



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

**UNIDAD ZACATENCO**  
**DEPARTAMENTO DE COMPUTACIÓN**

Tesis que presenta

**Eliot Yañez Ortega**

para obtener el Grado de

**Maestro en Ciencias en Computación**

**Director de la Tesis: Dr. Sergio Víctor Chapa Vergara**

Ciudad de México

Diciembre, 2016



AMBIENTE DE VISUALIZACIÓN DE UN  
SISTEMA DE INFORMACIÓN GEOGRÁFICA  
WEB



# Agradecimientos

Me gustaría agradecer a un sinnúmero de personas, que he conocido no solo en este periodo de maestría, sino a lo largo de mi vida , pero si lo hiciera no me alcanzaría este pequeño espacio, pero siempre los tengo presente. Agradezco a Dios, por cada momento de mi vida, cada momento difícil o feliz y todo lo que he recibido.

Agradezco a CONACYT por haberme brindado una beca escolar, con la cual pude realizar mis estudios en la Ciudad de México, esta beca fue una gran ayuda para poder pagar renta, comidas, ropa y material para apoyar mi maestría. Al CINVESTAV quiero agradecerle sus servicios como el comedor, el laboratorio de idiomas con el cual pude mejorar mis conocimientos en Inglés, y su biblioteca para consultar material académico valioso e indispensable.

A el Dr. Sergio Víctor Chapa Vergara por permitirme trabajar con el, por su valiosa asesoría en este proyecto de tesis, por sus consejos que fueron de gran utilidad. Sin olvidarme del MC. Andrés Bernal Jiménez por apoyarme con grandes consejos y sugerencias para resolver los problemas que se tuvieron a lo largo de la tesis.

Gracias a mis sinodales el Dr. Lorenzo Leija Salas y la Dra. Ana María Martínez Enríquez por tener la paciencia de revisar la tesis y ofrecerme observaciones oportunas.

Gracias a Sofía Reza Cruz por siempre ayudarme en los trámites que eran necesarios, por estar al pendiente de mi y mis compañeros. Agradezco también a las secretarias Felipa y Erika por su ayuda en todos los procesos administrativos. Agradezco al Dr. Amilcar, por sus consejos y por el apoyo administrativo.

A mis compañeros de maestría Jonathan, Daniel, José, Gil, Benjamín, Juan Luis, Daniel Torres, Pedro, Belem, Erik, Luis, Guillermo Falcon, Guillermo Monroy, Omar y Manuel, les agradezco por ser grandes compañeros.

Por último agradezco a mi familia, a mis Padres Fabian Yañez Huerta y Olivia Ortega Soto, por siempre estar ahí en los tiempos difíciles y en los tiempos felices , ellos hicieron posible que nunca careciera de comida o de educación, por ayudarme con sabios consejos y por darme la vida. A mis hermanos Fabian y Fabiola, por apoyarme, por siempre darme ánimos y por ser buenos hermanos. A mi tía Delia y a mi abuela Margarita quiero agradecerles, por su compañía cada fin de semana y por su gran apoyo.



# Resumen

Los Sistemas de Información Geográfica (SIG) son importantes en la computación, tienen una tecnología compleja con varias vertientes que se traducen en diversos proyectos de tecnología de software. Estos sistemas manejan inmensas cantidades de datos de diferentes tipos (comerciales, geográficos, militares, demográficos, entre otros). Los SIG han ido incorporando nuevas tecnologías dando el nacimiento de nuevas subáreas, entre estas los SIG-Web. Los SIG-Web son uno de los tantos proyectos de tecnologías Web, que unen la Web con los sistemas de información geográfica.

El objetivo de esta tesis es el desarrollo de un ambiente de visualización SIG-Web, el cual contiene cartografía de la República Mexicana. La cartografía fue obtenida por medio de cartas topográficas del Instituto Nacional de Estadística Geografía e Informática (INEGI) con escala de 1: 50, 000. En el caso de los SIG-Web la visualización de mapas y atributos es un problema relevante que requiere de métodos y técnicas, para el despliegue de datos vectoriales. Los SIG-Web utilizan diferentes tecnologías web, como los servicios web, los servicios de este ambiente están enfocados en el envío de datos espaciales.

El ambiente de visualización creado por medio de un SIG-Web consiste en tres capas lógicas: presentación, negocios y datos. Cada capa del sistema trabaja en conjunto para desplegar capas vectoriales. La capa de presentación por medio de un navegador que permite seleccionar zonas o estados, controlar, añadir y remover capas vectoriales. La capa de negocios comunica las capas de presentación y datos por medio de consultas a la capa de datos basándose en parámetros de la capa de presentación, las cuales generan dinámicamente archivos GeoJSON que son interpretados por OpenLayers para permitir el despliegue de nuevas capas vectoriales en una página Web. La capa de datos incorpora una base de datos espaciales que contiene todas las cartas topográficas para que sean consultados los datos vectoriales por medio de la capa de negocios. El ambiente de visualización SIG-Web será de vital importancia para el estudio de zonas geográficas o estados, estudios de planeación de infraestructura y toma de decisiones con base a la visualización de mapas. Por último el ambiente puede ser fácilmente adaptado a problemas específicos e incorporarle nuevas funcionalidades.

**Palabras Clave:** Base de datos espaciales: Datos vectoriales, Sistemas de Información Geográfica: SIG-Web, Tecnologías Web, servicios Web y Visualización.





