Con la finalidad de que la base de datos PostgreSQL soporte objetos geográficos se ha desarrollado el módulo PostGIS, convirtiéndola en una base de datos espacial que se puede utilizar en Sistemas de Información Geográfica.

PostGIS es una extensión espacial para PostgreSQL que permite almacenar entidades geométricas como objetos en tablas relacionales y ofrece multitud de funciones para el mantenimiento y análisis espacial⁴⁴.

PostGIS fue desarrollado por Refrations Research Inc (<http://www.refrations.net>), como un proyecto de investigación de tecnologías de bases de datos espaciales. Con PostGIS podemos usar todos los objetos que aparecen en la especificación OpenGIS como puntos, líneas, polígonos, multilíneas, multipuntos, y colecciones geométricas⁴⁵.

⁴⁴ COMPTE, M; STROBL, J.; RESL, R. (2007): Bases de datos espaciales. UNIGIS Professional para América Latina.

⁴⁵ RAMSEY, P.; MARTIN, M.: Manual PotsGIS. Disponible en <http://postgis.refrations.net>.

3. MATERIALES

3.1. EQUIPOS

Para el desarrollo del presente proyecto, la Unidad de Investigación Informática para el Ordenamiento Territorial y Ambiental (INFOTERRA), de la Universidad Técnica Estatal de Quevedo (UTEQ), ha facilitado los equipos y espacio necesarios, tanto para el levantamiento y preparación de los datos, así como para la creación de la aplicación. Estos equipos son un computador y un GPS, cuyas principales características se mencionan a continuación.

3.1.1. Computador

- Motherboard Intel DG33BU.
- Procesador Intel Core.2 DUO de 2.66 GHz.
- Memoria RAM 2GB.
- Disco duro Samsung de 320 GB.
- Tarjeta de vídeo Sapphire X1550 de 256 MB.
- Monitor LG Flatron L1718S.
- Unidad de DVD-RW LG GSA-H54N.
- Tarjeta de red Ethernet.
- UPS.

3.1.2. GPS

- Magellan 315.
- Precisión aproximada de 10 a 15 metros.
- Puerto (serial) estándar para comunicación con PC.

3.2. SOFTWARE

- Sistema operativo Linux Fedora Core 8.
- Servidor Web Apache 2.2.
- Lenguajes de programación PHP 5.2 y PHP MapScript,
- Lenguaje de creación de páginas Web HTML y lenguaje de scripts JavaScript.
- Servidor de Mapas MapServer 5.0.
- Sistema de gestión de base de datos PostgreSQL Plus 8.3, con extensión espacial PostGIS 1.3 e incorporando funciones de ruteo pgRouting.
- Cliente Web AJAX msCross 1.1.9.
- SIG de escritorio ArcGIS Desktop 9.0.
- Programa para GPS Ozi Explorer 3.95.4m.
- Programa para elaboración de diagramas StarUML 5.0.2.

3.3. DATOS

La Unidad de Investigación Informática para el Ordenamiento Territorial y Ambiental proporcionó parte de los datos necesarios para la elaboración del presente trabajo, los cuales se mencionan a continuación:

- Calles del cantón Quevedo, disponible en archivo de AutoCAD (.dwg).
- Manzanas del cantón Quevedo, como archivo shapefile (.shp).

Las principales características de estos datos son:

- Sistema de coordenadas UTM.
- Datum PSAD 56.
- Escala 1:2000.
- Año de última actualización 2006.

Además, gracias al convenio UTEQ – PAE (Programa de Alimentación Escolar) para la actualización del Archivo Maestro de Instituciones Educativas, período lectivo 2008 – 2009, ha sido posible obtener un listado de las instituciones pertenecientes al cantón Quevedo. Dicho listado corresponde a un archivo de Excel que contiene las siguientes columnas:

- Provincia.
- Cantón.
- Parroquia.

- Nombre.
- Sosténimiento.
- Género.
- Grado máximo (código).
- Grado mínimo (código).
- Tipo de coordenada.
- Grados.
- Minutos.
- Segundos.

4. METODOLOGIA

4.1. DESARROLLO DE SOFTWARE

Cuando se pretende construir un producto o sistema, es importante considerar una serie de pasos predecibles que se deben seguir a fin de obtener un resultado de alta calidad y que en la Ingeniería de Software se conocen como proceso de software, tales como RUP (Rational Unified Process) y MSF (Microsoft Solution Framework).

Aunque RUP es adaptable al contexto de cada organización, es más apropiado para proyectos grandes. Por su parte, MSF proporciona un marco de trabajo flexible y escalable, que también puede ser adaptado a las necesidades de cualquier proyecto, pero sin tener en cuenta su tamaño o complejidad, siendo posible aplicar sus componentes, individual o colectivamente, para mejorar la posibilidad de éxito. MSF está compuesto por seis modelos que son: Arquitectura empresarial, Equipo, Proceso, Gestión de Riesgo, Diseño de Componentes y Aplicación⁴⁶.

Para la creación del Mapa Interactivo de la ciudad de Quevedo se hará uso del Modelo de Proceso del MSF. Este modelo combina los mejores principios de otros modelos como el de cascada y el modelo en espiral; está basado en fases, las mismas que se encuentran manejadas por hitos y de forma iterativa, por lo

⁴⁶ MICROSOFT CORPORATION (2003): Microsoft Solutions Framework version 3.0 Overview. Disponible en <http://www.microsoft.com/msf/>.

que puede ser aplicado al desarrollo y empleo de no sólo de aplicaciones tradicionales sino también para aplicaciones distribuidas Web. Además, se guiará el proceso a través de casos de uso y se documentará con diagramas UML (con extensiones para aplicaciones Web y bases de datos).

4.1.1. Modelo de Proceso

El modelo de proceso (MSF Process Model for Application Development) “es el elemento principal de MSF y representa las prácticas recomendadas que Microsoft ha identificado, utilizado y optimizado a partir de sus propias experiencias en la coordinación de proyectos de implementación de aplicaciones e infraestructuras a gran escala”⁴⁷.

Las fases que comprenden este modelo de proceso son (figura 4.1):

- Visión.
- Planeación.
- Desarrollo.
- Estabilización.
- Utilización.

⁴⁷ MICROSOFT TECHNET (2004): Guía de entrega. Disponible en <http://www.microsoft.com/latam/technet/articulos/wireless/ap03.msp/>.

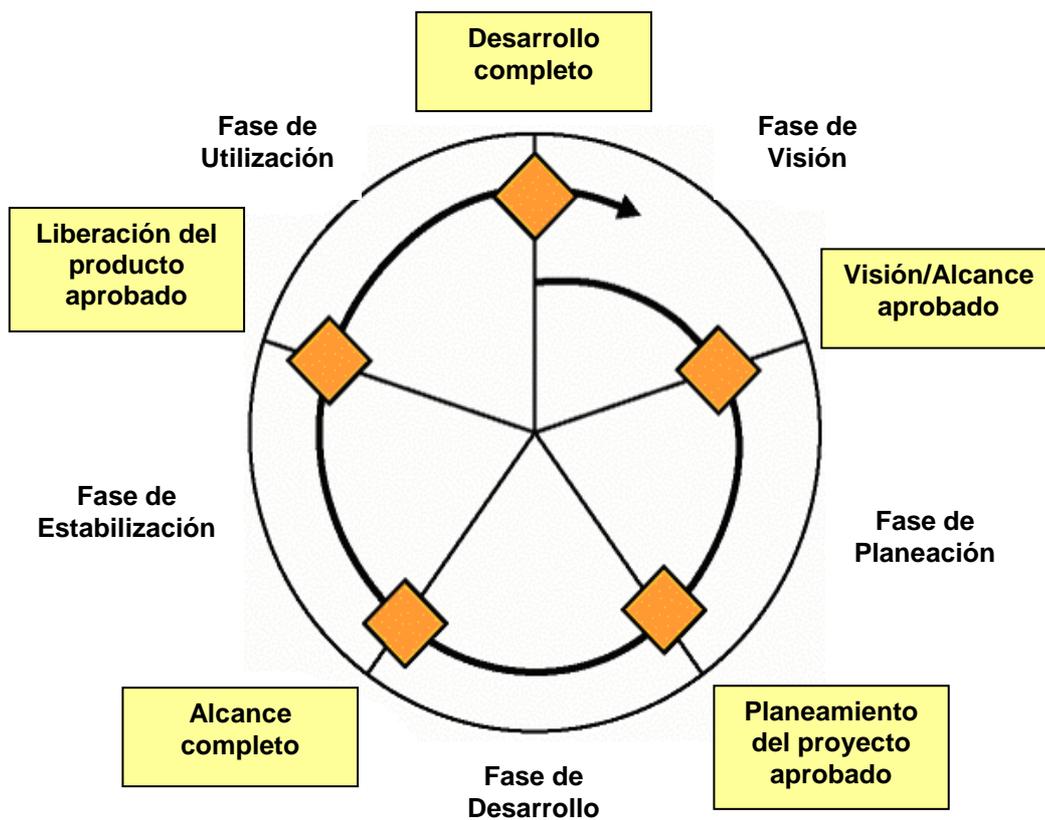


Figura 4.1. Fases e hitos del modelo de proceso MSF⁴⁸.

La **fase de visión** es la fase inicial del proceso MSF y puede entenderse como una descripción amplia de una meta y restricciones de un proyecto. Permite identificar el equipo y qué es lo que dicho equipo debe desarrollar.

En la **fase de planeación** se determina qué es lo que se va a desarrollar y se planea cómo crear la solución. Es decir, se establecen las especificaciones funcionales, se crea un diseño de la solución y se prepara un plan de trabajo. Por lo tanto, esta fase también involucra el análisis de requerimientos.

⁴⁸ MICROSOFT CORPORATION (2002): MSF Process Model v. 3.1. Disponible en <http://www.microsoft.com/msf/>.

Al llegar a la **fase de desarrollo** se procede a crear la solución; es decir, generar todo el código necesario que permite implementar la solución, así como también la respectiva documentación de ese código.

Durante la **fase de estabilización**, se realiza la integración, carga y pruebas necesarias de la solución, a fin de que esté preparada para su liberación.

En la fase final, o sea en la **fase de utilización**, se procede a estabilizar el desarrollo y se obtiene la aprobación final del mismo.

4.1.2. Casos de uso

Con la finalidad de mejorar la comprensión de los requerimientos se elaboran los casos de uso necesarios. “Un caso de uso es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso”⁴⁹.

Se pueden elaborar casos de uso alto nivel o expandidos, permitiendo los segundos alcanzar un conocimiento más profundo de los procesos y de los requerimientos.

Los casos de uso constan de un nombre, una lista de actores, objetivo, resumen, tipo, el orden normal en que se producen los eventos (interacciones actor-sistema) y los posibles cursos alternos.

⁴⁹ LARMAN, Craig (1999): UML y Patrones, Introducción al Análisis y Diseño Orientado a Objetos, Primera Edición. México, Prentice Hall, p. 8.

En el presente proyecto, previo a la elaboración de los casos de uso, y con la finalidad de mejorar la obtención de requerimientos, se realizará una búsqueda y observación de sitios Web similares al que se pretende desarrollar.

4.1.3. Lenguaje de modelado

Como se mencionaba en un párrafo anterior, en la fase de desarrollo se crea y documenta el código necesario. Esta documentación puede elaborarse empleando UML.

El UML o Lenguaje Unificado de Modelado (Unified Modeling Language) “es un lenguaje gráfico usado para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software”⁵⁰.

UML es el resultado de la combinación de tres métodos distintos usados por sus autores: Grady Booch, James Rumbaugh e Ivar Jacobson. UML ha tenido y tiene una gran acogida por parte de los desarrolladores de software, al punto que ha llegado a convertirse en el estándar para el modelado de aplicaciones de software, ya que puede utilizarse incluso para aplicaciones Web y bases de datos.

Esto último ha sido posible gracias a la posibilidad de expandir UML. Es así como, para el modelado de aplicaciones Web, Conallen⁵¹ ha definido una extensión del UML llamada WAE (Web Application Extension for UML), mientras

⁵⁰ BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar (2000): El Lenguaje Unificado de Modelado. España, Addison Wesley, 464 p.

⁵¹ CONALLEN, Jim (2002): Building Web Applications with UML, Second Edition. Addison Wesley, 496 p.

que Naiburg y Maksimchuk⁵² han realizado un trabajo similar para modelar y diseñar bases de datos.

4.2. ADQUISICION DE DATOS

Los datos requeridos para la creación y puesta en marcha del Mapa Interactivo de Quevedo serán integrados en una base de datos tipo PostgreSQL con extensión espacial. Previo a esta integración se procederá a efectuar el levantamiento (sitios de interés), digitalización (red de calles) y preparación (instituciones educativas) de los mismos.

El levantamiento de los puntos necesarios se realizará recorriendo la ciudad de Quevedo, haciendo uso de un GPS y recolectando a la vez toda la información alfanumérica posible, y en caso de no lograr obtenerla o de no ser suficiente, se recurrirá posteriormente a otros medios como guías, periódicos, etc. Una vez levantados todos los lugares, estos serán descargados en el computador utilizando el software respectivo (Ozi Explorer) y exportados a un archivo shapefile al que se le incorporará la información no geográfica obtenida.

La creación de la red de calles de la ciudad se llevará a cabo en el SIG de escritorio (ArcGIS), programa en el cual se efectuará una verificación previa de dicha red.

⁵² NAIBURG, E.; MAKSIMCHUK, R. (2001): UML for Database Design, First Edition. Addison Wesley, 320 p.

La capa correspondiente a los recorridos de buses urbanos se elaborará después de la red de calles, teniendo como base los recorridos descritos en los permisos de operación que otorga el Consejo Provincial de Tránsito y Transporte Terrestre de Los Ríos, y que serán constatados efectuando cada uno de tales recorridos.

Considerando que se dispone de información de las instituciones educativas, estas no serán levantadas pero sí se requerirá su preparación para integrarlos al conjunto de sitios de interés. Para el efecto, se efectuarán las operaciones necesarias, entre las cuales se incluyen la conversión de coordenadas geográficas a UTM y transformación a shapefile.

Teniendo en cuenta también, que la red de calles de la ciudad de Quevedo será integrada en una base de datos tipo PostgreSQL (con extensión espacial), se emplearán las funciones proporcionadas por pgRouting para la búsqueda de rutas. pgRouting es un proyecto de tipo OpenSource, mantenido por PostLBS, y que provee funcionalidades de ruteo para PostgreSQL/PostGIS. Sin embargo, es necesario determinar cuál de los algoritmos (Dijkstra y A*) implementados por pgRouting es el más adecuado para la red vial de Quevedo, tarea que se realizará haciendo uso de las funciones para planes de ejecución de consultas que incluye PostgreSQL.

A fin de mejorar el rendimiento de la base de datos se crearán los índices adicionales que sean necesarios.

5. IMPLEMENTACION

5.1. DOCUMENTO DE REQUERIMIENTOS

Req. ID	Descripción	Caso de Uso
RQ01	Visualizar un mapa con información del cantón Quevedo, que comprende: <ul style="list-style-type: none">a. Parroquias.b. Calles.c. Sitios de interés (hoteles, bancos, instituciones educativas, parques, etc.).	CU01
RQ02	Permitir navegar en el mapa, siendo posible: <ul style="list-style-type: none">a. Acercar.b. Alejar.c. Mover.d. Mostrar información de algún punto especificado.	CU02
RQ03	Permitir mostrar u ocultar las diferentes capas de información.	CU03
RQ04	Encontrar y mostrar la ruta óptima que se debe seguir para desplazarse de un lugar de la ciudad a otro.	CU04
RQ05	Ubicar o buscar en el mapa cierto sitio de interés según su categoría.	CU05

RQ06	Buscar intersecciones de calles.	CU06
RQ07	Visualizar los recorridos de los buses de transporte urbano de Quevedo.	CU07
RQ08	Desplegar toda la información disponible de un determinado sitio.	CU08
RQ09	Imprimir el mapa del cantón con todas las capas que estén visibles en ese momento.	CU09
RQ10	Permitir visualizar el mapa en su estado original en el instante deseado.	CU10

Tabla 5.1. Documento de Requerimientos para el Mapa Interactivo de la ciudad de Quevedo.

5.2. DIAGRAMA DE CASOS DE USO

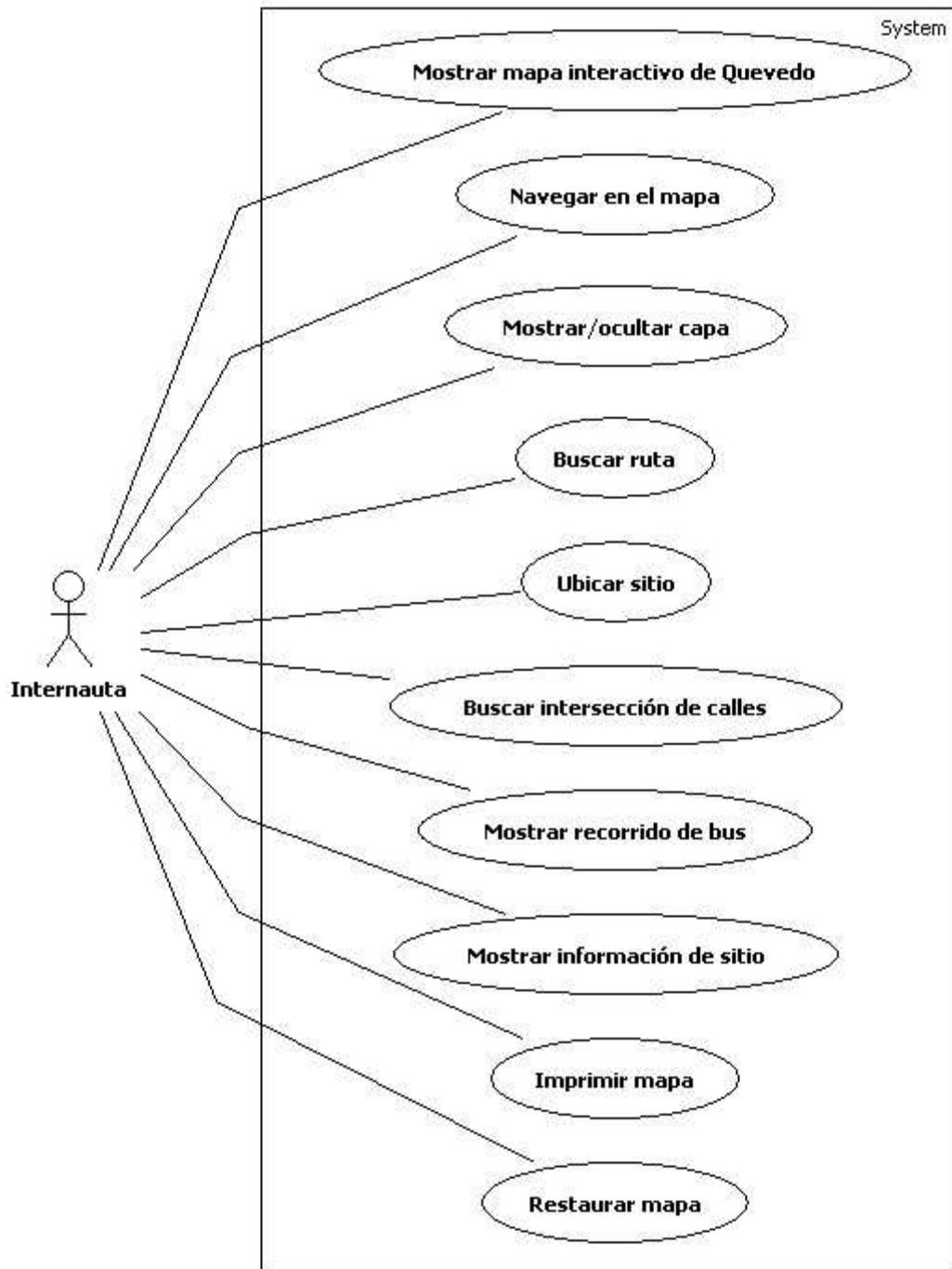


Figura 5.1. Diagrama de casos de uso para el Mapa Interactivo de la ciudad de Quevedo.

5.3. CASO DE USO MOSTRAR MAPA INTERACTIVO DE QUEVEDO

Caso de uso:	Mostrar mapa interactivo de Quevedo.
Caso de uso ID:	CU01
Actores:	Internauta (iniciador).
Propósito:	Generar y mostrar el mapa interactivo de Quevedo con las herramientas para la navegación y búsquedas.
Resumen:	El internauta ingresa al sitio Web y se muestra el mapa de Quevedo con su información básica, así como también su mapa de referencia, leyenda y las herramientas u opciones necesarias para que el usuario pueda navegar y realizar búsquedas.
Tipo:	Primario y real.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el internauta escribe la dirección URL para acceder al servidor donde se encuentra la aplicación.	2. Muestra la lista de capas disponibles junto con su simbología.
	3. Presenta las opciones (cargando sus valores iniciales necesarios) que permitirán realizar búsquedas en el

mapa:

- Ruta entre puntos.
- Sitios.
- Intersección de calles.
- Recorrido de buses urbanos.

4. Genera el visor de mapa con su barra de herramientas y añade también un mapa de referencia.

5. Muestra el mapa de Quevedo con las capas que serán visibles inicialmente:

- Calles.
- Parroquias.
- Río.

6. El internauta empieza a utilizar la aplicación.

Cursos alternos

- Línea 2: El servidor no está disponible y no puede hacer uso de la aplicación.

Diagrama de secuencia

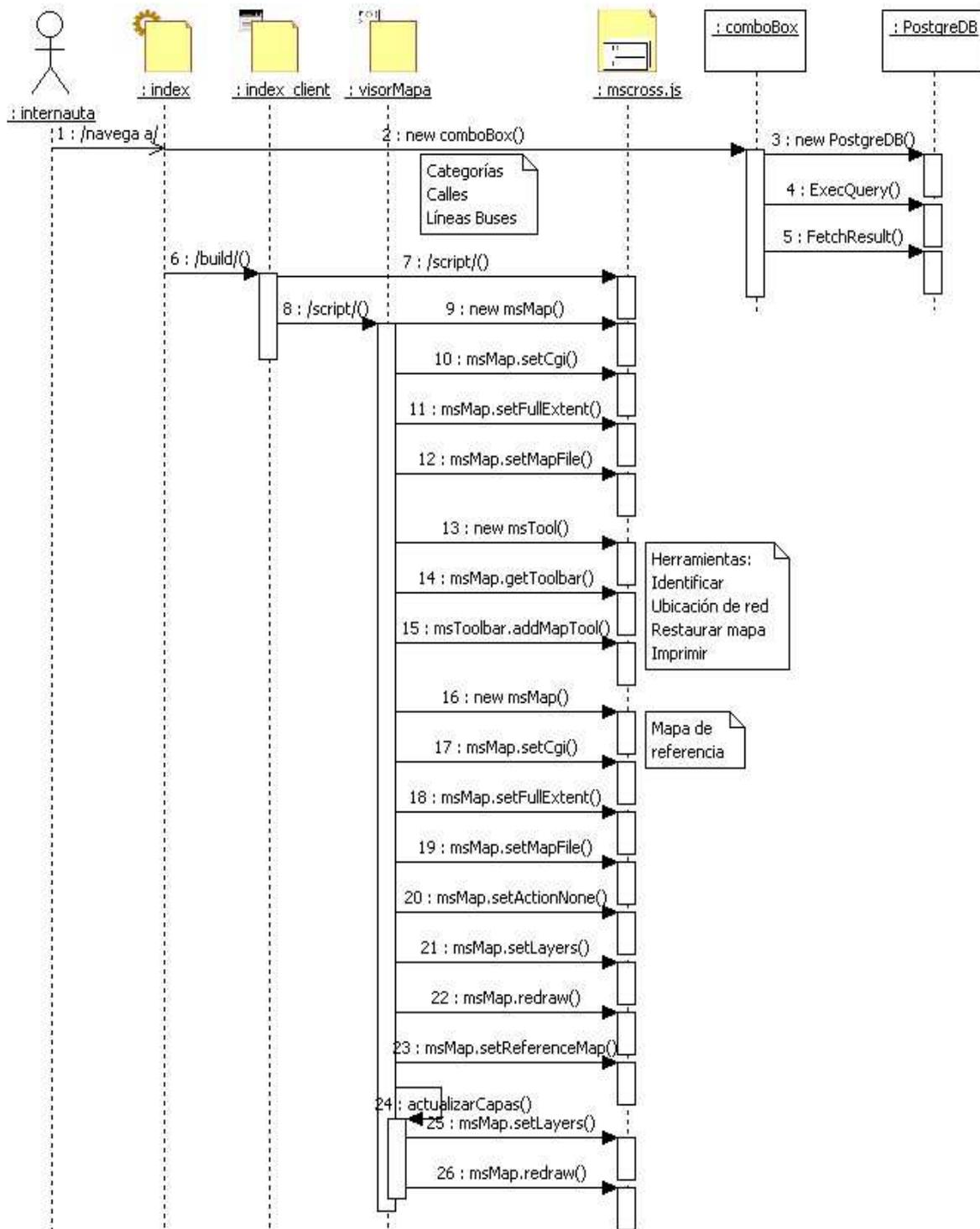


Figura 5.2. Diagrama de secuencia mostrar mapa interactivo de Quevedo.

5.4. CASO DE USO NAVEGAR EN EL MAPA

Caso de uso:	Navegar en el mapa.
Caso de uso ID:	CU02
Actores:	Internauta (iniciador).
Propósito:	Visualizar el mapa del cantón Quevedo y permitir la navegación en él.
Resumen:	El internauta selecciona una de las herramientas disponibles y se desplaza en el mapa haciendo clic con el mouse según lo necesite. Se genera la nueva vista del mapa.
Tipo:	Primario y real.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso comienza cuando el internauta selecciona una herramienta para navegar en el mapa:	
<ul style="list-style-type: none"> • Vista completa. • Mover. • Acercar con rectángulo. • Acercar. 	

- Alejar.
2. El internauta utiliza el mouse para desplazarse en el mapa según su necesidad.
 3. Actualiza el mapa mostrando la imagen correspondiente a su nuevo estado.
 4. El internauta observa el mapa y continúa usando la aplicación.

Cursos alternos

- Línea 2: La herramienta seleccionada no requiere acciones adicionales del internauta. Continúa en la línea 3.

Interfaz de usuario

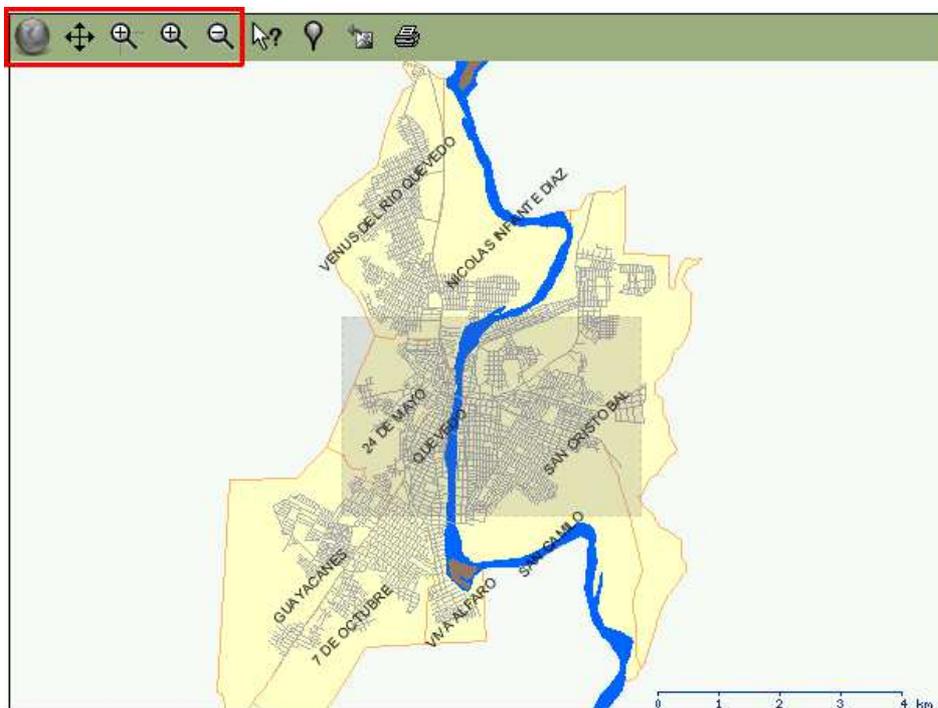


Figura 5.3. Herramientas para navegar en el mapa.

Diagrama de secuencia



Figura 5.4. Diagrama de secuencia navegar en el mapa.

5.5. CASO DE USO MOSTRAR/OCULTAR CAPA

Caso de uso:	Mostrar/ocultar capa.
Caso de uso ID:	CU03
Actores:	Internauta (iniciador).
Propósito:	Actualizar el mapa mostrando u ocultando las capas según corresponda.
Resumen:	El internauta hace clic en la casilla que corresponde a la capa deseada. Se actualiza la lista de capas visibles y se muestra la nueva imagen del mapa.
Tipo:	Primario y real.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el internauta hace clic en la casilla que pertenece a alguna capa.	2. Revisa el estado de las capas - teniendo en cuenta que serán visibles aquellas cuya casilla esté seleccionada - y genera una lista de capas visibles.
	3. Actualiza el mapa mostrando las capas cuyo estado es encendido (visible).
4. El internauta observa el mapa y continúa usando la aplicación.	

Interfaz de usuario



Figura 5.5. Lista de capas disponibles.

Diagrama de secuencia

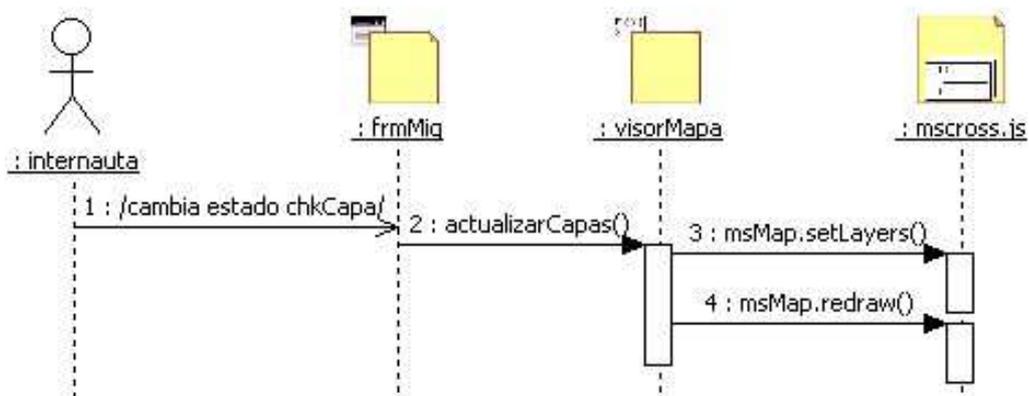


Figura 5.6. Diagrama de secuencia mostrar/ocultar capa.

5.6. CASO DE USO BUSCAR RUTA

Caso de uso:	Buscar ruta.
Caso de uso ID:	CU04
Actores:	Internauta (iniciador).
Propósito:	Encontrar y mostrar la ruta óptima que se debe seguir para desplazarse de un lugar de la ciudad a otro.
Resumen:	Utilizando la herramienta para añadir localizaciones de red, el internauta coloca dos puntos en el mapa. Selecciona la opción de búsqueda de ruta, especifica los parámetros necesarios y la ejecuta. El mapa se actualiza mostrando además la ruta encontrada.
Tipo:	Primario y real.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el internauta selecciona la herramienta para añadir localizaciones de red.	
2. Haciendo clic en el mapa el internauta coloca los puntos (dos: origen y destino) en los lugares deseados.	3. Agrega los puntos al mapa según corresponda.

4. Selecciona la opción *Ruta entre puntos* para configurar y ejecutar la búsqueda.

6. Especifica el tipo de ruta que desea obtener; es decir, la forma de realizar el recorrido (a pie o en carro).

7. Hace clic en el botón *Dibujar*.

9. El internauta observa los resultados y continúa utilizando la aplicación.

5. Muestra los parámetros que deben configurarse, con los valores por defecto que correspondan.

8. Ejecuta la búsqueda con los puntos y parámetros indicados y actualiza el mapa, mostrando una nueva imagen que incluye la ruta encontrada.

Cursos alternos

- Línea 4: La opción para buscar la ruta ya está visible. Continúa en la línea 6.
- Línea 7: No se han especificado los dos puntos. Se informa lo sucedido y se cancela la operación
- Línea 8: No se encontró una solución por lo que no se muestra ninguna ruta.

Interfaz de usuario



Figura 5.7. Agregar localización de red y parámetros para buscar ruta.

Diagrama de secuencia

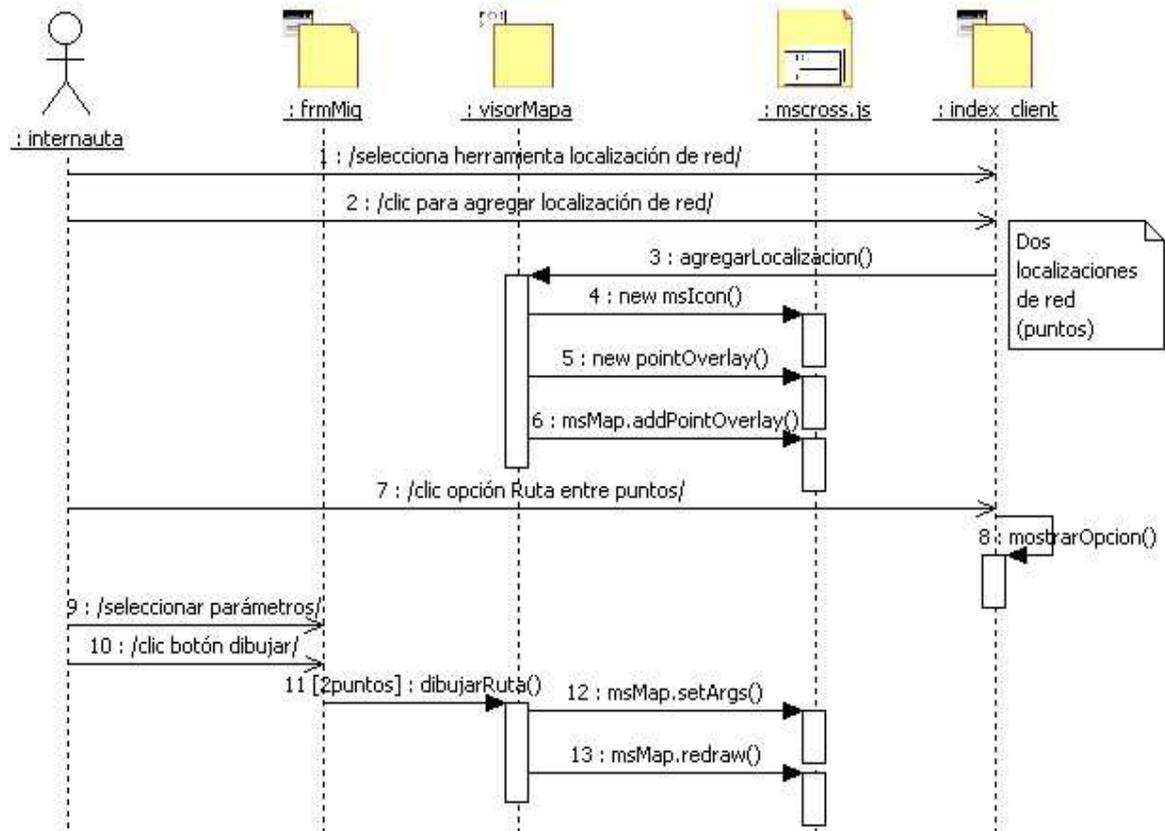


Figura 5.8. Diagrama de secuencia buscar ruta.

5.7. CASO DE USO UBICAR SITIO

Caso de uso:	Ubicar sitio.
Caso de uso ID:	CU05
Actores:	Internauta (iniciador).
Propósito:	Buscar algún sitio de interés y localizarlo en el mapa.
Resumen:	El internauta selecciona la opción para buscar sitios. Escoge una categoría, una subcategoría y el sitio que deseado; escoge la opción para ubicarlo y se muestra en el mapa.
Tipo:	Primario y real.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el internauta selecciona la opción de buscar <i>Sitios</i> .	2. Muestra los parámetros que deben ser especificados.
3. Selecciona una categoría.	4. Carga la lista de subcategorías que pertenecen a la categoría seleccionada.
5. Selecciona una subcategoría.	6. Carga la lista de sitios que pertenecen a esa subcategoría.
7. Selecciona el sitio deseado.	
8. Presiona el botón <i>Ubicar</i> .	9. Hace zoom (acercar) al sitio indicado

y muestra el resultado.

10. El internauta observa el mapa y continúa haciendo uso de la aplicación.

Cursos alternos

- Línea 1: La opción para buscar sitios ya está visible. Continúa en la línea 3.
- Línea 3: Ya tiene seleccionada una categoría. Continúa en la línea 5.
- Línea 5: Ya tiene seleccionada una subcategoría. Continúa en la línea 7.
- Línea 8: No ha seleccionado ningún sitio. Se informa lo ocurrido y cancela la operación.

Interfaz de usuario



The image shows a web form titled "Sitios" with a light green header. Below the header, there are three dropdown menus. The first is labeled "Categoría" and has "-Seleccione-" selected. The second is labeled "Subcategoría" and also has "-Seleccione-" selected. The third is labeled "Sitio" and has "-Seleccione-" selected. Below these dropdowns is a button labeled "Ubicar". The entire form is enclosed in a red border.

Figura 5.9. Parámetros para ubicar sitio.

Diagrama de secuencia

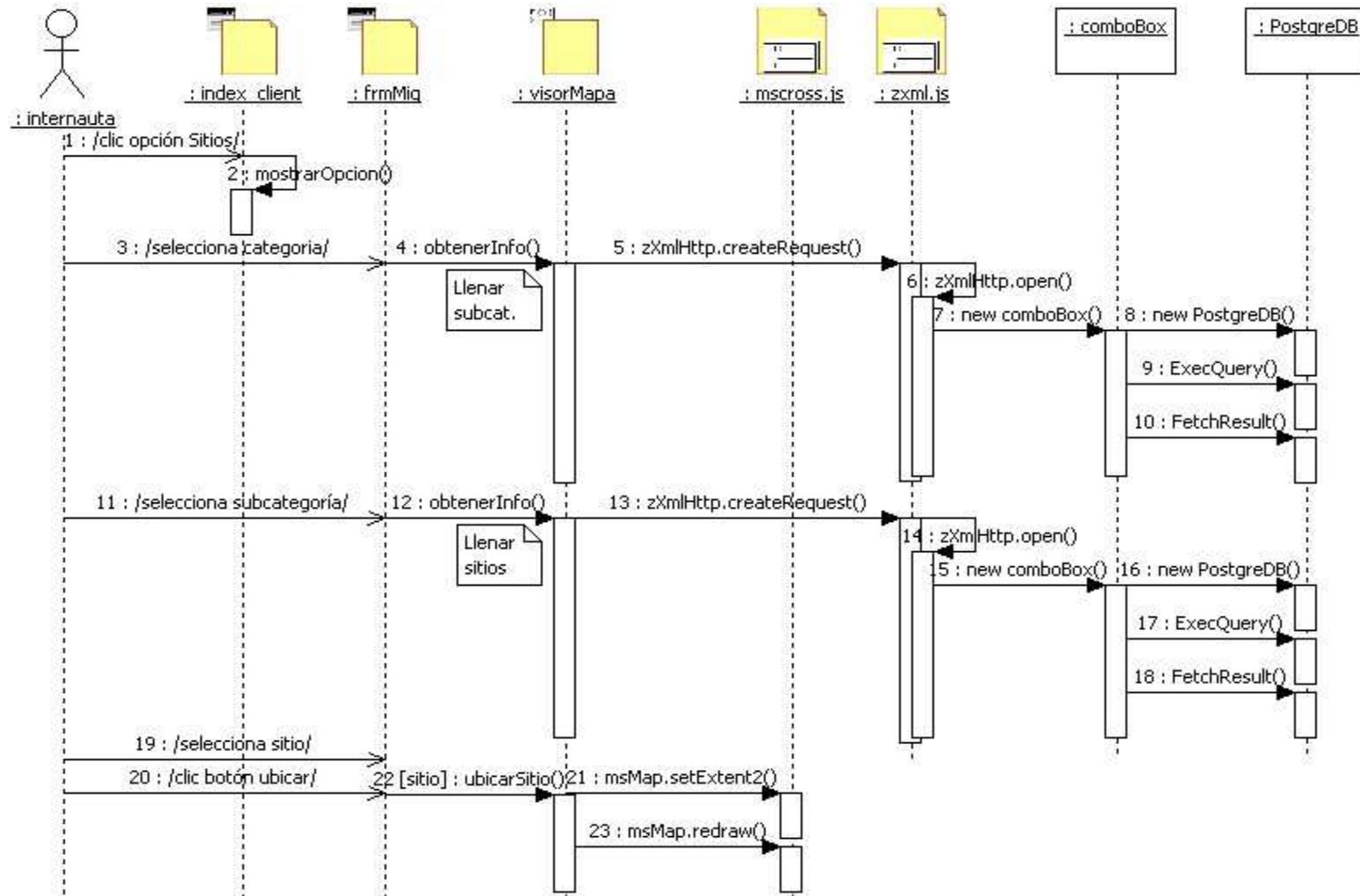


Figura 5.10. Diagrama de secuencia ubicar sitio.