# Abstract

Geographic Information Systems (GIS) are important in computing, have a complex technology with several slopes that translate into various software technology projects. These systems handle immense amounts of data of different types (commercial, geographic, military, demographic, among others). GIS have been incorporating new technologies giving rise to new sub-areas, among them the Web GIS. Web GIS are one of the many Web technology projects that link Web with geographic information systems.

The objective of this thesis is the development of a Web GIS visualization environment, which contains cartography of Mexican Republic. The cartography was obtained by means of topographical charts from Instituto Nacional de Estadística Geografía e Informática (INEGI) with scale of 1: 50, 000. In the case of GIS Web visualization of maps and attributes is a relevant problem that requires methods and techniques for the deployment of vector data. GIS Web use different web technologies, such as web services, the services of this environment are focused on the sending of spatial data.

The visualization environment created through a SIG-Web consists of three logical layers: presentation, business and data. Each layer of the system works together to deploy vector layers. The presentation layer by means of a browser that allows to select zones or states, to control, to add and to remove vectorial layers. The business layer communicates presentation layers and data by querying the data layer based on presentation layer parameters, which dynamically generate GeoJSON files that are interpreted by OpenLayers to allow the deployment of new vector layers in a Web page. The data layer incorporates a spatial database that contains all topographic maps so that the vector data can be queried through the business layer.

The GIS Web visualization environment will be of vital importance for the study of geographic areas or states, infrastructure planning studies and decision making based on the visualization of maps. Finally the environment can be easily adapted to specific problems and incorporate new functionalities.

*Keywords:* Spatial database: Vector data, Geographic Information Systems: GIS Web, Web Technologies, Web services and Visualization.



# Índice general

Agradecimientos	I
Resumen	III
Abstract	V
Lista de figuras	XI
Lista de tablas	XV
Lista de códigos	XVII
Lista de abreviaturas	XIX
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Planteamiento del problema . . . . .	3
1.3. Motivación . . . . .	4
1.4. Objetivos del proyecto . . . . .	5
1.5. Organización de la tesis . . . . .	5
<b>2. Marco teórico y tecnológico de los SIG</b>	<b>7</b>
2.1. Sistemas de Información Geográfica (SIG) . . . . .	8
2.2. Conceptualización del mundo real en los SIG . . . . .	11
2.3. Modelos de datos . . . . .	19
2.3.1. Modelo de datos Vectorial . . . . .	19
2.3.2. Modelo de datos Raster . . . . .	21
2.3.3. Estándar de tipos de datos y funciones geográficas . . . . .	21
2.4. Bases de datos espaciales . . . . .	24
2.4.1. Sistemas de Gestión de bases de datos (DBMS) . . . . .	24
2.4.2. Sistema de gestión de bases de datos relacionales(RDBMS) . . . . .	25
2.4.3. Bases de datos espaciales . . . . .	25

2.5.	Arquitectura de los SIG . . . . .	28
2.6.	Proyección de mapas . . . . .	30
2.6.1.	Sistemas de coordenadas geográficas . . . . .	32
2.6.2.	Escala . . . . .	32
2.6.3.	Datum . . . . .	33
2.6.4.	El sistema de coordenadas UTM . . . . .	34
2.7.	SIG-Web . . . . .	37
2.7.1.	Web 2.0 . . . . .	38
2.7.2.	Fundamentos Web . . . . .	38
2.7.3.	Sistemas de Información Geográfica Web . . . . .	40
2.8.	Arquitectura y componentes básicos de un SIG Web . . . . .	47
2.8.1.	Arquitectura <i>Thin Client</i> . . . . .	49
2.8.2.	Arquitectura <i>Thick Client</i> . . . . .	49
2.9.	Resumen . . . . .	49
<b>3.</b>	<b>Estado del arte</b>	<b>51</b>
3.1.	MapBeing: Una arquitectura para la manipulación y publicación de datos vectoriales en un SIG basado en Web . . . . .	51
3.2.	Diseño e Implementación de un Modelo, basado en Web, SIG de un atlas de Cáncer . . . . .	54
3.3.	Integración del Enfoque AJAX de Servicios Web en SIGs de visualización	55
3.4.	Desarrollo de una aplicación SIG Web para asignación espacial en un inventario de emisiones basado en herramientas <i>opensource</i> . . . . .	57
3.5.	Diseño e Implementación de una plataforma SIG Web Basada en un <i>Middleware</i> SIG de libre distribución . . . . .	58
3.6.	Trabajos SIG, SIG Web y de visualización . . . . .	61
3.7.	Resumen . . . . .	65
<b>4.</b>	<b>Diseño del SIG Web</b>	<b>67</b>
4.1.	Diseño por capas lógicas . . . . .	67
4.1.1.	Capa de Presentación . . . . .	69
4.1.2.	Diseño de la interfaz . . . . .	74
4.1.3.	Capa de Negocios . . . . .	75
4.1.4.	Capa de datos . . . . .	77
4.1.5.	Interacción entre capas . . . . .	81
4.2.	Casos de Uso del SIG Web . . . . .	83
4.3.	Consideraciones de Diseño . . . . .	85
4.4.	Resumen . . . . .	87