5.8. CASO DE USO BUSCAR INTERSECCIÓN DE CALLES

Caso de uso:	Buscar intersección de calles.
Caso de uso ID:	CU06
Actores:	Internauta (iniciador).
Propósito:	Visualizar el lugar donde se intersecan dos calles.
Resumen:	El internauta selecciona la opción para buscar una intersección de calles, especifica las calles deseadas y ejecuta la búsqueda. Se actualiza el mapa mostrando la intersección de dichas calles.
Tipo:	Primario y real.

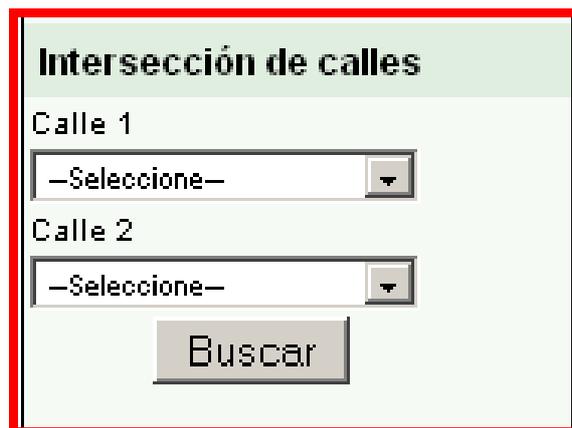
Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso comienza cuando el internauta selecciona la opción <i>Intersección de calles</i> .	2. Muestra los parámetros que deben ser seleccionados.
3. Determina las calles (dos) que forman la intersección.	
4. Hace clic en el botón <i>Buscar</i> .	5. Busca la intersección y muestra el resultado.
6. El internauta observa el mapa y continúa utilizando la aplicación.	

Cursos alternos

- Línea 1: La opción para buscar intersección de calles ya está visible. Continúa en la línea 3.
- Línea 4: Las calles seleccionadas son iguales. Se informa lo ocurrido y cancela la operación.
- Línea 5: La intersección no pudo ser localizada por lo que no se muestra.

Interfaz de usuario



The image shows a user interface window titled "Intersección de calles". It contains two dropdown menus labeled "Calle 1" and "Calle 2", both showing the text "-Seleccione-". Below the dropdowns is a button labeled "Buscar". The entire window is enclosed in a red border.

Figura 5.11. Parámetros para buscar intersección de calles.

Diagrama de secuencia

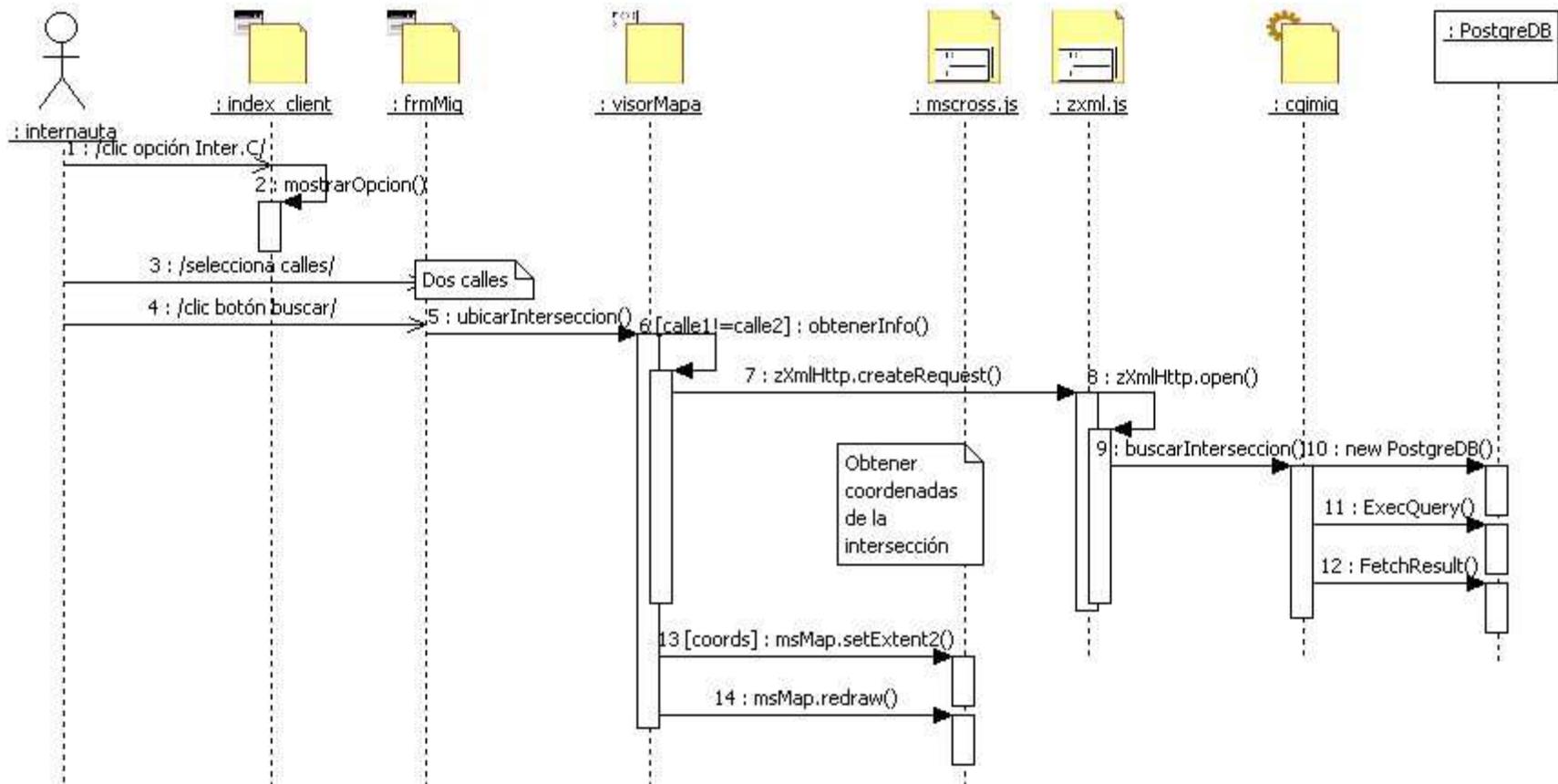


Figura 5.12. Diagrama de secuencia buscar intersección de calles.

5.9. CASO DE USO MOSTRAR RECORRIDO DE BUS

Caso de uso:	Mostrar recorrido de bus.
Caso de uso ID:	CU07
Actores:	Internauta (iniciador).
Propósito:	Visualizar el recorrido de las líneas de buses urbanos de Quevedo.
Resumen:	El internauta selecciona la opción para buscar el recorrido de buses, indica una línea de bus y se muestra su recorrido.
Tipo:	Primario y real.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso comienza cuando el internauta escoge la opción del buscador llamada <i>Recorrido de buses urbanos</i> .	2. Muestra los ítems que deben completarse para realizar la búsqueda.
3. Escoge una línea de bus.	
4. Selecciona la opción para mostrar el recorrido.	5. Obtiene el recorrido que corresponde a la línea especificada y muestra la nueva imagen del mapa.
6. El internauta observa el resultado	

obtenido y sigue utilizando la aplicación.

Cursos alternos

- Línea 1: La opción para buscar el recorrido de bus ya está visible. Continúa en la línea 3.
- Línea 4: No ha seleccionado ninguna línea de bus. Se informa y cancela la operación.
- Línea 5: La capa que corresponde a los recorridos de buses no está visible por lo que no podrá observarlos.

Interfaz de usuario



The image shows a window titled "Recorrido buses urbanos". Inside the window, there is a label "Línea" above a dropdown menu. The dropdown menu currently displays the text "-Seleccione-" and has a small downward-pointing arrow on its right side. Below the dropdown menu is a button labeled "Mostrar". The entire window is enclosed in a red rectangular border.

Figura 5.13. Parámetros para mostrar recorrido de bus.

Diagrama de secuencia

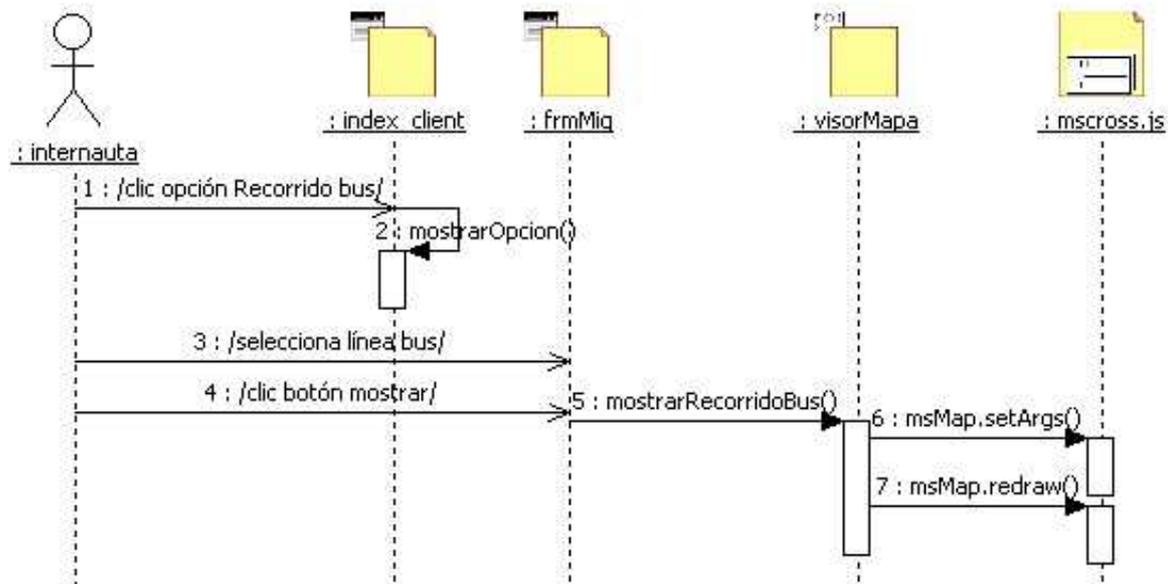


Figura 5.14. Diagrama de secuencia mostrar recorrido de bus.

5.10. CASO DE USO MOSTRAR INFORMACIÓN DE SITIO

Caso de uso:	Mostrar información de sitio.
Caso de uso ID:	CU08
Actores:	Internauta (iniciador).
Propósito:	Identificar los sitios que forman parte del mapa y obtener su información disponible.
Resumen:	El internauta selecciona la herramienta para identificar sitios, señala una capa, hace clic en el mapa y aparece el nombre del sitio. Selecciona la opción para obtener el resto de información la cual es visualizada.
Tipo:	Primario y real.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso empieza cuando el internauta elige la herramienta que permite identificar los sitios (<i>Identificar</i>).	
2. Selecciona la capa cuyos elementos desea identificar.	
3. Hace clic en el lugar deseado dentro del mapa.	4. Consulta el sitio de la capa seleccionada que se encuentra en el

- punto señalado.
5. Muestra el nombre del sitio encontrado.
 6. El internauta hace clic en la opción que permite mostrar la información adicional del sitio.
 7. Presenta el toda la información disponible del sitio.
 8. El internauta revisa los detalles mostrados y sigue dando uso a la aplicación.

Cursos alternos

- Línea 2: La capa que desea consultar ya está seleccionada. Continúa en la línea 3.
- Línea 5: No existe ningún sitio en ese punto por lo que no se muestra nada.

Interfaz de usuario



Figura 5.15. Capa activa, identificar sitio y mostrar su información.

Diagrama de secuencia

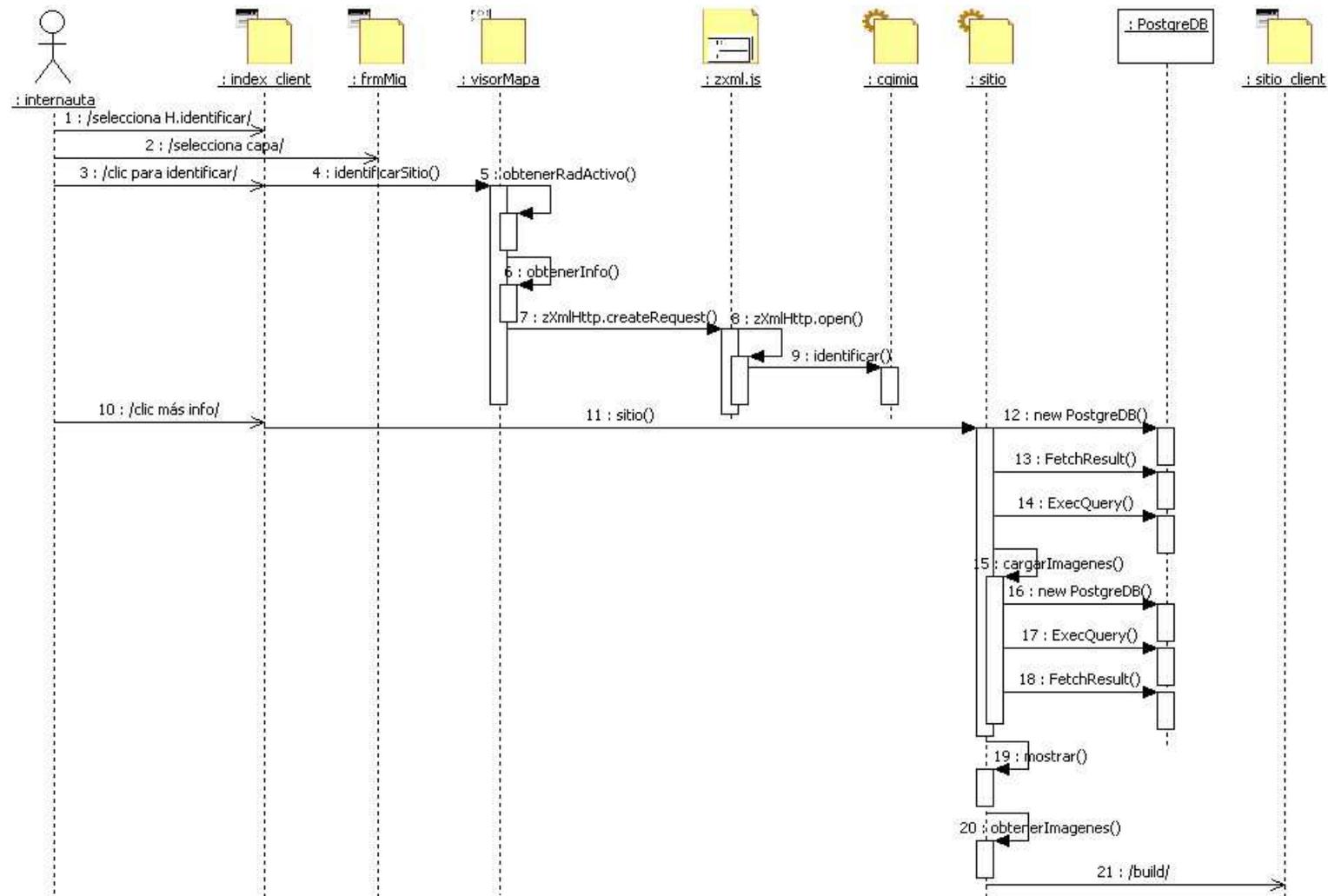


Figura 5.16. Diagrama de secuencia mostrar información de sitio.

5.11. CASO DE USO IMPRIMIR MAPA

Caso de uso:	Imprimir mapa.
Caso de uso ID:	CU09
Actores:	Internauta (iniciador).
Propósito:	Obtener una copia impresa de la vista actual del mapa de Quevedo.
Resumen:	El internauta selecciona la herramienta imprimir. Se despliega una ventana con el mapa preparado y el internauta ejecuta la orden para que inicie la impresión.
Tipo:	Primario y real.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el internauta hace clic en la herramienta para imprimir el mapa.	2. Abre una ventana en la cual aparece el mapa de Quevedo, con los elementos necesarios, listo para ser impreso.
3. Selecciona la opción <i>Imprimir</i> .	4. Envía la orden para imprimir el mapa mostrado.
5. El internauta obtiene su copia impresa del mapa, cierra la ventana y sigue utilizando la aplicación.	

Cursos alternos

- Línea 2: La configuración del navegador usado por el internauta no permite abrir la ventana, por lo que debe corregirla y volver a empezar.

Interfaz de usuario

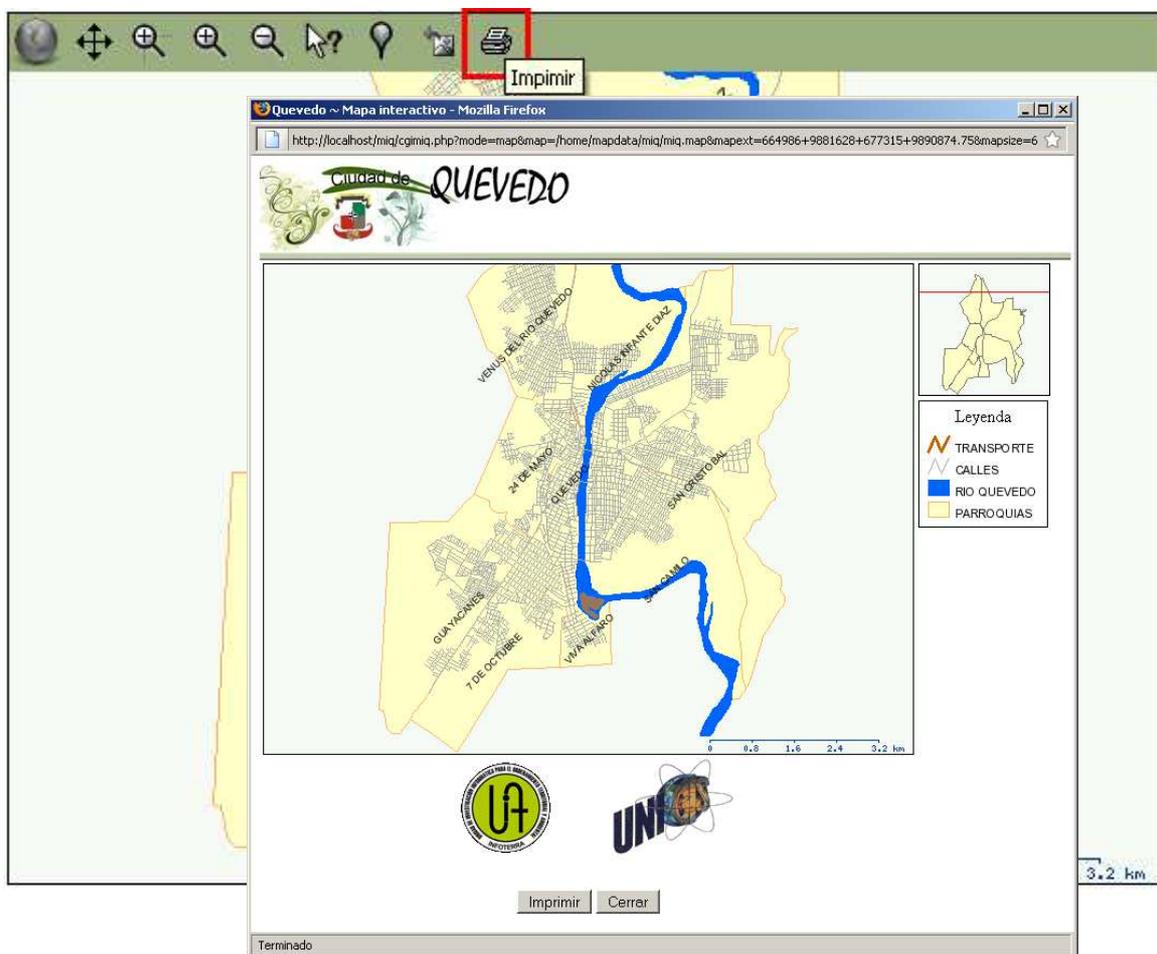


Figura 5.17. Imprimir el mapa.

Diagrama de secuencia

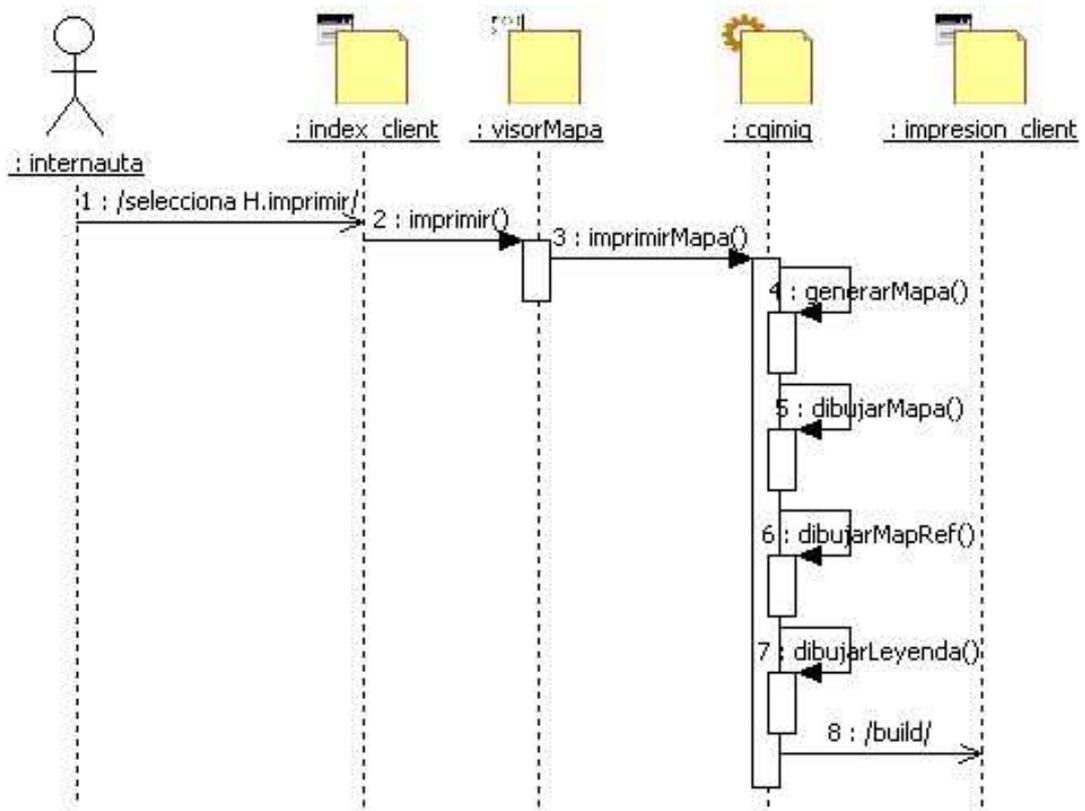


Figura 5.18. Diagrama de secuencia imprimir mapa.

5.12. CASO DE USO RESTAURAR MAPA

Caso de uso:	Restaurar mapa.
Caso de uso ID:	CU10
Actores:	Internauta (iniciador).
Propósito:	Restaurar el mapa a su estado original.
Resumen:	El internauta selecciona la herramienta para restaurar el mapa. Restablece todos los parámetros necesarios y muestra el mapa como la primera vez que se cargó la página.
Tipo:	Primario y real.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el internauta hace clic en la herramienta restaurar mapa.	2. Restablece la extensión (extent) del mapa.
	3. Elimina las localizaciones de red.
	4. Actualiza el mapa mostrando una imagen como la original.
5. El internauta observa el resultado y continúa dando uso a la aplicación.	

Interfaz de usuario

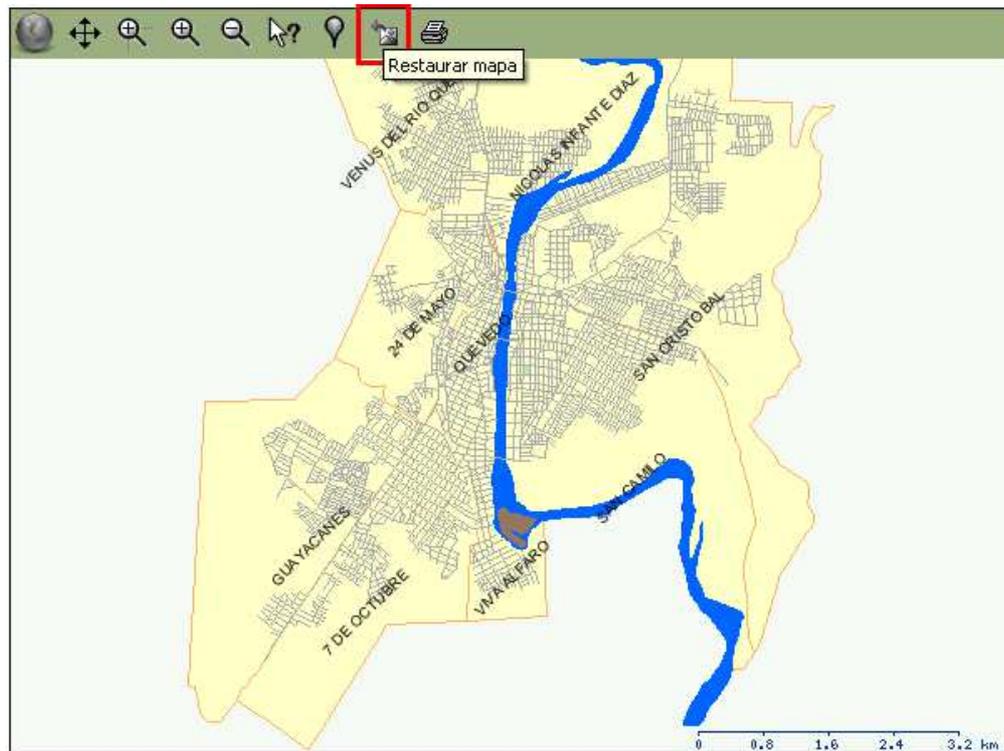


Figura 5.19. Restaurar el mapa.

Diagrama de secuencia

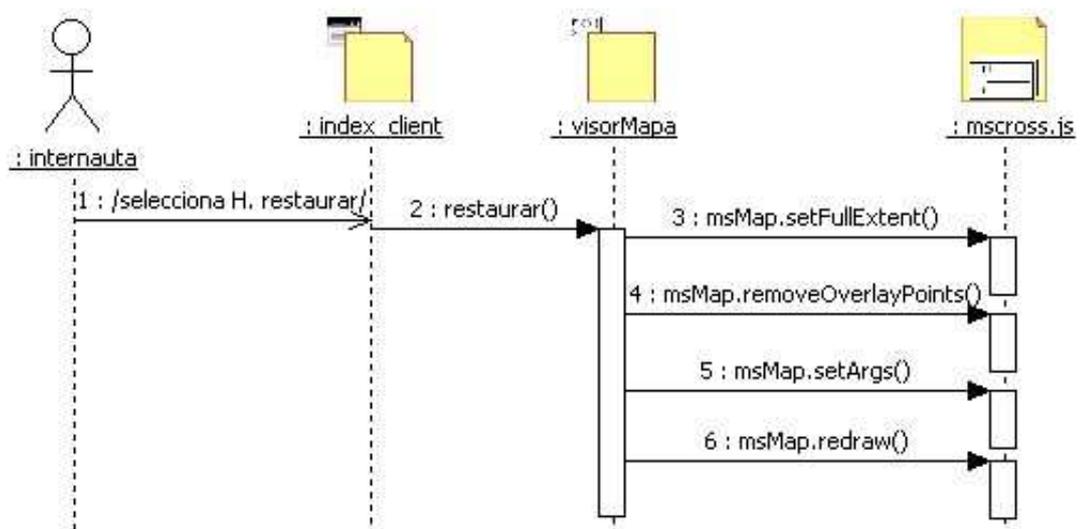


Figura 5.20. Diagrama de secuencia restaurar mapa.

5.13. DIAGRAMA DE CLASES

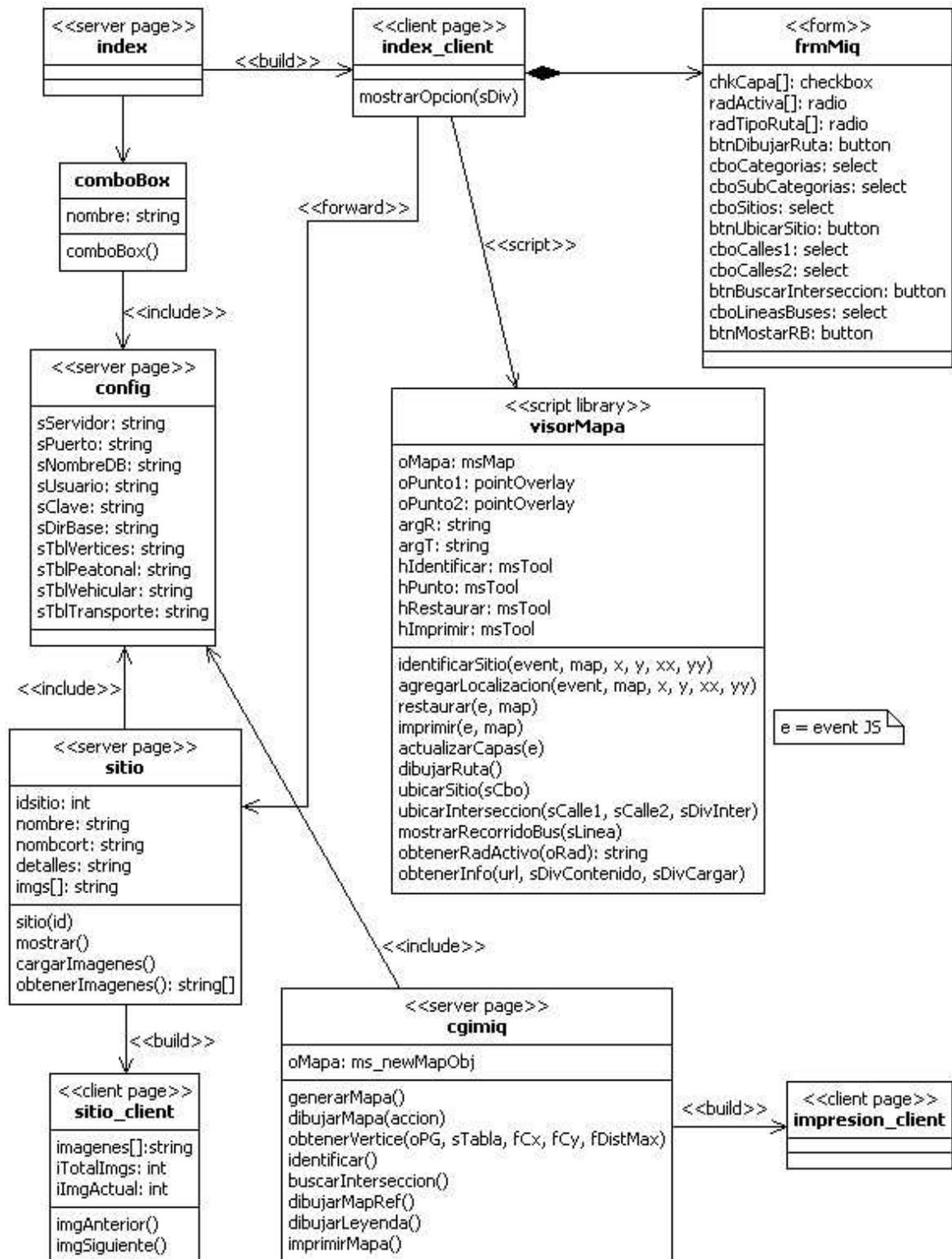


Figura 5.21. Diagrama de clases para el Mapa Interactivo de la ciudad de Quevedo.

5.14. DISEÑO DE LA BASE DE DATOS

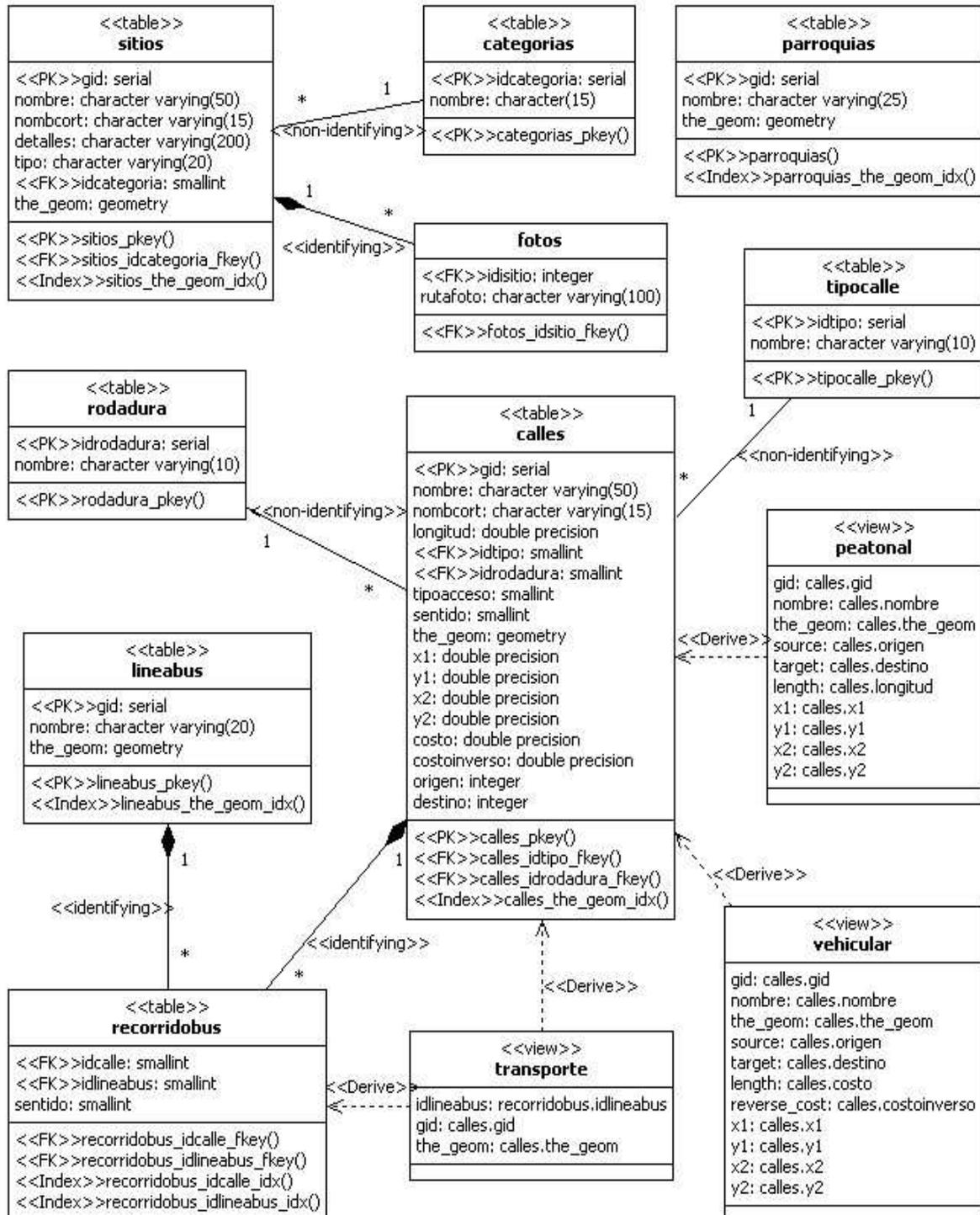


Figura 5.22. Diseño de la base de datos para el Mapa Interactivo de la ciudad de Quevedo.

5.15. DIAGRAMA DE COMPONENTES

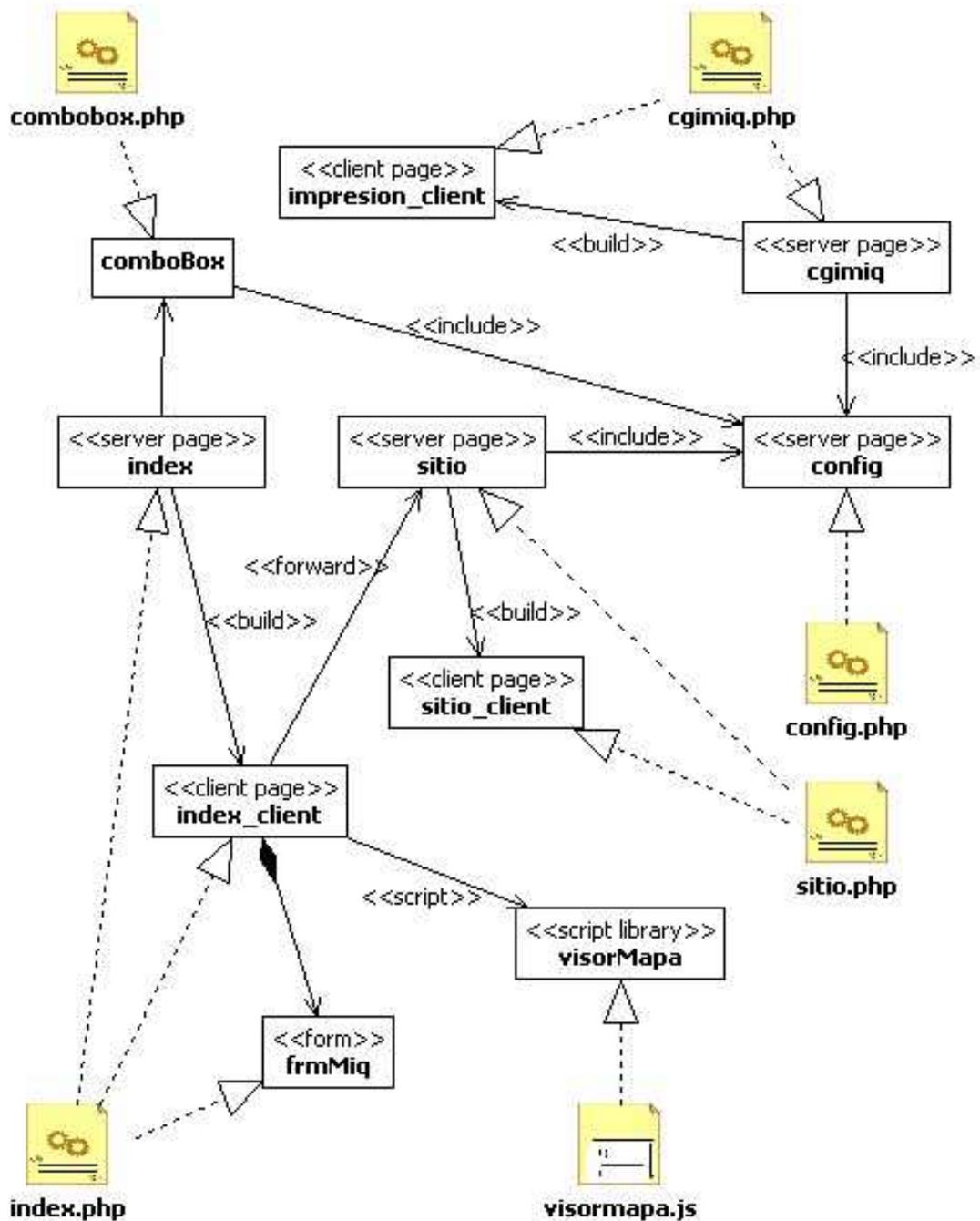


Figura 5.23. Diagrama de componentes para el Mapa Interactivo de la ciudad de Quevedo.