<b>5. Implementación</b>	<b>89</b>
5.1. Requerimientos . . . . .	89
5.2. Implementación de la capa de datos . . . . .	90
5.2.1. Implementación de la base de datos . . . . .	90
5.2.2. Zonas UTM y geográficas . . . . .	93
5.3. Implementación de la capa de negocios . . . . .	95
5.3.1. Elección de zonas geográficas . . . . .	95
5.3.2. Elección de Estados de la República Mexicana . . . . .	95
5.3.3. Generación de archivos GeoJSON . . . . .	98
5.4. Implementación de la capa de presentación . . . . .	101
5.4.1. Implementación de la interfaz . . . . .	102
5.5. Resumen . . . . .	110
<b>6. Validación y pruebas al SIG Web</b>	<b>113</b>
6.1. Primera Validación . . . . .	113
6.2. Segunda Validación, Caso de Estudio: Estado de México . . . . .	115
6.2.1. Validación de la función de categorización . . . . .	117
6.3. Tercera Validación . . . . .	119
6.4. Análisis de resultados . . . . .	120
6.5. Resumen . . . . .	122
<b>7. Conclusiones y trabajo a futuro</b>	<b>123</b>
7.1. Conclusiones . . . . .	123
7.2. Trabajo a futuro . . . . .	124
<b>A. OpenLayers 3.0</b>	<b>129</b>
<b>B. GeoJSON</b>	<b>133</b>
<b>Referencias</b>	<b>136</b>



# Índice de tablas

2.1. Ventajas y Desventajas del formato Vectorial . . . . .	21
2.2. Ventajas y Desventajas del formato Raster . . . . .	22
4.1. Tablas obtenidas por medio de cartas topográficas. . . . .	81
4.2. Caso de Uso: Agregar Capa. . . . .	83
4.3. Caso de Uso: Generar y desplegar Capa. . . . .	84
4.4. Caso de Uso: Clasificar por medio de datos nominales. . . . .	84
4.5. Caso de Uso: Mostrar o Ocultar Capa. . . . .	85
4.6. Caso de Uso: Mostrar información de los objetos geográficos. . . . .	85
5.1. Proyecciones WGS 84/UTM para México . . . . .	93
5.2. Ejemplo de claves de las 2313 Zonas Geográficas . . . . .	94
6.1. Comparativa entre las diferentes arquitecturas estudiadas. . . . .	122





# Índice de figuras

2.1. Esquema de un GIS (por sus siglas en Inglés como sistema automatizado de información geográfica y datos del LANDSAT, de acuerdo con Ellen M. Knapp y Deborah Rider [1]. . . . .	8
2.2. Cubo multidimensional del espacio, tiempo y atributos, donde cada tema apunta a un modelo de base de datos (De acuerdo con [2]). . . . .	12
2.3. Capas de datos geográficos de acuerdo con [3]. . . . .	13
2.4. Pasos en el desarrollo de un SIG (adaptado de Hendrick & Ottens en 1997 [4]) . . . . .	15
2.5. Conceptualización del mundo real en un SIG (adaptado de Hendrick & Ottens 1997 [4]) . . . . .	16
2.6. Etapas del modelado de datos (de acuerdo con [5] . . . . .	17
2.7. Diferentes conceptualizaciones espaciales de un proceso (de acuerdo con Raper (1999) [6] . . . . .	18
2.8. Dimensiones de una entidad representada por un objeto (de acuerdo con Behr (1998) [7] ) . . . . .	19
2.9. Modelo Capa-SIG (de acuerdo con Laurini & Thompson (1992) [8], [5]) . .	19
2.10. Modelo ER (de acuerdo con [5]) . . . . .	20
2.11. Formato raster y vectorial . . . . .	20
2.12. Modelo de los tipos de datos geográficos (de acuerdo a la OGC [9]) . . . .	23
2.13. Funciones SQL Espacial: ST_Intersects, ST_Contains y ST_Difference . . .	27
2.14. Componentes de un SIG. . . . .	29
2.15. Ejemplo de una proyección cartográfica. . . . .	31
2.16. Notaciones de las escalas . . . . .	33
2.17. Zonas UTM de Estados Unidos de América [10]. . . . .	35
2.18. Arquitectura básica de una aplicación Web incluye tres capas. . . . .	39
2.19. Modelo Típico de un SIG Web . . . . .	41
2.20. Servicios Web basados en SOAP. . . . .	45
2.21. Servicios Web basados en REST. . . . .	45
2.22. Modelo Típico de un SIG Web . . . . .	47
2.23. Aplicaciones del lado del Servidor . . . . .	49
2.24. Aplicaciones del lado del Cliente . . . . .	50