5.16. DIAGRAMA DE DESPLIEGUE

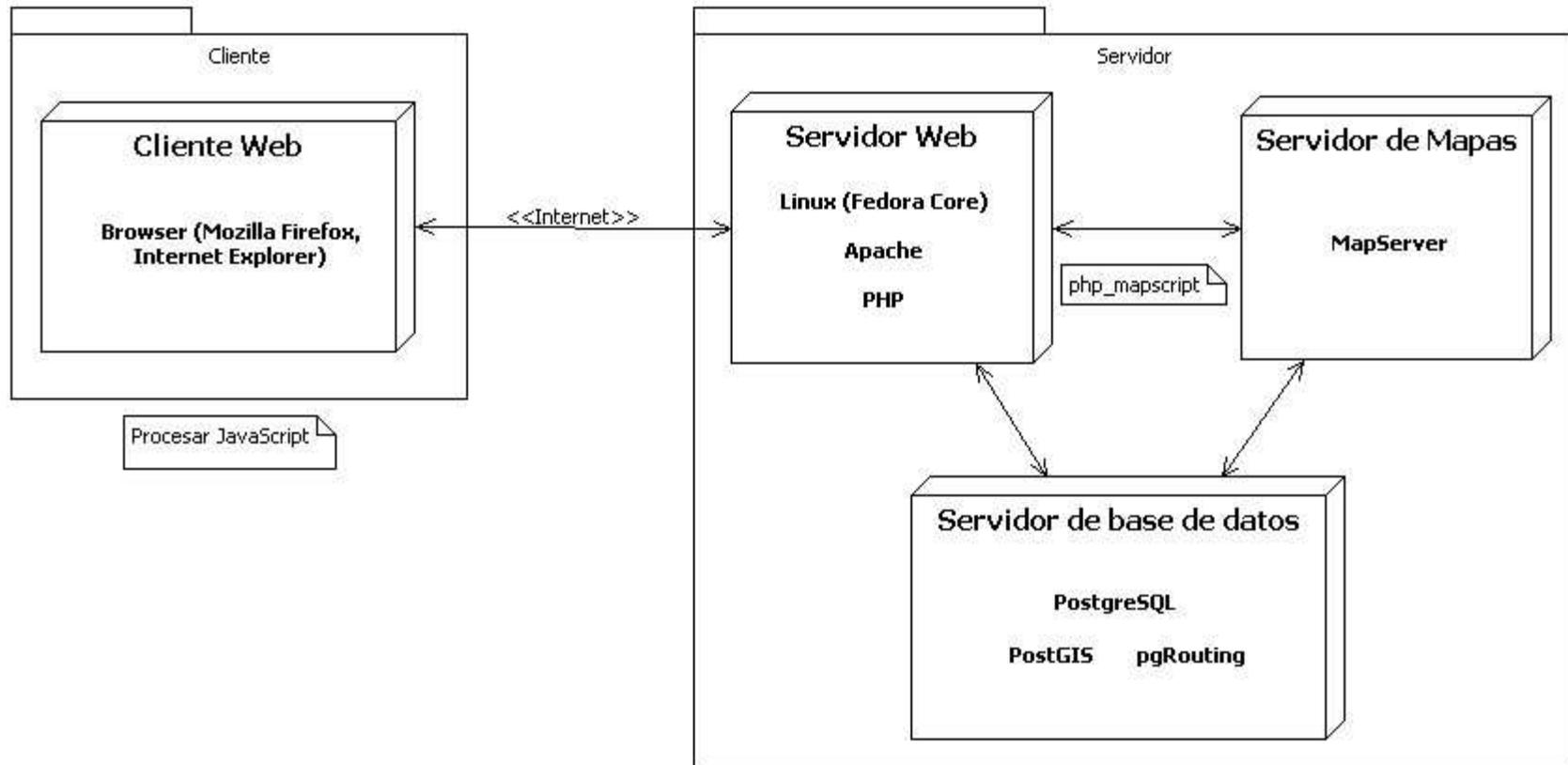


Figura 5.24. Diagrama de despliegue para el Mapa Interactivo de la ciudad de Quevedo.

6. RESULTADOS

Mediante el seguimiento de la metodología descrita anteriormente se ha elaborado un Mapa Interactivo para la ciudad de Quevedo. Este mapa consiste en una aplicación Web Mapping; es decir, un sitio Web que permite la visualización del mapa de Quevedo con su información básica, la cual comprende:

- Parroquias (urbanas).
- Río e islotes.
- Red de calles.
- Sitios de interés.
- Recorridos de buses urbanos.

Estos datos han sido integrados en una base de datos con capacidades de manejo de información espacial.

El mapa incluye una barra de escala y las herramientas necesarias para la navegación en él y para la realización de otras tareas. Las herramientas en mención son:

- Vista completa.
- Mover.
- Acercar (con rectángulo).

- Acercar.
- Alejar.
- Identificar.
- Agregar localización de red
- Restaurar el mapa.
- Imprimir.

El sitio también incluye un mapa de referencia y una sección de búsquedas. Esta sección permite:

- Dibujar la ruta más corta para desplazarse de un lugar de la ciudad a otro, ya sea caminando o en vehículo, considerando para lo segundo los sentidos de las calles.
- Buscar y ubicar algún sitio específico a partir de la categoría y subcategoría requeridas.
- Mostrar (si existe) el lugar en donde se intersecan dos calles.
- Visualizar en forma individual el recorrido que realizan los buses de la transportación urbana.

Además, se pueden mostrar u ocultar las diferentes capas de información del mapa y consultar (identificar) los diferentes objetos en ellas contenidos, con la

posibilidad de desplegar información alfanumérica adicional y fotografías del sitio deseado.

Este sitio Web ha sido construido utilizando la técnica AJAX, lo que evita tener que recargar la página completamente ante alguna acción del usuario, mejorando el rendimiento de la aplicación. Su desarrollo ha sido alcanzado utilizando herramientas de tipo Open Source, las cuales comprenden sistema operativo, servidor Web, lenguajes de programación, servidor de mapas, sistema de gestión de base de datos.

Las siguientes secciones muestran los resultados obtenidos atendiendo a los casos de uso de la aplicación.

6.1. MOSTRAR MAPA INTERACTIVO DE QUEVEDO

La figura 6.1 corresponde a una imagen de lo que los usuarios observarán al ingresar al sitio Web del mapa interactivo de Quevedo; es decir la página principal, que contiene el mapa junto con las herramientas y elementos necesarios.

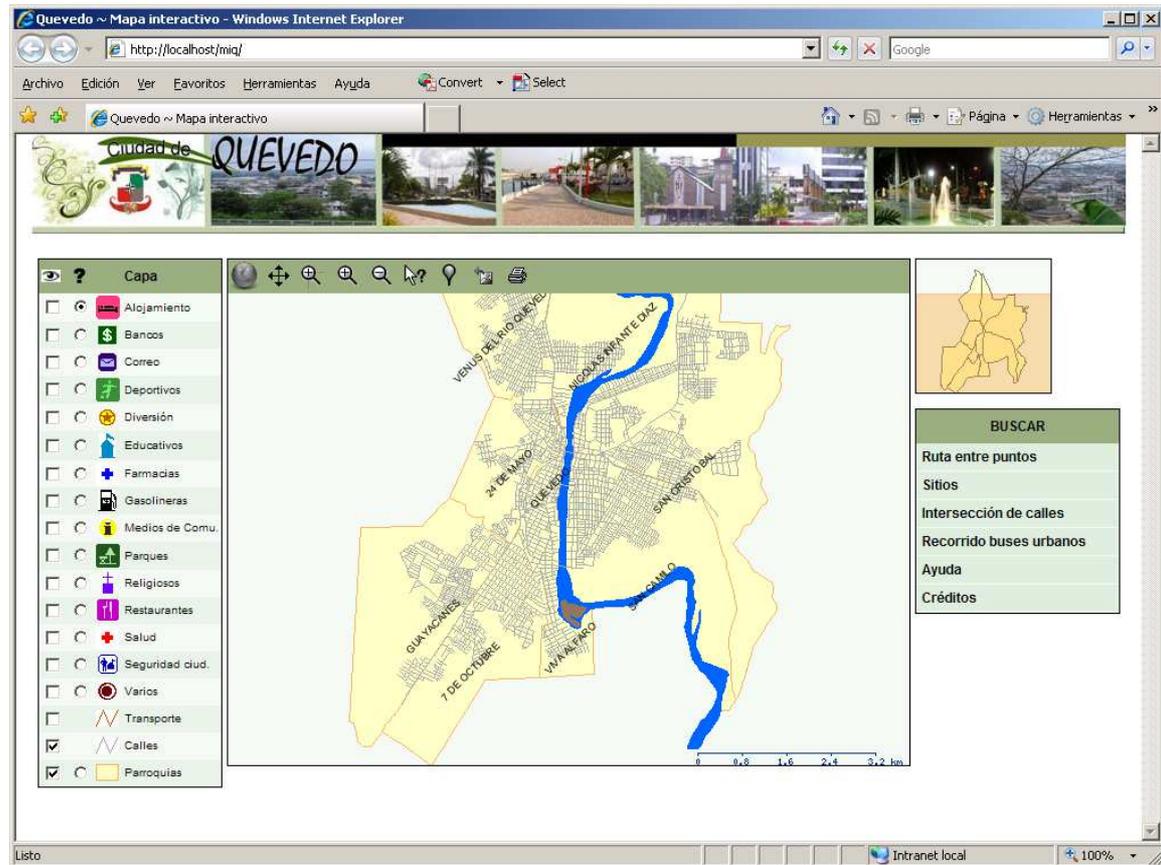


Figura 6.1. Página principal del Mapa Interactivo de la ciudad de Quevedo.

6.2. NAVEGAR EN EL MAPA

Haciendo uso de las herramientas de navegación (vista completa, mover, acercar, alejar), el usuario puede visualizar los diferentes lugares del cantón Quevedo. Por ejemplo, la figura 6.2 muestra el resultado de utilizar la herramienta acercar con rectángulo para visualizar una parte del centro de la ciudad.

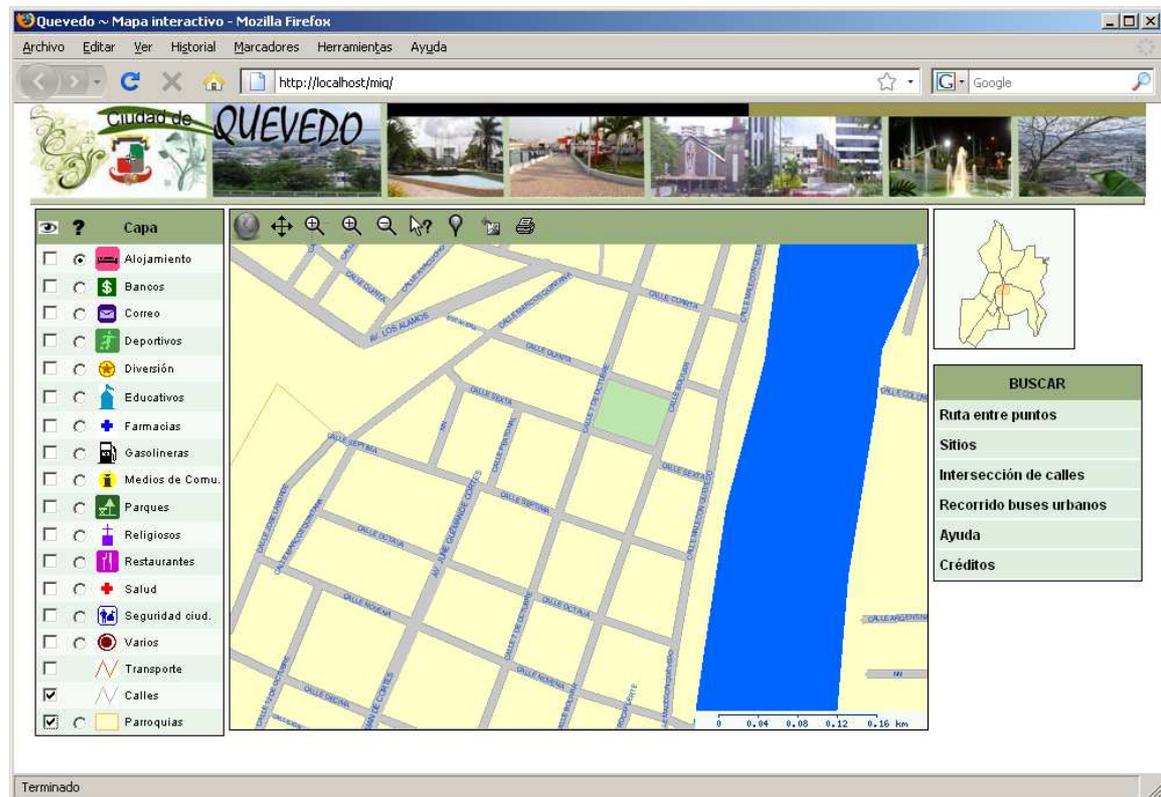


Figura 6.2. Centro de la ciudad de Quevedo visualizado usando la herramienta acercar con rectángulo.

6.3. MOSTRAR/OCULTAR CAPA

Junto a la sección donde se visualiza el mapa se encuentra la lista de capas con su simbología (leyenda), con la posibilidad de mostrarlas u ocultarlas, ya sea activando o desactivando su casilla, respectivamente. Esto es lo que muestra la figura 6.3 que es similar a la figura 6.2 pero con las capas alojamiento y bancos visibles y la capa de parroquias oculta.

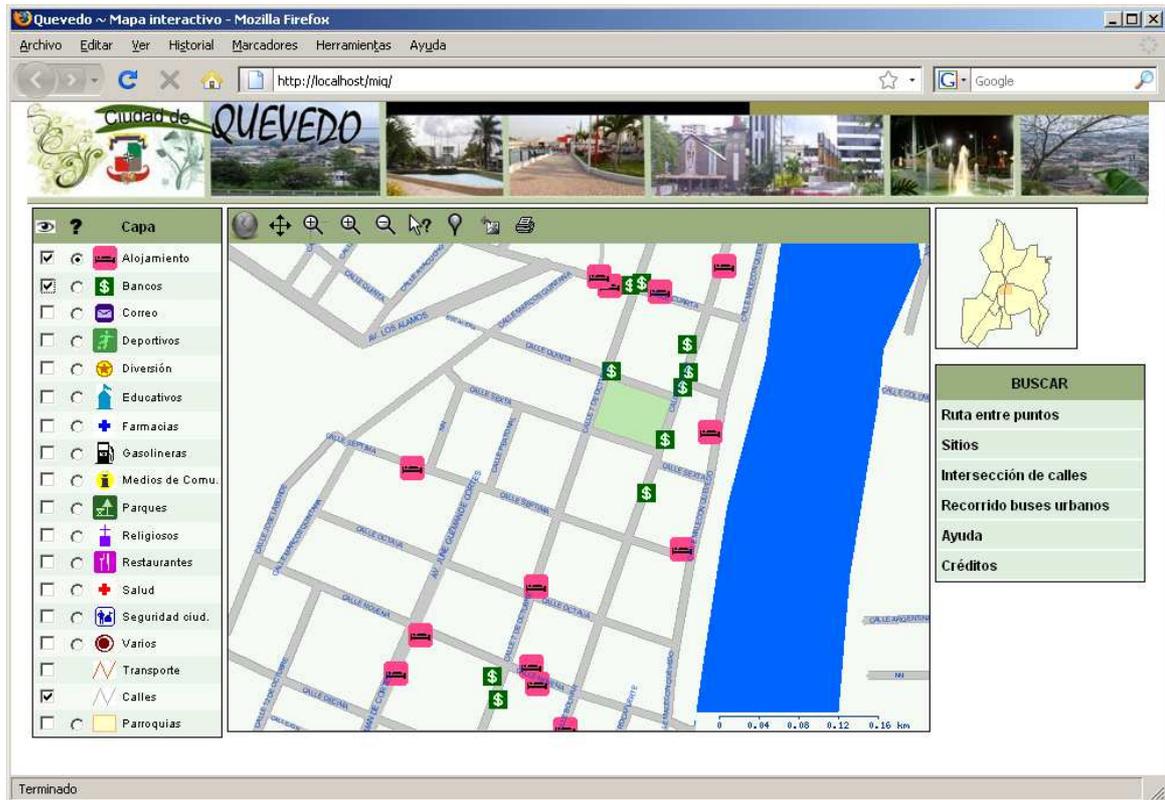


Figura 6.3. Centro de la ciudad de Quevedo con las capas alojamiento y bancos visible y ocultando las parroquias.

6.4. BUSCAR RUTA

Como se mencionara anteriormente, el software desarrollado permite encontrar la ruta óptima (basada en la distancia) entre dos puntos de la ciudad. Así, la figura 6.4 muestra la ruta más corta para ir desde la Asociación de Discapitados hasta el Instituto Ecuatoriano de Seguridad Social (IESS) de Quevedo, tanto a pie (figura 6.4) como en carro (figura 6.5), siendo importante observar que en el primer caso la ruta considera el puente colgante que es transitable sólo por peatones.

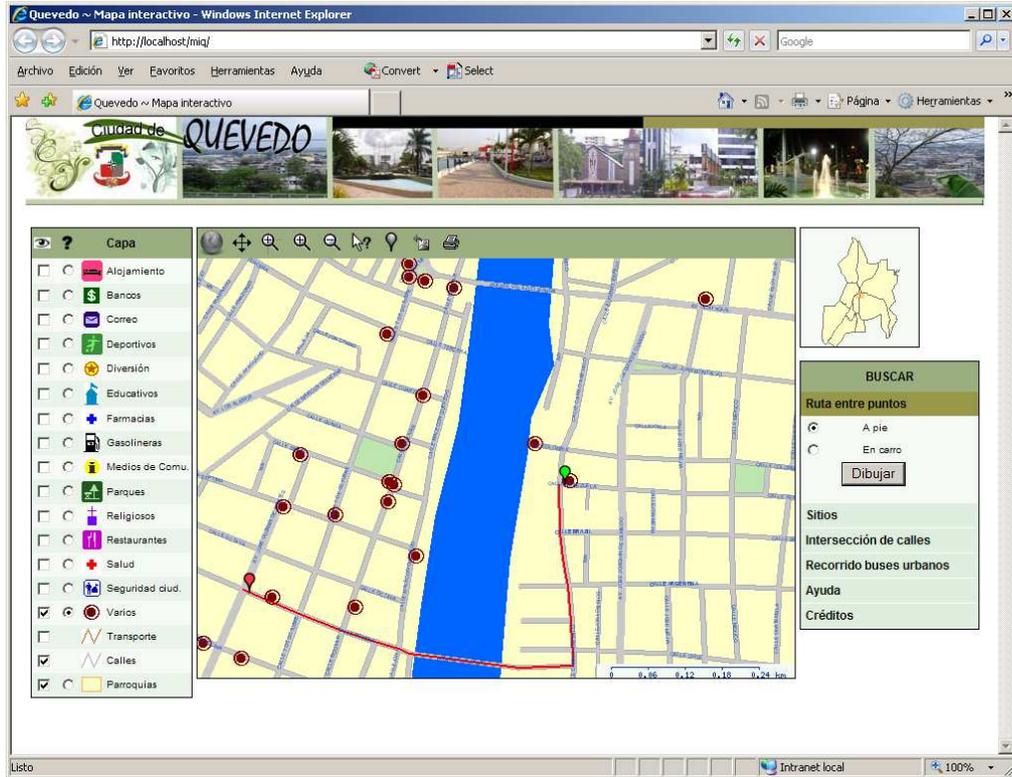


Figura 6.4. Ruta más corta a pie de la Asociación de Discapitados al IESS.

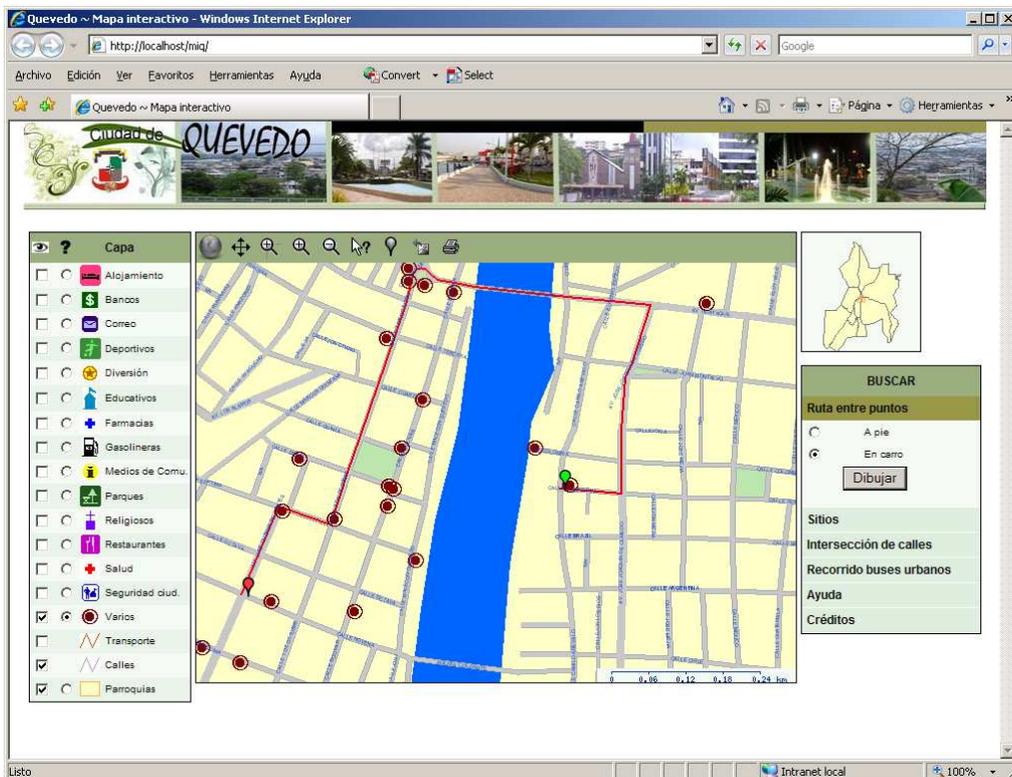


Figura 6.5. Ruta más corta en carro de la Asociación de Discapitados al IESS.

Esta operación de búsqueda de rutas óptimas se realiza a nivel de la base datos y haciendo uso del algoritmo de Edsger Dijkstra, que según las pruebas realizadas con la red de calles de la ciudad de Quevedo, resultó ser el que permite encontrar la solución en un menor tiempo en comparación con A star (A*). La tabla 6.1 contiene los resultados de las pruebas efectuadas para ocho rutas diferentes, cada una de las cuales tiene como punto de partida el Municipio de Quevedo y como destino un lugar diferente por cada una de las parroquias; los valores incluido en ella corresponden al tiempo (en milisegundos) requerido para encontrar la ruta correspondiente, tanto para recorridos peatonales como vehiculares.

Ruta (Origen - Destino)	Tiempos (milisegundos)			
	PEATONAL		VEHICULAR	
	Dijkstra	A*	Dijkstra	A*
Municipio - Recinto ferial	22,5	25,6	24,0	24,2
Municipio - Esc.Arnulfo Chávez	22,9	24,1	24,4	24,6
Municipio - Terminal terrestre	22,9	23,3	23,7	23,5
Municipio - Iglesia Santísima Trinidad	23,1	23,2	23,7	24,2
Municipio - Colegio 7 de Octubre	23,4	23,6	23,9	24,0
Municipio - Estación Línea 4 (El Desquite)	23,0	23,5	23,6	24,3
Municipio - Mercado de Promejoras	22,7	22,7	24,0	23,9
Municipio - U.Técnica de Babahoyo	23,6	23,8	24,2	24,2
PROMEDIO	23,0	23,7	23,9	24,1

Tabla 6.1. Comparación de algoritmos para encontrar la ruta óptima para varios pares de puntos (lugares).

Como se mencionó en la descripción de la metodología, la red de calles fue creada primeramente en ArcGIS como archivo shapefile. Esto permitió crear una geodatabase a la que se importó dicha red, y haciendo uso de la extensión Network Analyst fue posible realizar la búsqueda de rutas con la finalidad de detectar y corregir errores. Por ejemplo, en las figuras 6.6 y 6.7 se muestran las rutas óptimas, tanto a pie (a) como en carro (b), para el mismo caso ilustrado en las figura 6.4 y 6.5.

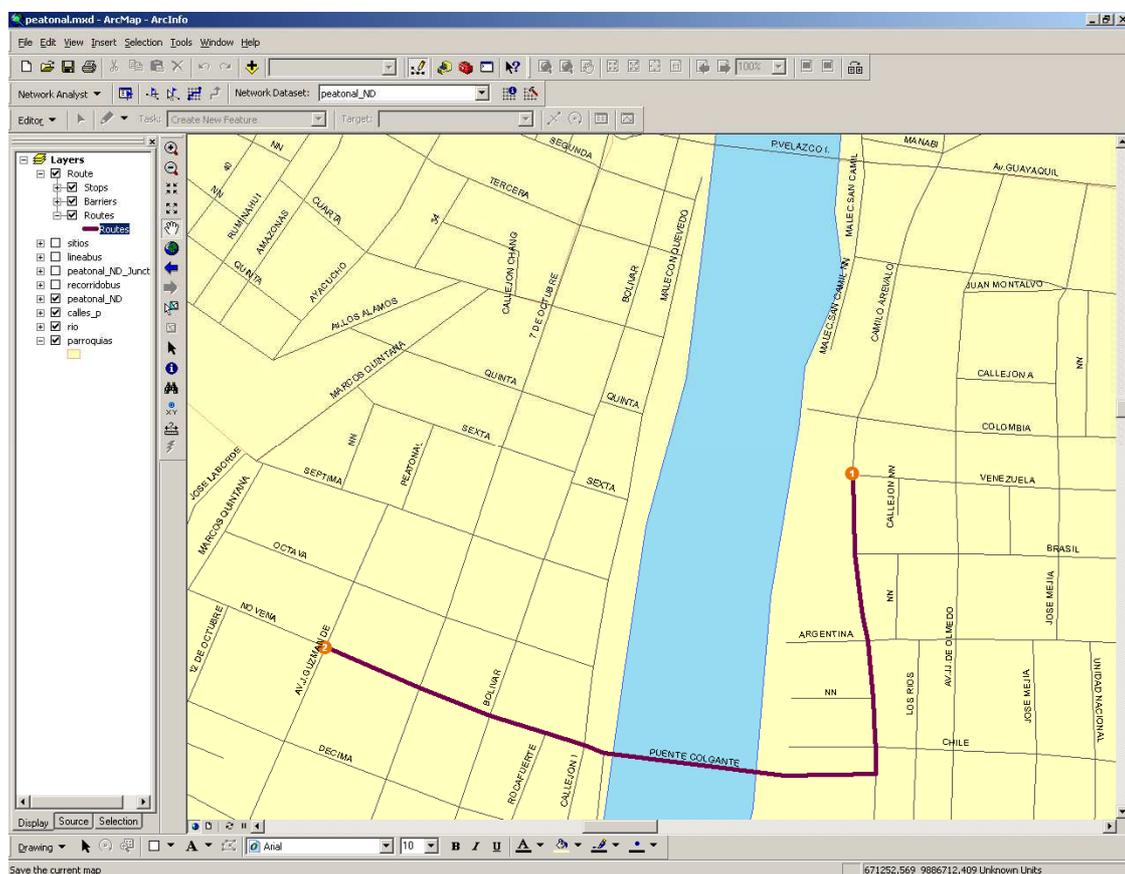


Figura 6.6. Ruta más corta a pie de la Asociación de Discapitados al IESS utilizando ArcGIS.

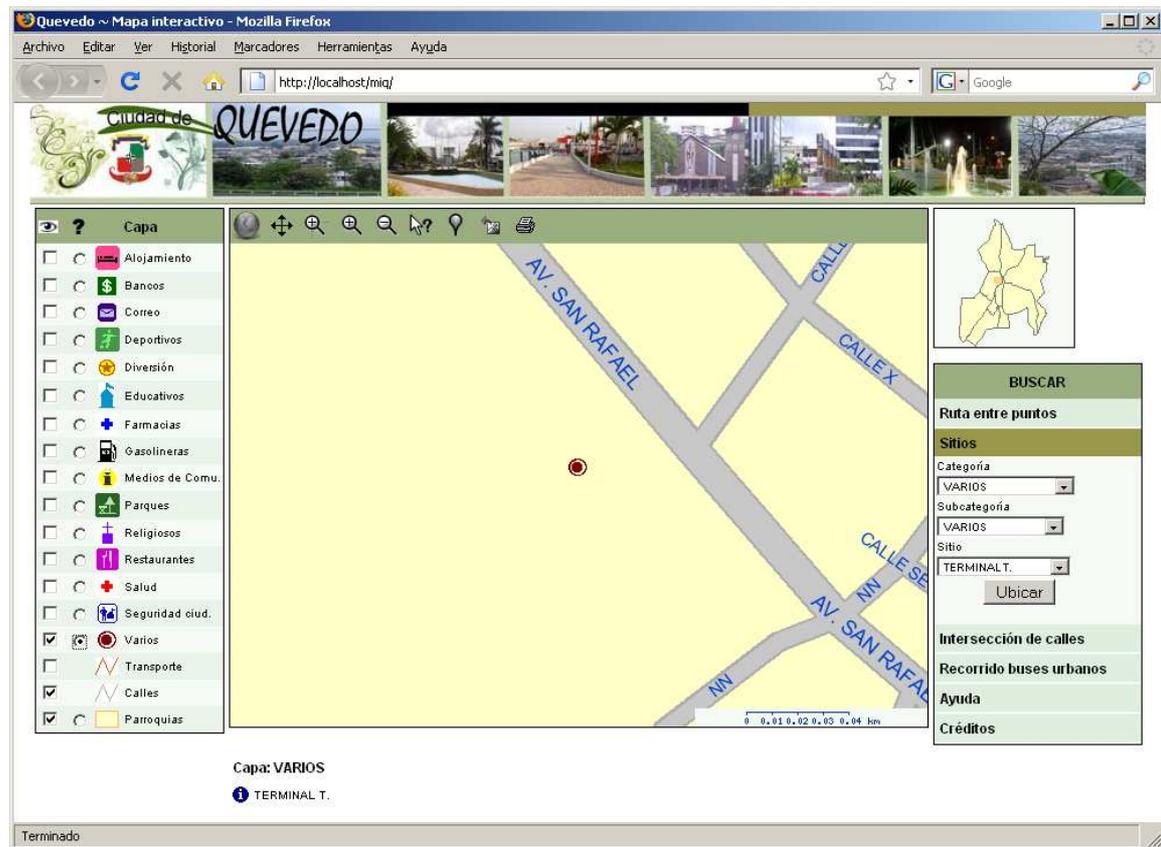


Figura 6.8. Ubicación en el mapa del Terminal Terrestre de Quevedo.

6.6. BUSCAR INTERSECCIÓN DE CALLES

De manera similar a los sitios, desde la sección Buscar es posible localizar intersecciones de calles (si existen), indicando previamente las dos calles que conforman dicha intersección. La figura 6.9 es el resultado obtenido al buscar la intersección de las calles 7 de Octubre y Quinta.

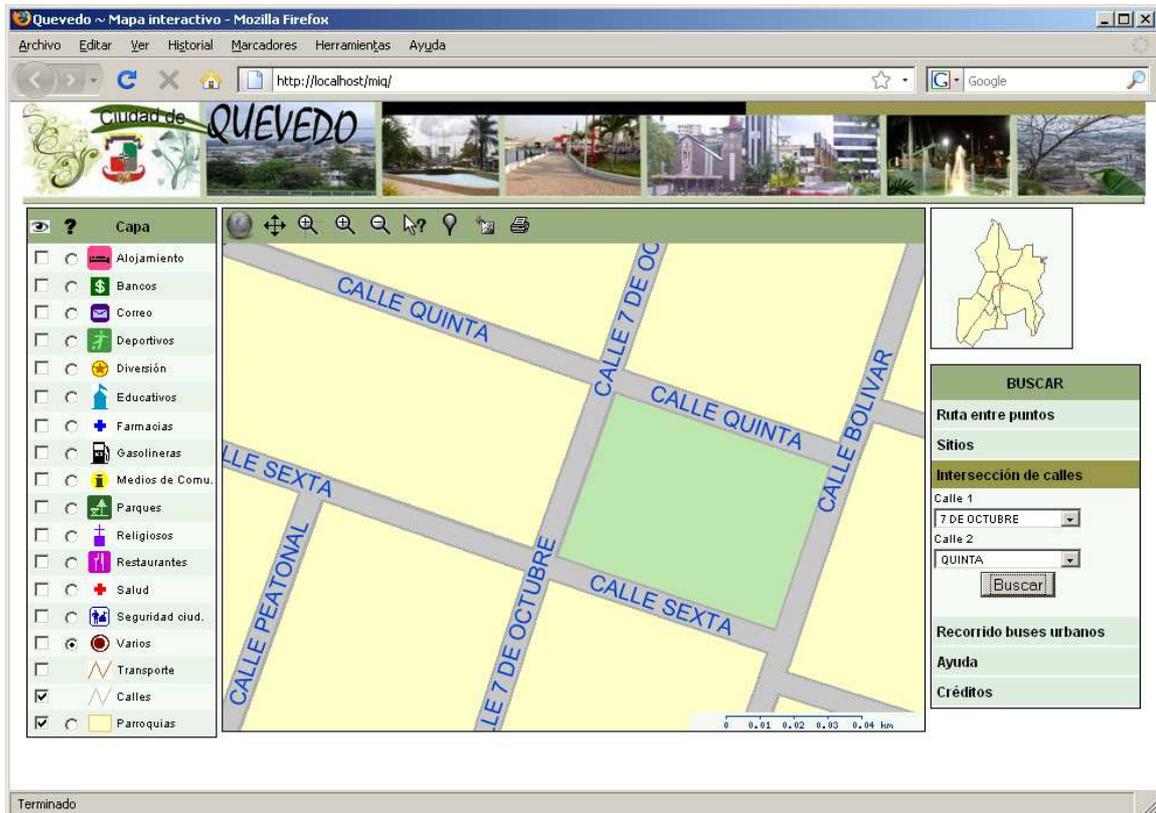


Figura 6.9. Intersección de calles 7 de Octubre y Quinta.

6.7. MOSTRAR RECORRIDO DE BUS

Teniendo como base la red de calles de la ciudad, también se incluye en el mapa los recorridos que realizan los buses urbanos, para lo que es menester indicar la línea deseada ya que sólo se podrá visualizar un recorrido a la vez, como se puede observar en la figura 6.10.

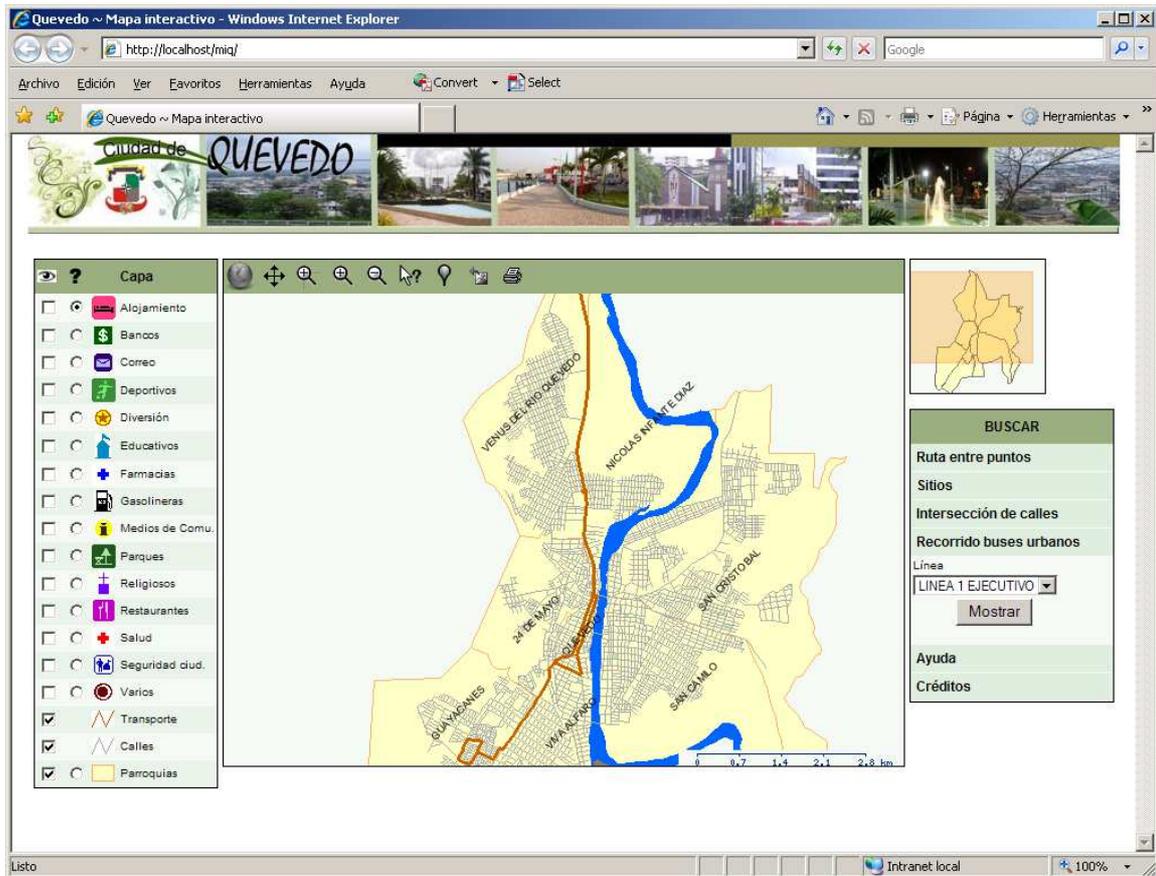


Figura 6.10. Recorrido de buses urbanos (Línea 1 ejecutivo).

6.8. MOSTRAR INFORMACIÓN DE SITIO

Además de ubicar en el mapa los sitios de interés, el usuario puede seleccionar una capa, escoger la herramienta identificar y haciendo clic sobre alguna de las entidades mostradas conocer su nombre y desplegar información adicional, en la que se incluye el nombre completo, detalles según la categoría a la que pertenece y fotos del lugar junto con controles para navegar entre ellas. En la figura 6.11 se presenta el resultado de hacer clic en la opción para mostrar más información del sitio que en este caso es la plaza central de la parroquia San Camilo.