3.1.	Arquitectura de MapBeing . . . . .	53
3.2.	Arquitectura del sistema PA Cancer Atlas . . . . .	55
3.3.	Integración de Google Maps con OGC WMS . . . . .	56
3.4.	Arquitectura del SIG Web para el inventario de emisiones . . . . .	57
3.5.	Arquitectura del SIG Web basado en software de libre distribución . . . . .	60
3.6.	El Concepto LIVEWall propuesto por Sakuraba et al. [11] . . . . .	63
3.7.	Prueba PD-GRASS en un sistema local . . . . .	63
3.8.	Uso de la Pared de Vídeo con WallGlobe . . . . .	64
3.9.	Visualización del terreno . . . . .	65
4.1.	Arquitectura de tres capas para aplicaciones Web (de acuerdo a [12]). . . . .	68
4.2.	Comparación del modelo clásico y Ajax. . . . .	73
4.3.	Diseño de la capa de presentación con sus componentes. . . . .	74
4.4.	Diseño de la interfaz del SIG-Web. . . . .	75
4.5.	Componentes de la Capa de Negocios. . . . .	78
4.6.	Ejemplo de Carta Topográfica, Zona La Guajolota Clave:H13C33. . . . .	80
4.7.	Arquitectura de tres capas del SIG. . . . .	82
4.8.	Diagrama de actividades por capa. . . . .	82
4.9.	Caso de uso del ambiente de visualización. . . . .	83
5.1.	Las 2313 zonas Geográficas de los Estados Unidos Mexicanos. . . . .	91
5.2.	Las 6 Zonas UTM de los Estados Unidos Mexicanos. . . . .	92
5.3.	BD MapasMexico. . . . .	96
5.4.	Pasos para generar una capa de forma dinámica. . . . .	98
5.5.	Archivo GeoJSON generado por aeropuerto50_a_utm . . . . .	101
5.6.	OpenStreetMap en funcionamiento. . . . .	104
5.7.	Interfaz de selección de estados. . . . .	107
5.8.	Interfaz de selección de zonas geográficas. . . . .	108
5.9.	Arquitectura Final del ambiente de visualización. . . . .	111
6.1.	Despliegue de la capa camino50_l_utm en cuatro Estados. (a) Veracruz , (b) Zacatecas, (c) Puebla y (d) Colima . . . . .	114
6.2.	Pruebas de visualización a tres diferentes zona geográficas. (a)E14B49 Ve- racruz, (a)E14B43 Puebla, (c)G14C26 Monterrey. . . . .	115
6.3.	Visualización del Estado de México con 11 capas vectoriales. a) Sin la capa localidad50_p_utm, b) Con la capa localidad50_p_utm. . . . .	116
6.4.	Visualización de una Zona del Estado de México con 10 capas vectoriales. . . . .	117
6.5.	Categorización de la capa carretera50_l_utm a su campo “jurisdicci” (juris- dicción). . . . .	118
6.6.	Categorización de la capa camino50_l_utm a su campo tipo. . . . .	118
6.7.	Categorización de la capa localidad50_p_utm a su campo categoría. . . . .	119
6.8.	Capa de municipios obtenida de un archivo KML. . . . .	120

6.9. Capa de climas obtenida de un archivo GeoJSON. . . . .	120
A.1. Diagrama UML de OpenLayers 3. . . . .	129



# Lista de códigos

5.1.	Consulta SQL para crear la tabla clave_entidades_zona. . . . .	94
5.2.	Selección por medio de claves. . . . .	95
5.3.	Selección por medio de Estado. . . . .	97
5.4.	Consulta para generar GeoJSON. . . . .	99
5.5.	Conexion con BD MapasMexico. . . . .	100
5.6.	Archivo PHP aeropuerto50_a_utm.php. . . . .	100
5.7.	Creación de un mapa con OpenLayers 3.0. . . . .	101
5.8.	Dos Mapas HTML. . . . .	102
5.9.	Creacion de “map” en OpenLayers. . . . .	103
5.10.	Creación del menú de control de capas. . . . .	105
5.11.	Creación del menú de seleccionó de capas. . . . .	105
5.12.	Formulario de Categorización de capas. . . . .	106
5.13.	Categorización de capas. . . . .	108
5.14.	Función getDataCamino para generar consultas en la interfaz. . . . .	109
A.1.	Creación de elemento DOM para asignar un mapa . . . . .	131
A.2.	Declaración de un objeto del tipo ol.View . . . . .	131
A.3.	Creación de un objeto ol.source . . . . .	132
A.4.	Declaración de una nueva capa . . . . .	132
A.5.	Mapa de OpenLayers 3 . . . . .	132
B.1.	Ejemplo de un Point en GeoJSON . . . . .	134
B.2.	Ejemplo de un MultiPoint . . . . .	134
B.3.	Ejemplo de un LineString . . . . .	134
B.4.	Ejemplo de un MultiLineString . . . . .	134
B.5.	Ejemplo de un Polygon . . . . .	135
B.6.	Ejemplo de un MultiPolygon . . . . .	135
B.7.	Ejemplo de un GeometryCollection . . . . .	135



# Lista de Abreviaturas

- AJAX - Asynchronous JavaScript And XML.
- API - Application Programming Interface.
- BD- Base de Datos
- BDE- Base de Datos Espacial
- CGIS - Canada Geographic Information System.
- CONABIO - Comisión nacional para el conocimiento y uso de la biodiversidad.
- CSS - Cascading Style Sheet - Hojas de estilo en cascada.
- C/S - Cliente - Servidor.
- CRS - Coordinate Reference Systems.
- DBMS - Database Management System - Sistema de gestión de bases de datos
- DVO - Distributed Visual Objects - Objetos Distribuidos Visuales.
- EPSG - European Petroleum Survey Group
- ESRI - Environmental Systems Research Institute.
- GCS - geographic coordinate system
- GIS - Geographic Information System.
- GML - Geography Markup Language.
- GeoJSON - Geographic JSON.
- GPS - Global Positioning System - Sistema de Posicionamiento Global. ion.

- HTML - Hypertext Markup Language.
- HTTP - Hypertext Transfer Protocol.
- INEGI - Instituto Nacional de Estadística y Geografía.
- ISO - International Standards Organizat
- JDBC - Java Database Connectivity.
- JSON - JavaScript Object Notation.
- JSP - JavaServer Pages.
- KML - Keyhole Markup Language.
- PEMEX - Petróleos Mexicanos.
- PHP - PHP Pre Hypertext -processor.
- OGC - Open Geospatial Consortium.
- OWS - OGC Web Services.
- RDBMS - Relational database management system - Sistema de gestión de bases de datos relacionales.
- SIG - Sistemas de Información Geográfica.
- SIG Web - Sistemas de Información Geográfica Web.
- SIGINM - Sistema de Información Geográfica para el Instituto Nacional de las Mujeres.
- SOA - Service-Oriented Architecture.
- SOAP - Simple Object Access Protocol.
- SQL - Structured Query Language.
- SRID - Spatial Reference System Identifier.
- SVG - Scalable Vector Graphics.
- URI - Uniform Resource Identifier.
- URL - Uniform Resource Locator.
- UTM - Universal Transverse Mercator.



- W3C - World Wide Web Consortium.
- WCS - Web Coverage Service.
- WFS - Web Feature Service.
- WFS-T - Transactional WFS.
- WGS 84 - World Geodetic System 1984.
- WMS - Web Map Service.
- WWW - World Wide Web.
- XHTML - Extensible HTML.
- XML - Extensible Markup Language.



# Capítulo 1

## Introducción

El presente capítulo hace una breve descripción del proyecto de tesis. Se hace una breve reseña de los antecedentes del proyecto y de los trabajos previos desarrollados. Se describe el problema a solucionar y se menciona la motivación del proyecto. El capítulo también describe objetivo general y los objetivos particulares, por último se describe la estructura de la tesis.

### 1.1. Antecedentes

Desde el año 1991, en la sección de Computación del Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, en el Laboratorio de Computación Científica y Base de Datos se han llevado a cabo diferentes proyectos, tesis de investigación, acerca de : lenguajes visuales, bases de datos geográficas, sistemas de información geográfica y visualización científica. Entre los antecedentes importantes tenemos:

- El proyecto de Sistema de Información Geográfica para planes de inversión en exploración petrolera, PEMEX Exploración y Explotación (1993-1995) [13].
- Base de Datos de la Colección Nacional de Microorganismos CDBB-500 [13].
- SIGINM: Sistema de Información Geográfica para el Instituto Nacional de las Mujeres [14].
- Diseño y desarrollo de una API para la visualización de datos espaciales en sistemas Web de información geográfica para dispositivos móviles” [15].
- Minería de datos visual sobre una pared de vídeo [16].
- Objetos Distribuidos Visuales [17].

El proyecto que se planteó para PEMEX dentro del área de exploración petrolera. Consistió en el diseño conceptual de la base de datos mediante la aplicación de la metodología DATAI, se desarrolló un gestor de base de datos geográfica. Se implementó un modelo relacional para los datos nominales, un modelo de datos de punto línea área para los datos geográficos. También se migró el sistema a estaciones de trabajo, bajo la plataforma Unix en el ambiente X-Windows/Motif, obteniendo una contribución en la creación de un nuevo lenguaje visual de consulta y manipulación de datos.

El Sistema Micro-500 [13] para administrar una base de datos de microorganismos denominada CDBB-500, es otro de los sistemas importantes desarrollados. La base de datos CDBBB-500, contiene la información de 3000 Cultivos Microbianos, diferentes zonas geográficas del territorio nacional y del mundo. El sistema fue instalado en un servidor Mac G4, con tecnología de software abierta: servidor de base de datos *PostgreSQL*, *Apache Tomcat*. Con esta versión se concluyó una importante etapa, en donde el Sistema Micro-500 permite consultas remotas en un ambiente de Internet y es la base para integrar bases de datos, donde es necesario considerar la integridad mediante un modelo activo.

El sistema SIGINM [14], es un SIG orientado a la *Web*, desarrollado a partir de *software open source* como *PostgreSQL* y *PostGIS* para el almacenamiento y manejo de datos. Se usó tecnologías como *JSP*, *Beans*, *Apache Tomcat* para ser un contenedor y un servidor *Web* con soporte de *servlets* y *JSPs*. Este SIG está enfocado al Instituto Nacional de las mujeres y su objetivo es ayudar a la toma de decisiones en base a la visualización de mapas con diferentes capas.

En el trabajo de tesis [15] se hace uso de la tecnología del dispositivo móvil iPod-Touch, de los avances en SIG y de las tecnologías para Web 2.0 para plantear el diseño y desarrollo de una API para visualizar objetos geográficos y crear SIG en Web en dispositivos móviles. Ofreciendo al desarrollador módulos para la generación de objetos geográficos para ser visualizados en el navegador en formato Canvas y SVG. Se ofrece también un módulo para realizar consultas a la base de datos espacial. La API está formada por una biblioteca de tags-JSP y archivos en JavaScript.