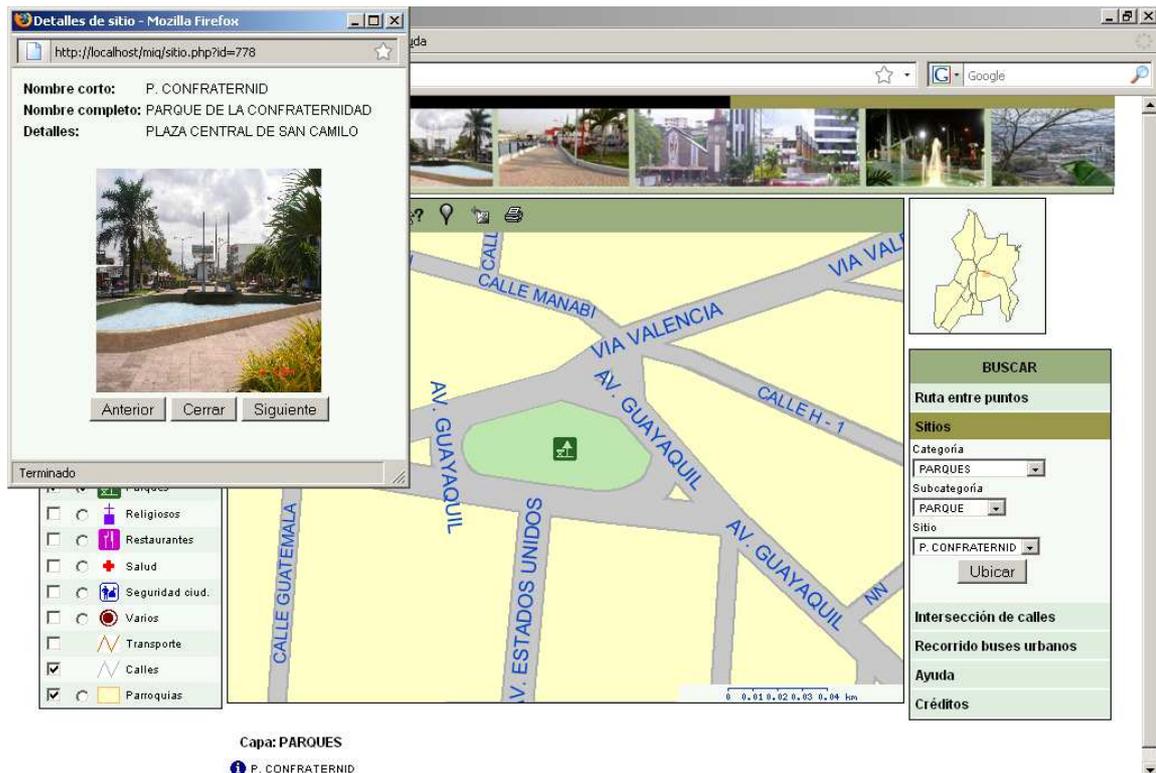


Figura 6.11. Visualización de información del Parque de la Confraternidad.

6.9. IMPRIMIR MAPA

Luego de que el usuario personalice el mapa según su necesidad (mostrar/ocultar capas, buscar ruta, etc.), puede proceder a imprimirlo. Para el efecto, después de que haya hecho clic en la herramienta Imprimir, se desplegará una vista previa del resultado que obtendrá impreso, similar al que se aprecia en la figura 6.12 en donde se han hecho visibles algunas capas con un acercamiento al Parque Central de Quevedo y sectores aledaños.

7. CONCLUSIONES

En el Ecuador, las aplicaciones de tipo Web Mapping han tenido acogida principalmente en los últimos años sin que la ciudad de Quevedo sea la excepción, en donde incluso personas e instituciones que tienen relación con el empleo de los Sistemas de Información Geográfica desconocen las bondades que el Web Mapping ofrece y el provecho que podrían obtener.

En muchos casos, el uso del Web Mapping y/o de una base de datos con capacidad de manejo de información espacial, ha estado limitado a grandes empresas debido a los gastos requeridos. Esta situación ha empezado a cambiar con el uso de Software Libre, que por tener una licencia de comercialización gratuita, permite disminuir considerablemente los costos, lo cual no implica que los productos así creados sean de mala calidad. No obstante, varias herramientas Open Source presentan como desventaja la dificultad para instalarlas y configurarlas.

Aprovechando las bondades que brinda el software de libre distribución se ha implementado un Mapa Interactivo para la ciudad Quevedo basado en el Web, utilizando para ello sistema operativo (Linux Fedora Core), servidor Web (Apache), lenguaje de programación (PHP), servidor de mapas (MapServer), sistema de gestión de base de datos (PostgreSQL). Estas herramientas han proporcionado las funcionalidades necesarias para alcanzar los objetivos planteados.

El mapa interactivo desarrollado permite visualizar la división política-administrativa del cantón, su red de calles, recorridos de buses y lugares de interés (organizados según categorías). Además de la navegación en el mapa es posible realizar la búsqueda de sitios, intersecciones de calles y rutas más cortas para ir de una ubicación de la ciudad a otra.

Toda la información requerida ha sido integrada en una base de datos espacial, creada con PostgreSQL e incluyendo la extensión PostGIS. PostGIS ha permitido el tratamiento de los datos geográficos, pero para llevar a cabo la búsqueda de rutas ha sido necesario incorporar capacidades de ruteo, empleando para el efecto las funciones de pgRouting.

En base a las pruebas realizadas se determinó que el algoritmo más adecuado para la búsqueda de rutas de menor recorrido en la red de calles de Quevedo es el Algoritmo de Dijkstra, aunque en comparación con A star, las diferencias de tiempos son pequeñas (un milisegundo aproximadamente para los escenarios considerados). Además, conviene tener en cuenta que A star es ideal para redes con una gran cantidad de nodos y arcos, ya que emplea funciones heurísticas para encontrar la solución, situación que no corresponde a la red utilizada (6443 arcos, 4200 nodos).

Este tipo de rutas (más corta, a pie y en carro) son las únicas que pueden buscarse, pero el diseño de la base de datos está concebido para permitir la ampliación futura de este software y considerar también las rutas más rápidas. Esto es posible porque a la red elaborada se le incorporaron características de las

calles como tipo y rodadura, y conociendo las velocidades de circulación se podrá determinar el tiempo para atravesar cada segmento. De manera similar, la base de datos y las funciones incorporadas en ella permitirán posteriormente la implementación de otras opciones como búsqueda de rutas entre varios puntos, definición de áreas de captación, entre otras.

A pesar de que la Unidad de Investigación Informática para el Ordenamiento Territorial y Ambiental de la UTEQ ha facilitado parte de los datos necesarios para la elaboración del presente trabajo, estos no han sido suficientes, evidenciando con ello dificultades en la adquisición de datos, en especial por el tiempo que se necesitó para recorrer la ciudad levantando los lugares de interés junto con su información alfanumérica y elaborar la red calles.

La presente aplicación ha sido implementada bajo el ambiente del sistema operativo Linux, pero gracias a las bondades del software utilizado también puede funcionar en MS Windows, para lo cual será necesario obtener las versiones de las herramientas correspondientes para tal plataforma y revisar las referencias a directorios.

8. RECOMENDACIONES

- En el Anexo A se describen los pasos para la instalación y configuración de la aplicación desarrollada. Sin embargo, en el caso de utilizar Linux, si se quieren implementar otras opciones de ruteo (como áreas de captación, por ejemplo) será necesario modificar ciertas opciones o compilar/installar librerías adicionales (como CGAL), por lo que para mayores detalles deberá referirse a la documentación o archivos correspondientes (típicamente Readme o Install).
- Es menester tener en cuenta que para que aplicaciones de este tipo puedan funcionar de mejor forma deberían instalarse, junto con todo el software necesario, en un computador o servidor con las mejores características posibles y disponer de una conexión a Internet con un ancho de banda no limitado o bajo, según las propias posibilidades.
- Cuando el usuario navega en el mapa y genera una nueva vista del mismo se crean archivos temporales que corresponden a la imagen mostrada. El tamaño de estos archivos no es muy grande (aproximadamente de 1KB a 25 KB) pero el constante uso de la aplicación provoca su acumulación, pudiendo llegar a ocupar un espacio considerable del disco duro, siendo necesario proceder cada cierto tiempo (según la carga de peticiones que tenga el sitio Web) a eliminarlos. Su eliminación puede efectuarse desde el directorio en que son ubicados (por ejemplo, “../Apache/htdocs/tmp”, “/var/www/html/tmp”) según se especifique en el archivo de mapa.

- En caso de necesitar actualizar la información almacenada en la base de datos se puede hacer uso de sentencias SQL o utilizar programas como gvSIG que permite visualizar y manipular tanto las entidades geográficas como sus atributos.
- El mapa interactivo de Quevedo es accesible desde Internet siendo importante acompañarlo de un cortafuegos a fin de proteger al servidor en que esté instalado de posibles ataques de Hackers. En este sentido, se puede aprovechar una de las herramientas de seguridad que brindan las distribuciones de Linux, como lo es Iptables, que a través de un conjunto de comandos permite crear reglas para aceptar o rechazar peticiones según sea el caso particular.

BIBLIOGRAFIA

- STROBL, J.; RESL, R. (2005): Orientación e Introducción en Sistemas de Información Geográfica. UNIGIS Professional para América Latina, pp. 67-70 y pp. 81-93.
- GUTIERREZ, J.; GOULD, M. (2000): SIG: Sistemas de Información Geográfica. Madrid, Síntesis, pp. 225-234.
- BOSQUE, J. (1992): Sistemas de Información Geográfica, Primera edición. Madrid, p. 207.
- STROBL, J.; RESL, R. (2000): Fundamentos: Modelos y Estructuras de Datos espaciales. UNIGIS Professional para América Latina, p. 63.
- COMELLAS, F. et al. (2001): Matemática Discreta. Barcelona, Ediciones UPC, 336 p.
- STROBL, J.; RESL, R. (2000): Fundamentos: Modelos y Estructuras de Datos espaciales. UNIGIS Professional para América Latina, p. 73.
- GUTIERREZ, J.; GOULD, M. (2000): SIG: Sistemas de Información Geográfica. Madrid, Síntesis, pp. 185-186.
- LONGLEY, P. et al. (2001): Geographic Information Systems And Science. John Wiley & Sons, pp.13-17.
- NEUMANN, A. (2008): Web Mapping and Web Cartography. En S. Shekar y H. Xiong (Eds.), Encyclopedia of GIS. Springer, pp. 1261-1262.
- GREEN, D.; BOSSOMAIER, T. (2002): Online GIS and Spatial Metadata. Londres, Taylor & Francis, pp. 13-26.
- KRAAK, M.; BROWN, A. (2001): Web Cartography. Londres, Taylor & Francis, pp. 1-37.
- NEUMANN, A. (2008): Web Mapping and Web Cartography. En S. Shekar y H. Xiong (Eds.), Encyclopedia of GIS. Springer, pp. 1263-1264.
- GEOGRAMA (2004): Optimización de un servidor de mapas en Internet. Disponible en: <http://www.cartesia.org/>.
- MITCHELL, T. (2005): Web Mapping Illustrated. O'Reilly, pp. 5-11.
- NEUMANN, A. (2008): Web Mapping and Web Cartography. En S. Shekar y H. Xiong (Eds.), Encyclopedia of GIS. Springer, pp. 1266-1267.

- ESRI (2004): ArcIMS 9 Architecture and Functionality. ESRI White Paper. Disponible en: <http://www.esri.com>.
- KRAAK, M.; BROWN, A. (2001): Web Cartography. Londres, Taylor & Francis, pp. 73-85.
- GREEN, D.; BOSSOMAIER, T. (2002): Online GIS and Spatial Metadata. Londres, Taylor & Francis, pp. 27-44.
- TITTEL, E.; BURMEISTER, M. (2005): HTML 4 For Dummies, Quinta edición. Wiley Publishing, 432 p.
- TITTEL, E.; NOBLE, J. (2008): HTML, XHTML & CSS For Dummies, Sexta edición. Wiley Publishing, 384 p.
- POSADAS, M. (2000): HTML dinámico, modelos de objetos y JavaScript. Grupo EIDOS, pp. 9-14.
- GOODMAN, D.; MORRISON, M. (2004): JavaScript Bible, Quinta edición. Wiley Publishing, 1272 p.
- GARRETT, J. (2005, Febrero): Ajax: A New Approach to Web Applications. Disponible en: <http://www.adaptivepath.com/>.
- ZAKAS, N.; MCPEAK, J.; FAWCETT, J. (2006): Profesional Ajax. Anaya Multimedia, pp 23-74.
- NETCRAFT (2008, Octubre): October 2008 Web Server Survey. Disponible en: <http://news.netcraft.com/>.
- APACHE (2008): Apache HTTP Server Project. Disponible en: <http://httpd.apache.org/>.
- KABIR, M. (2003): LA BIBLIA DE SERVIDOR APACHE 2. Madrid, Anaya Multimedia, 862 p.
- SEGUY, D. (2008, Noviembre): PHP statistics for October 2008. Disponible en: <http://www.nexen.net/>.
- ACHOUR, M. et al. (2008): PHP Manual. Philip Olson (Ed.), PHP Documentation Group.
- COLL, E. (2005): Introducción a la publicación de cartografía en Internet. Universidad Politécnica de Valencia, pp. 12-14.
- MAPSERVER (2008). Welcome to MapServer. Disponible en: <http://mapserver.gis.umn.edu/>.

- GEOSERVER: GeoServer Features. Disponible en: <http://geoserver.org/>.
- KROPLA, B.(2005): Beginning MapServer: Open Source GIS Development. New York, Apress, 418 p.
- MAPSERVER: Documentation. Disponible en: <http://mapserver.gis.umn.edu/docs>.
- BY, R.A. de (2001): Principles of Geographic Information Systems: An Introductory Textbook. International Institute for Aerospace Survey and Earth Sciences (ITC), p. 52.
- COMPTE, M; STROBL, J.; RESL, R. (2007): Bases de datos espaciales. UNIGIS Professional para América Latina, pp. 111-113.
- RIGAUX, P.; SCHOLL, M.; VOISARD, A. (2001): Spatial Databases: With Application to GIS, Segunda edición. Morgan Kauffman, pp. 25-26.
- COMPTE, M.; MOLINA, J.; TURBAU, E.: d-Ruta, un sistema móvil de información turística. En I Jornadas de SIG Libre. Disponible en: <http://www.sigte.udg.es/>.
- MIRA, J.; NAVARRO, J.; RAMON, A.: SIGUA: SIG libre para la gestión del suelo de la Universidad de Alicante. En I Jornadas de SIG Libre. Disponible en: <http://www.sigte.udg.es/>.
- GORNI, D. et al.: Sistema de Información Geográfica para Web con informaciones espaciales (geográficas) de output y input para Expediciones. En I Jornadas de SIG Libre. Disponible en: <http://www.sigte.udg.es/>.
- HALL, G.; ALPERIN, J.; LEON, S. (2007, Mayo): El uso de Internet con software libre y fuentes abiertas espaciales para colaborar en la toma de decisiones. En XI Conferencia Iberoamericana en Sistemas de Información Geográfica. Buenos Aires, Argentina.
- ANDERSON, J.; KUNGYS, V.; STEINBAUER, J.: Ride the City: Disponible en: <http://www.ridethecity.com>.
- PostgreSQL: About. Disponible en <http://www.postgresql.org>.
- COMPTE, M; STROBL, J.; RESL, R. (2007): Bases de datos espaciales. UNIGIS Professional para América Latina.
- RAMSEY, P.; MARTIN, M.: Manual PotsGIS. Disponible en <http://postgis.refractory.net>.

- MICROSOFT CORPORATION (2003): Microsoft Solutions Framework version 3.0 Overview. Disponible en <http://www.microsoft.com/msf/>.
- MICROSOFT TECHNET (2004): Guía de entrega. Disponible en <http://www.microsoft.com/latam/technet/articulos/wireless/ap03.mspx>
- MICROSOFT CORPORATION (2002): MSF Process Model v. 3.1. Disponible en <http://www.microsoft.com/msf/>.
- LARMAN, Craig (1999): UML y Patrones, Introducción al Análisis y Diseño Orientado a Objetos, Primera Edición. México, Prentice Hall, p. 8.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar (2000): El Lenguaje Unificado de Modelado. España, Addison Wesley, 464 p.
- CONALLEN, Jim (2002): Building Web Applications with UML, Second Edition. Addison Wesley, 496 p.
- NAIBURG, E.; MAKSIMCHUK, R. (2001): UML for Database Design, First Edition. Addison Wesley, 320 p.

ANEXO A. MANUAL DE INSTALACION

A.1. REQUERIMIENTOS DEL SISTEMA

En lo correspondiente al hardware, el Mapa Interactivo de Quevedo debe ser instalado en un servidor o en un computador que tenga características similares a las señaladas a continuación:

- Procesador Core.2 DUO de 2.66 GHz.
- Memoria RAM 2GB.
- Disco duro Samsung de 320 GB.
- Tarjeta de red Ethernet configurada con una dirección IP estática y pública.
- Conexión a Internet.

Además, se debe tener instalado el siguiente software, con las versiones indicadas o posteriores.

- Sistema operativo Linux (probado en Fedora Core 6 y 8).
- Servidor Web Apache 2.2.
- Lenguajes de programación PHP 5.2.
- Servidor de Mapas MapServer 5.0 con soporte para PHP MapScript.
- Sistema de gestión de base de datos PostgreSQL 8.2, con extensión espacial PostGIS.

A.2. INSTALACIÓN

Para poner en marcha la aplicación desarrollada se requiere compilar e instalar librerías adicionales, copiar algunos archivos y crear la base de datos. Las dos primeras tareas se describen a continuación mientras que la tercera en la sección siguiente.

Nota: Recuerde cambiar los nombres de los directorios según sea su caso.

- Compilación e instalación de BGL (Boost Graph Library):

```
# cd /directorio_BGL
```

```
# ./configure
```

```
# make
```

```
# make install
```

- Instalación de cmake (solo en caso de no estar instalado):

```
# rpm -i cmake-version.rpm
```

- Compilación e instalación de pgRouting:

```
# cd /directorio_pgRouting
```

```
# cmake .
```

```
# make
```

```
# make install
```

- Desempaquetar los archivos del código fuente (archivos .php) y los archivos de configuración (de mapa, símbolos, etc.), y colocarlos en las carpetas que corresponda:

```
# mkdir /directorio_temporal
```

```
# cp /media/CD/miq.tar.gz /directorio_temporal
```

```
# cp /media/CD/miq_data.tar.gz /directorio_temporal
```

```
# cd /directorio_temporal
```

```
# gunzip miq.tar.gz
```

```
# gunzip miq_data.tar.gz
```

```
# cd /directorio_Web
```

```
# tar -xf /directorio_temporal/miq.tar
```

```
# cd /directorio_datos
```

```
# tar -xf /directorio_temporal/miq_data.tar
```

- En forma adicional se debe crear una carpeta para los archivos temporales generados por MapServer, con permisos de escritura para el usuario Apache:

```
# mkdir /directorio_Web/tmp
```

```
# chown apache:apache /directorio_Web/tmp
```

A.3. CREACIÓN DE LA BASE DE DATOS

La creación de la base de datos comprende algunas tareas realizadas mediante la ejecución de comandos y sentencias SQL; para diferenciarlas se ha antepuesto los signos “\$” y “miq=>” respectivamente. A continuación se describen todos los pasos necesarios y que deben realizarse desde una ventana de terminal con el usuario de PostgreSQL (comando “su postgres”).