Otro antecedente fue la creación de un sistema [16], el cual realiza minería visual con el apoyo de una pared de vídeo, aplicando un enfoque matemático-computacional, enlazando una correlación de Pearson con el manejo de una pared de vídeo. El sistema se construyó con 6 Mac mini, 6 pantallas de 23 pulgadas, un servidor de vídeo Mac pro y un servidor de base de datos G5. El manejador de base de datos que se empleó es PostgreSQL. La aplicación fue desarrollada con C-objetivo, con las bibliotecas de libpq (para la comunicación con PostgreSQL) y el framework de OpenGL (para la generación de los gráficos).

El trabajo de Objetos Distribuidos Visuales [17], propuso el modelo DVO (*Distributed Visual Objects*) basado en capas. Este modelo se enfoca principalmente en la creación de un manejador de ventanas distribuido. El componente principal del modelo DVO es el objeto distribuido visual. EL modelo se compone de diferentes capas que son la interfaz visual, capa de contenedor, capa de control, capa de comunicación y capa de reconocimiento de eventos. Se propuso este modelo, con la finalidad de ser un manejador de ventanas distribuido. El modelo se basa principalmente en el paradigma orientado a objetos. También se implementó un visor de imágenes sobre la pared de vídeo CinvesWall en forma distribuida en cada nodo del clúster de visualización. También se realizó una aplicación para la pared de vídeo CinvesWall llamada Minero de Datos Visual, con la finalidad de obtener datos de diversas fuentes y generar correlación de Pearson [18, 16].

El trabajo de [19] se desarrolló un generador de código de sistemas Web con un caso de estudio en los SIG. El generador esta basado en descripciones XML (eXtensible Markup Language) para añadir características de visualización en mapas y manejo de datos geográficos con base en GML(Geography Markup Language). La aplicación Web se colocó en un servidor que permitió la conexión JDBC a una base de datos relacional, con un modelo de datos geográficos vectoriales de mapas del INEGI con escala 1:50,000 del estado de Colima. La visualización se lleva a cabo en el navegador del cliente en donde se tiene un soporte para XHTML (eXtensible Hypertext Markup Language).

En otro trabajo [20], se creó una herramienta de modelado de datos para diseñar sistemas e integrar las fases de desarrollo de los mismos. El caso de estudio fue el diseño de un modelo de datos para un SIG. La herramienta permite crear tres tipos de modelos (base de datos, interfaz de usuario, comportamiento).

En los trabajos anteriormente mencionados, se observa la utilización de diferentes tecnologías que son de interés de esta tesis como son Bases de Datos Geográficas o Espaciales, Sistemas de Información y Visualización.

## 1.2. Planteamiento del problema

Los sistemas de información geográfica manejan inmensas cantidades de datos nominales y geográficos. Los Sistemas de información Geográfica basados en la Web deben tomar en cuenta la cantidad de información a visualizar para no afectar el desempeño del sistema. En las aplicaciones Web es esencial el manejo de los recursos y no cargarlas con demasiada información, pero en la visualización de datos vectoriales, éstos suelen ser de gran tamaño. Por esto es la creación de una arquitectura apropiada para el SIG, que permita la visualización de capas vectoriales sin que el sistema pierda desempeño. Los ambientes de visualización de datos espaciales o geográficos no utilizan solamente una capa con pocas

características, esto depende del estudio a realizar y del área al de interés, se requiere de diversas capas, la cuales cuentan con una gran cantidad de polígonos e información.

El ambiente de visualización de un SIG-Web debe visualizar diferentes tipos de cartas topográficas del Instituto Nacional de Estadística y Geografía(INEGI) con escala 1: 50, 000, que están georeferenciadas de acuerdo a una zona UTM, las cuales contienen toda la cartografía de la República Mexicana, por esta razón es necesario que el SIG visualice cartas de cualquier zona geográfica, transformando el tipo de coordenadas que utilizan estas cartas de acuerdo a su ubicación. El SIG-Web debe permitir la visualización por medio de la selección de estados o zonas de la república Mexicana, con el fin de prevenir el uso de grandes cantidades de datos y no afectar el desempeño de la aplicación.

### 1.3. Motivación

Desde hace varios años los sistemas de información manejan inmensas cantidades de datos de diferentes tipos (comerciales, geográficos, militares, demográficos, entre otros). La evolución de los sistemas de información crearon una subárea, llamada sistemas de información geográfica (SIG), la cual específicamente tratan información espacial como son ubicación, lugares, coordenadas, polígonos, imágenes georeferenciadas entre otras. Los SIG desde su creación en el año de 1962 han llegado a ser parte esencial en muchas áreas de estudio, debido a que éstos integran, analizan y visualizan información geográfica. Pero estos sistemas tenían el límite de solo ser accesibles a usuarios especializados de grupos. Con el crecimiento de las tecnologías Web, éstas se combinaron con los SIG, surgiendo los SIG-Web.

SIG-Web son más flexibles, proporcionan ventajas como son un menor costo de desarrollo, por su naturaleza son sistemas distribuidos que son accedidos por usuarios de y desde cualquier parte del mundo. SIG-Web no necesitan de instalación en un equipo de computo, solo es requerido un navegador Web compatible y la URL del sistema para poder accederlo sin importar el sistema operativo que el usuario éste usando.

Gran parte de la investigación y el desarrollo de los SIG se ha centrado en la construcción de métodos de servicios *Web* geográficos y herramientas para proporcionar a los usuarios por medio de la Web, la utilización de diversos tipos de datos geográficos y ayudar en la toma de decisiones. La presente tesis tiene la principal motivación de apoyar la visualización por medio de un SIG de diferentes áreas o zonas geográficas del país, ecosistemas, hidrología, red de caminos, red ferroviaria, entre otras.

## 1.4. Objetivos del proyecto

### Objetivo general

Diseñar e implementar un ambiente de visualización por medio de un sistema de información geográfica basado en tecnología Web, el cual debe aprovechar los diferentes beneficios de estos sistemas, con la finalidad de desplegar diferentes objetos geográficos y brindar la visualización de información geográfica.

### Objetivos particulares

1. Permitir que el SIG Web despliegue diferentes cartas geográficas de los Estados Unidos Mexicanos, para la creación de capas vectoriales.
2. Visualizar las cartas topográficas del INEGI a partir de la selección de áreas o estados, para crear capas vectoriales de un lugar específico.
3. Crear módulo de consulta de datos geográficos, que seleccione y consulte datos nominales de forma visual, con el fin de observar información específica en el mapa.
4. Realizar una función de categorización de capas vectoriales a partir de sus datos nominales, para identificar las diferencias de cada elemento geográfica.
5. Desarrollar dos tipos de interfaz, para permitir la selección de estados o zonas geográficas.

## 1.5. Organización de la tesis

La tesis esta estructurada con los siguientes capítulos:

- **Capítulo 1 Introducción.** Describe el planteamiento del problema, antecedentes, los objetivos de esta investigación y el esquema de tesis
- **Capítulo 2 Marco teórico y tecnológico de los SIG.** Contiene la definición de SIG, elementos de los SIG.
- **Capítulo 3 Estado del arte.** Se describe los trabajos relacionados a esta tesis y se complementa el marco teórico.
- **Capítulo 4 Diseño del SIG.** Se hablará sobre la arquitectura usada en el SIG, el diseño de la base de datos, las herramientas y tecnologías utilizadas.
- **Capítulo 5 Implementación del SIG.** Descripción de como se implementó cada una de las capas del SIG-Web.

- **Capítulo 6 Validación y Pruebas.** Describe diferentes tipos de pruebas al SIG con diferentes capas vectoriales para validar su funcionamiento.
- **Capítulo 7 Conclusiones y trabajo a futuro.** Se hace una conclusión con respecto a la validación del ambiente de visualización de un SIG-Web y por ultimo se incluye las actividades a futuro del proyecto de tesis.



## Capítulo 2

# Marco teórico y tecnológico de los SIG

Este capítulo contiene la definición de SIG, su historia y cuales son sus componentes esenciales. Se explica el marco teórico que se enfoca en definiciones como proyección, escala, sistemas de coordenadas, y en la parte de marco técnico se detalla las tecnologías Web necesarias para entender los SIG Web, la importancia de estos sistemas y como están conformados.

Los *Sistemas de Información Geográficos* (SIG), que por sus siglas en inglés se denominan GIS, su historia inicia en la década de 1960 ([21] [Burrough y McDonnell, 1998]),<sup>1</sup> aproximadamente diez años después del surgimiento de las primeras computadoras, lo cual ha significado que en su desarrollo se tiene una fuerte interacción entre la teoría de la computación y su tecnología. Por ejemplo, en la década de los setentas del siglo veinte, el análisis espacial y los modelos de datos son representativos de esfuerzos en la geometría computacional [23] [Peucker y Chrisman 1975]; pero, por otro lado, las computadoras del tipo “mainframe” como: IBM 370, CDC 3100, HP 3000, CDC Cyber 74, PDP-10, entre otras<sup>2</sup>, propiciaron avances importantes para los SIG y, en ese momento relacionado a la tecnología satelital remota (ver Figura 2.1).

Actualmente, el Internet en los SIG tiene un gran valor en los denominados SIG-Web con alcances en la visualización y diseminación de los datos espaciales. En la perspectiva de las tecnologías que soportan y dan consejos prácticos sobre la publicación de proyectos SIG en el “World Wide Web.<sup>es</sup>: Harder C. (1998) *Serving Maps on the Internet: Geographic Information on the World Wide Web*, ESRI Press [24].

---

<sup>1</sup>Los aspectos históricos de los SIG son tratados en gran detalle por Foresman TW (ed) (1998) *The historical of Geographic Information Systems: perspectives from the pioneers* [22] [Foresman TW, 1998]

<sup>2</sup>El lector puede recurrir al artículo *Automated Geographic Information Systems an Landsat Data: A Survey* [1], Ellen M. Knapp y Deborah, donde se expone un cuadro sumario de los sistemas de computadoras y organizaciones en contra de los estándares de las categorías y características.