- Crear la base de datos con extensión espacial PostGIS y funcionalidades de ruteo:

```
$ createdb -U postgres -E SQL_ASCII miq
```

```
$ createlang -U postgres plpgsql miq
```

```
$ psql -U postgres -f /opt/PostgresPlus/8.3/share/lwpostgis.sql miq
```

```
$ psql -U postgres -f /opt/PostgresPlus/8.3/share/spatial_ref_sys.sql miq
```

```
$ psql -U postgres -f /usr/share/postlbs/routing_core.sql miq
```

```
$ psql -U postgres -f /usr/share/postlbs/routing_core_wrappers.sql miq
```

```
$ psql -U postgres -f /usr/share/postlbs/routing_topology.sql miq
```

- Creación del usuario con el que se accede a la base de datos y modificación de los permisos necesarios:

```
$ psql -U postgres miq
```

```
miq=# CREATE ROLE usrmiq LOGIN
```

```
ENCRYPTED PASSWORD 'suclave'
```

```
NOSUPERUSER NOINHERIT NOCREATEDB NOCREATEROLE;
```

```
miq=# ALTER TABLE geometry_columns OWNER TO usrmig;
```

```
miq=# GRANT ALL ON TABLE geometry_columns TO usrmig;
```

```
miq=# GRANT SELECT ON TABLE geometry_columns TO public;
```

```
miq=# \q
```

- Cargar los datos necesarios a la base:

```
$ shp2pgsql -c ../directorio_datos/miq/data/parroquias.shp parroquias | psql -d miq -U usrmig
```

```
$ shp2pgsql -c ../directorio_datos/miq/data/calles.shp calles | psql -d miq -U usrmig
```

```
$ shp2pgsql -c ../directorio_datos/miq/data/lineabus.shp lineabus | psql -d miq -U usrmig
```

```
$ shp2pgsql -c ../directorio_datos/miq/data/sitios.shp sitios | psql -d miq -U usrmig
```

```
$ shp2pgsql -c ../directorio_datos/miq/data/recorridobus.shp recorridobus | psql -d miq -U usrmig
```

- Completar la creación de la red de calles:

```
$ psql -U usrmig miq
```

```
miq=> ALTER TABLE calles ADD COLUMN costo double precision;
```

```
miq=> ALTER TABLE calles ADD COLUMN costoinverso double precision;
```

```
miq=> update calles set costo=longitud, costoinverso=longitud where
sentido=0;
```

```
miq=> update calles set costo=longitud, costoinverso=longitud+100000
where sentido=1;
```

```
miq=> update calles set costo=longitud+100000, costoinverso=longitud
where sentido=-1;
```

```
miq=> ALTER TABLE calles ADD COLUMN source integer;
```

```
miq=> ALTER TABLE calles ADD COLUMN target integer;
```

```
miq=> SELECT assign_vertex_id('calles', 5, 'the_geom', 'gid');
```

```
miq=> ALTER TABLE vertices_tmp OWNER TO usrmq;
```

```
miq=> ALTER TABLE calles RENAME COLUMN source TO origen;
```

```
miq=> ALTER TABLE calles RENAME COLUMN target TO destino;
```

- Creación de vistas para realizar la búsqueda de rutas de las dos formas permitidas (peatonal y vehicular):

```
miq=> CREATE OR REPLACE VIEW peatonal AS
```

```
SELECT calles.gid, calles.nombre, calles.the_geom, calles.origen as
source, calles.destino as target, calles.longitud AS length, calles.x1,
calles.y1, calles.x2, calles.y2 FROM calles;
```

```
miq=> ALTER TABLE peatonal OWNER TO usrmq;
```

```
miq=> CREATE OR REPLACE VIEW vehicular AS
```

```
SELECT calles.gid, calles.nombre, calles.the_geom, calles.origen as
source, calles.destino as target, calles.costo as length, calles.costoinverso as
reverse_cost, calles.x1, calles.y1, calles.x2, calles.y2 FROM calles
```

```
WHERE calles.tipoacceso = 0;
```

```
miq=> ALTER TABLE vehicular OWNER TO usrmiq;
```

- Realizar los ajustes necesarios a la tabla de recorridos de buses y crear una vista para obtener sólo el recorrido de la línea deseada:

```
miq=> ALTER TABLE recorridobus DROP CONSTRAINT
enforce_srid_the_geom;
```

```
miq=> ALTER TABLE recorridobus DROP CONSTRAINT
enforce_geotype_the_geom;
```

```
miq=> ALTER TABLE recorridobus DROP CONSTRAINT
enforce_dims_the_geom;
```

```
miq=> ALTER TABLE recorridobus DROP COLUMN the_geom;
```

```
miq=> ALTER TABLE recorridobus DROP COLUMN gid;
```

```
miq=> ALTER TABLE recorridobus ADD CONSTRAINT
recorridobus_idcalle_fkey FOREIGN KEY (idcalle) REFERENCES calles
(gid);
```

```
miq=> ALTER TABLE recorridobus ADD CONSTRAINT
recorridobus_idlineabus_fkey FOREIGN KEY (idlineabus) REFERENCES
lineabus (gid);
```

```
miq=> CREATE OR REPLACE VIEW transporte AS
```

```
SELECT rb.idlineabus, c.gid, c.the_geom
```

```
FROM recorridobus rb, calles c
```

```
WHERE c.gid = rb.idcalle;
```

```
ALTER TABLE transporte OWNER TO usrmiq;
```

- Crear y llenar las tablas adicionales:

```
miq=> CREATE TABLE tipocalle(  
    idtipo serial NOT NULL,  
    nombre character varying(10),  
    CONSTRAINT tipocalle_pkey PRIMARY KEY (idtipo)  
) WITHOUT OIDS;
```

```
miq=> ALTER TABLE tipocalle OWNER TO usrmig;
```

```
miq=> insert into tipocalle (nombre) values('CALLE');
```

```
miq=> insert into tipocalle (nombre) values('AVENIDA');
```

```
miq=> insert into tipocalle (nombre) values('VIA');
```

```
miq=> insert into tipocalle (nombre) values('CALLEJON');
```

```
miq=> insert into tipocalle (nombre) values('ESCALERA');
```

```
miq=> insert into tipocalle (nombre) values('PUENTE');
```

```
miq=> CREATE TABLE rodadura(  
    idrodadura serial NOT NULL,  
    nombre character(10),  
    CONSTRAINT rodadura_pkey PRIMARY KEY (idrodadura)  
) WITHOUT OIDS;
```

```
miq=> ALTER TABLE rodadura OWNER TO usrmig;
```

```
miq=> insert into rodadura (nombre) values('ASFALTO');
```

```
miq=> insert into rodadura (nombre) values('ADOQUIN');
```

```
miq=> insert into rodadura (nombre) values('ESCALERA');
```

```
miq=> insert into rodadura (nombre) values('HORMIGON');
miq=> insert into rodadura (nombre) values('LASTRE');
miq=> insert into rodadura (nombre) values('TIERRA');
miq=> ALTER TABLE calles ADD CONSTRAINT calles_idrodadura_fkey
FOREIGN KEY (idrodadura) REFERENCES rodadura (idrodadura);
miq=> ALTER TABLE calles ADD CONSTRAINT calles_idtipo_fkey
FOREIGN KEY (idtipo) REFERENCES tipocalle (idtipo);

miq=> CREATE TABLE categorias(
  idcategoria serial NOT NULL,
  nombre character(15),
  CONSTRAINT categorias_pkey PRIMARY KEY (idcategoria)
) WITHOUT OIDS;
miq=> ALTER TABLE categorias OWNER TO usrmiq;
miq=> insert into categorias (nombre) values('ALOJAMIENTO');
miq=> insert into categorias (nombre) values('BANCOS');
miq=> insert into categorias (nombre) values('CORREO');
miq=> insert into categorias (nombre) values('DEPORTIVOS');
miq=> insert into categorias (nombre) values('DIVERSION');
miq=> insert into categorias (nombre) values('EDUCATIVOS');
miq=> insert into categorias (nombre) values('FARMACIAS');
miq=> insert into categorias (nombre) values('GASOLINERAS');
miq=> insert into categorias (nombre) values('MEDIOS DE COMUN');
miq=> insert into categorias (nombre) values('RELIGIOSOS');
miq=> insert into categorias (nombre) values('RESTAURANTES');
```

```

miq=> insert into categorias (nombre) values('SALUD');
miq=> insert into categorias (nombre) values('SEGURIDAD CIUDA');
miq=> insert into categorias (nombre) values('VARIOS');
miq=> insert into categorias (nombre) values('PARQUES');
miq=> ALTER TABLE sitios RENAME COLUMN idcategori TO idcategoria;
miq=> ALTER TABLE sitios ADD CONSTRAINT sitios_idcategoria_fkey
FOREIGN KEY (idcategoria) REFERENCES categorias (idcategoria);

```

```

miq=> CREATE TABLE fotos(
idsitio integer,
rutafoto character varying(100),
CONSTRAINT fotos_idsitio_fkey FOREIGN KEY (idsitio)
REFERENCES sitios (gid) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
) WITHOUT OIDS;
miq=> ALTER TABLE fotos OWNER TO usrmq;

```

Para llenar la tabla de fotos se pueden ejecutar sentencias como la siguiente:

```

miq=> insert into fotos values(27, 'directorio/imagen.JPG');

```

- Creación de índices:

```

miq=> CREATE INDEX parroquias_the_geom_idx ON parroquias USING
gist (the_geom);

```

```
miq=> CREATE INDEX calles_the_geom_idx ON calles USING gist  
(the_geom);
```

```
miq=> CREATE INDEX sitios_the_geom_idx ON sitios USING gist  
(the_geom);
```

```
miq=> CREATE INDEX lineabus_the_geom_idx ON lineabus USING gist  
(the_geom);
```

```
miq=> CREATE INDEX recorridobus_idcalle_idx ON recorridobus (idcalle);
```

```
miq=> CREATE INDEX recorridobus_idlineabus_idx ON recorridobus  
(idlineabus);
```

```
miq=> \q
```

ANEXO B. ARCHIVO DE MAPA miq.map

```
MAP
NAME "Quevedo"
UNITS meters
EXTENT 664986 9881628 677315 9893957
SIZE 450 450
IMAGETYPE PNG
IMAGECOLOR 245 250 245#255 255 240
SYMBOLSET "/home/mapdata/miq/etc/simbologia.sym"
FONTSET "/home/mapdata/miq/etc/letras.txt"
SHAPEPATH "/home/mapdata/miq/data"
#----- Objeto Web -----#
WEB
IMAGEPATH "/opt/fgs/www/htdocs/tmp/"
IMAGEURL "/tmp/"
MAXSCALEDENOM 100000
MINSCALEDENOM 900
END
#----- Mapa de referencia -----#
REFERENCE
IMAGE "/opt/fgs/www/htdocs/miq/img/mapref.png"
SIZE 120 120
EXTENT 667473 9882480 674550 9893060
STATUS ON
```

COLOR -1 -1 -1

OUTLINECOLOR 255 0 0

END

#----- Barra de escala -----#

SCALEBAR

STATUS embed

COLOR 0 50 150

OUTLINECOLOR -1 -1 -1

IMAGECOLOR 245 250 245

UNITS KILOMETERS

INTERVALS 4

SIZE 200 2

STYLE 1

POSITION LR

LABEL

TYPE bitmap

SIZE tiny

OFFSET 0 0

BUFFER 0

MINDISTANCE -1

MINFEATURESIZE -1

COLOR 0 50 150

SHADOWSIZE 0.5 0.5

BACKGROUNDSHADOWSIZE 1.5 1.5

PARTIALS TRUE

FORCE FALSE

END

END

#----- Leyenda -----#

LEGEND

STATUS on

IMAGECOLOR 255 255 255

KEYSIZE 20 15

KEYSPACING 5 5

POSITION II

TRANSPARENT off

LABEL

TYPE truetype

FONT arial

COLOR 0 0 0

SIZE 8

ANTIALIAS true

END

END

#----- Capa de parroquias -----#

LAYER

NAME "parroquias"

DATA "the_geom from parroquias"

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

```
STATUS default
TYPE polygon
CLASSITEM "nombre"
LABELITEM "nombre"
LABELMINSIZE 15000
TEMPLATE "t.html"
CLASS
NAME "PARROQUIAS"
LABEL
TYPE TRUETYPE
FONT "arial"
SIZE 7
POSITION lc
ANGLE 45
END
STYLE
COLOR 255 255 200
OUTLINECOLOR 255 180 110
END
END
END
#----- Capa rio: polygon -----#
LAYER
NAME "rio"
DATA "rio"
```

STATUS default

TYPE polygon

CLASS

NAME "RIO QUEVEDO"

STYLE

COLOR 0 100 255

OUTLINECOLOR -1 -1 -1

END

END

END

#----- Capa Islas: polygon -----#

LAYER

NAME "islas"

DATA "islas"

STATUS default

TYPE polygon

CLASS

STYLE

COLOR 149 118 96

OUTLINECOLOR -1 -1 -1

END

END

END

#----- Capa calles: line-----#

LAYER

```
NAME "calles"
STATUS on
TYPE line
CONNECTIONTYPE postgis
CONNECTION "user=x password=x dbname=miq host=localhost port=5432"
DATA "the_geom FROM calles"
TOLERANCE 5
TOLERANCEUNITS pixels
TEMPLATE "t.html"
LABELITEM "nombre"
LABELMAXSCALEDENOM 5000
SIZEUNITS meters
CLASS #general
NAME "CALLES"
MINSCALEDENOM 25000
STYLE
COLOR 175 175 175
SYMBOL "cartoline"
SIZE 10
END
END
CLASS #calles y puentes
EXPRESSION ([idtipo]=1 or [idtipo]=6)
MAXSCALEDENOM 25000
STYLE
```

COLOR 175 175 175

SYMBOL "cartoline"

SIZE 8

MINSIZE 1

END

STYLE

COLOR 200 200 200

SYMBOL "cartoline"

SIZE 6

END

LABEL

TYPE TRUETYPE

FONT "arial"

SIZE 5

POSITION cc

ANGLE auto

MINDISTANCE 150

COLOR 0 64 200

END

END

CLASS #avenidas y vías

EXPRESSION ([idtipo]=2 or [idtipo]=3)

MAXSCALEDENOM 25000

STYLE

COLOR 175 175 175

```
SYMBOL "cartoline"  
SIZE 13  
MINSIZE 1  
END  
STYLE  
COLOR 200 200 200  
SYMBOL "cartoline"  
SIZE 11  
END  
LABEL  
TYPE TRUETYPE  
FONT "arial"  
SIZE 6  
POSITION cc  
ANGLE auto  
MINDISTANCE 150  
COLOR 0 64 200  
END  
END  
CLASS #callejones, escalinatas  
MAXSCALEDENOM 25000  
STYLE  
COLOR 175 175 175  
SYMBOL "cartoline"  
SIZE 5
```

MINSIZE 1

END

STYLE

COLOR 200 200 200

SYMBOL "cartoline"

SIZE 3

END

LABEL

TYPE TRUETYPE

FONT "arial"

SIZE 4

POSITION cc

ANGLE auto

MINDISTANCE 150

COLOR 0 64 200

END

END

END

#----- Capa transporte: line-----#

LAYER

NAME "transporte"

STATUS on

TYPE line

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

```
DATA "the_geom FROM (select gid, the_geom from transporte where  
idlineabus=0) as g using unique gid using SRID=-1"
```

```
CLASS
```

```
NAME "TRANSPORTE"
```

```
STYLE
```

```
COLOR 200 100 0
```

```
SYMBOL "circulo"
```

```
SIZE 3
```

```
END
```

```
END
```

```
END
```

```
#----- Capa de ruta más corta: line -----#
```

```
LAYER
```

```
NAME "ruta"
```

```
STATUS ON
```

```
TYPE LINE
```

```
CONNECTION "user=x password=x dbname=miq host=localhost port=5432"
```

```
CONNECTIONTYPE postgis
```

```
CLASS
```

```
NAME "ruta"
```

```
STYLE
```

```
COLOR 250 0 0
```

```
WIDTH 2
```

```
END
```

```
END
```

END

#----- SITIOS -----#

#---- Capa varios----#

LAYER

NAME "varios"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='14'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "VARIOS"

STYLE

SYMBOL "var"

SIZE 15

COLOR 128 0 0

END

END

END

#-- Capa seguridad: policía, militares, UPCs, defensa civil, bomberos, cruz roja--#

LAYER

NAME "seguridad"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='13'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "EDUCATIVOS"

STYLE

SYMBOL "cuadrolet"

SIZE 16

COLOR 255 255 255

END

STYLE

SYMBOL "seguridad"

SIZE 16

COLOR 0 0 200

END

END

END

#----- Capa salud-----#

LAYER

NAME "salud"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria=12"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "SALUD"

STYLE

SYMBOL "cuadrolet"

SIZE 12

COLOR 255 0 0

END

STYLE

SYMBOL "salud"

SIZE 15

COLOR 255 255 255#120 180 255#0 128 255

END

END

END

#----- Capa restaurantes -----#

LAYER

NAME "restaurantes"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='11'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "RESTAURANTES"

STYLE

SYMBOL "cuadrolet"

SIZE 16

COLOR 255 210 255

END

STYLE

SYMBOL "restaurant"

SIZE 18

COLOR 200 0 200

END

END

END

#----- Capa religiosos-----#

LAYER

NAME "religiosos"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='10'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "RELIGIOSOS"

STYLE

SYMBOL "religioso"

SIZE 18

COLOR 128 0 255

END

END

END

#----- Capa parques-----#

LAYER

NAME "parques"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='15'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "PARQUES"

STYLE

SYMBOL "cuadrolet"

SIZE 16

COLOR 190 230 175

END

STYLE

SYMBOL "parque"

SIZE 20

COLOR 40 100 40

END

END

END

#----- Capa medios de comunicación-----#

LAYER

NAME "medicomu"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='9'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "MEDIOS DE COMU."

STYLE

SYMBOL "cuadrolet"

SIZE 12

COLOR 50 50 50

END

STYLE

SYMBOL "medicomu"

SIZE 15

COLOR 255 255 0

END

END

END

#----- Capa gasolineras-----#

LAYER

NAME "gasolineras"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='8'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "GASOLINERAS"

STYLE

SYMBOL "gasolinera"

SIZE 20

COLOR 0 0 0

END

END

END

#----- Capa farmacias-----#

LAYER

NAME "farmacias"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='7'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "FARMACIAS"

STYLE

SYMBOL "cuadrolet"

SIZE 12

COLOR 0 0 255

END

STYLE

SYMBOL "salud"

SIZE 15

COLOR 255 255 255

END

END

END

#----- Capa educativos-----#

LAYER

NAME "educativos"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='6'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "EDUCATIVOS"

STYLE

SYMBOL "educativo"

SIZE 18

COLOR 0 150 200

END

END

END

#----- Capa diversión-----#

LAYER

NAME "diversion"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='5'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "DIVERSION"

STYLE

SYMBOL "cuadrolet"

SIZE 13

COLOR 255 255 0

END

STYLE

SYMBOL "diversion"

SIZE 15

```
COLOR 200 100 0
END
END
END
#----- Capa deportivos-----#
LAYER
NAME "deportivos"
STATUS on
TYPE point
CONNECTIONTYPE postgis
CONNECTION "user=x password=x dbname=miq host=localhost port=5432"
DATA "the_geom FROM sitios"
FILTER "idcategoria='4'"
TOLERANCE 2
TOLERANCEUNITS pixels
TEMPLATE "t.html"
MAXSCALEDENOM 25000
CLASS
NAME "DEPORTIVOS"
STYLE
SYMBOL "cuadrolet"
SIZE 20
COLOR 120 240 120
END
STYLE
```

```
SYMBOL "deportivo"  
SIZE 20  
COLOR 70 150 70  
END  
END  
END  
#----- Capa correo-----#  
LAYER  
NAME "correo"  
STATUS on  
TYPE point  
CONNECTIONTYPE postgis  
CONNECTION "user=x password=x dbname=miq host=localhost port=5432"  
DATA "the_geom FROM sitios"  
FILTER "idcategoria='3'"  
TOLERANCE 2  
TOLERANCEUNITS pixels  
TEMPLATE "t.html"  
MAXSCALEDENOM 25000  
CLASS  
NAME "CORREO"  
STYLE  
SYMBOL "cuadrolet"  
SIZE 15  
COLOR 190 190 255
```

END

STYLE

SYMBOL "correo"

SIZE 15

COLOR 64 0 128

END

END

END

#----- Capa bancos: puntos-----#

LAYER

NAME "bancos"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='2'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "BANCOS"

STYLE

SYMBOL "cuadrolet"

SIZE 15

COLOR 220 255 220

END

STYLE

SYMBOL "banco"

SIZE 15

COLOR 0 100 0

END

END

END

#----- Capa alojamiento-----#

LAYER

NAME "alojamiento"

STATUS on

TYPE point

CONNECTIONTYPE postgis

CONNECTION "user=x password=x dbname=miq host=localhost port=5432"

DATA "the_geom FROM sitios"

FILTER "idcategoria='1'"

TOLERANCE 2

TOLERANCEUNITS pixels

TEMPLATE "t.html"

MAXSCALEDENOM 25000

CLASS

NAME "ALOJAMIENTO"

STYLE

SYMBOL "cuadrolet"

SIZE 19

COLOR 0 0 0

END

STYLE

SYMBOL "hospedaje"

SIZE 20

COLOR 255 70 140

END

END

END

END #map

ANEXO C. CONTENIDO DEL CD

CD

Adicionales/	Software necesario para la instalación de la aplicación creada.
Documentos/	Tesis en formato digital (archivos PDF y DOC).
MIQ/	Código fuente del Mapa Interactivo de la ciudad de Quevedo.
UML/	Diagramas UML.
Contenido.doc	Contenido del CD.