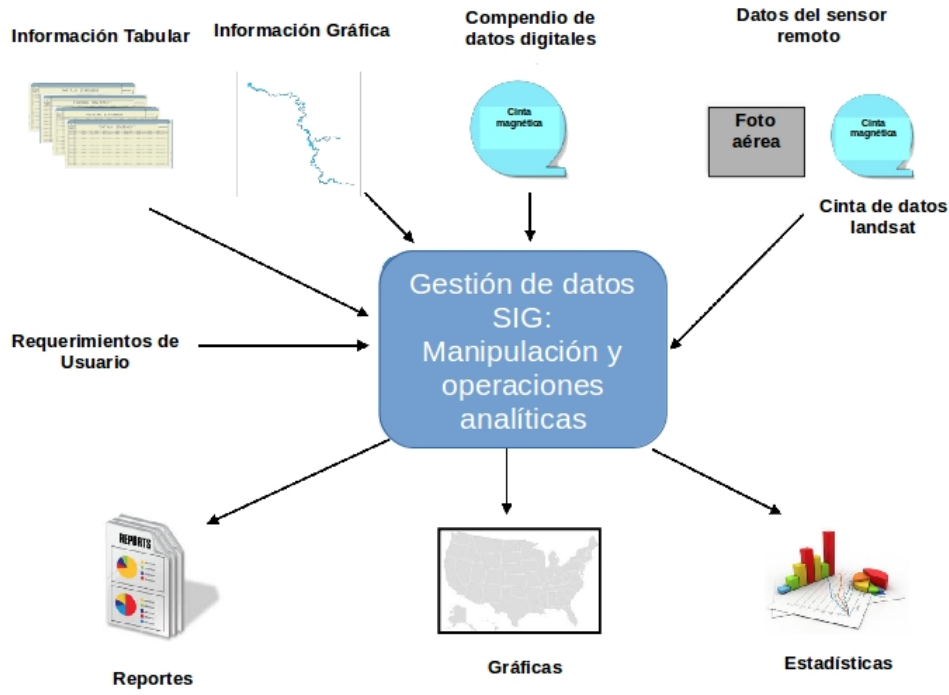


Figura 2.1: Esquema de un GIS (por sus siglas en Inglés como sistema automatizado de información geográfica y datos del LANDSAT, de acuerdo con Ellen M. Knapp y Deborah Rider [1].

## 2.1. Sistemas de Información Geográfica (SIG)

Todo lo que sucede en el tiempo, en algún lugar pasa, de tal forma que el conocer el “Qué” es importante en el “Donde” y en consecuencia explicar el porque sucede ahí. De esta manera lo que sucede en el mundo real, es ejecutado por una persona, empresa o el gobierno, depende de la localización en donde se encuentra y afecta al espacio circundante que lo define en su circunstancia. Los SIG son sistemas computacionales que toman el modelo del mundo real en una tecnología que manipula la información de entidades que se vinculan con el “donde” para explicar el estado geográfico, ayudar a la toma de decisiones y simular en los espacios geográficos cuando tomamos la componente del tiempo.

Para mostrar la importancia que tienen los SIG actualmente y el papel que desempeñan en la sociedad, basta mencionar que aproximadamente un 70% de la información que manejamos en nuestra vida cotidiana está georeferenciada. Por otro lado, en el reporte *Towards 2020 Science: A Draft Roadmap* ([Emmott2005] [25]), se señala que una de las metas para el 2020 es el surgimiento de una nueva ciencia, en donde sobresale el papel fundamental que tiene la *Ciencia de la Computación* en este contexto, los SIG representan

una de las áreas muy importantes de desarrollo.

Los SIG, en el sentido mas estricto son un tipo de sistemas de información que tienen la capacidad de: integrar, almacenar, editar, compartir y mostrar información geográficamente referenciada. Es decir, que se trata de información a la cual se le asigna una posición geográfica, información que viene acompañada de otra información relativa a su localización. De esta manera, los SIG los podemos caracterizar en tres aspectos: el concepto de bases de datos espaciales, como un conjunto de instrumentos y tecnologías basados en aplicaciones. Los SIG no tienen una única definición, existen otras definiciones en la literatura de acuerdo a BD, al conjunto de instrumentos y definen a un SIG como:

1. Un sistema de base de datos en el cual la mayor parte de los datos esta indexada espacialmente y sobre la cual un conjunto de procedimientos son operados con el fin de responder a las consultas sobre las entidades espaciales en la base de datos [26].
2. Cualquier conjunto de procedimientos manuales o basados en computadora usados para almacenar y manipular datos referenciados geográficamente [27].
3. Un poderoso conjunto de herramientas para recolectar, almacenar, recuperar a voluntad, transformar y mostrar datos espaciales del mundo real [21].
4. Un sistema para capturar, almacenar, revisar, manipular, analizar y mostrar datos, los cuales están referenciados espacialmente a la tierra.
5. Un sistema de toma de decisiones, que implica la integración de los datos referenciados espacialmente, en un entorno de resolución de problemas [28].
6. SIG es un sistema de hardware, software y procedimientos que captura, almacena, edita, manipula, maneja, analiza, comparte y muestra datos georeferenciados [29].

Por sus componentes los SIG se configuran en cuatro procesos principales: entrada almacenamiento, transformación y presentación. Es una base de datos digital de propósito especial en el que un sistema espacial de coordenadas común es el principal medio de referencia [30]. Los SIG requieren un medio de:

1. Entrada de datos: mapas, fotos aéreas, satélites, encuestas y otras fuentes.
2. Almacenamiento de datos: recuperación y consulta.
3. Transformación de datos: análisis y modelado, incluyendo estadística espacial.
4. Presentación de datos: mapas, informes y planes.

Algunos componentes de los SIG son tecnológicos, éstos incluyen recopiladores de datos de la era espacial, redes avanzadas de comunicación, computo sofisticado, etc. La clave de las definiciones de SIG es tecnología donde las consultas tienen en común el “donde” o tres dimensiones asociada con “que”. Los mapas han sido la base y el medio de estos en su ruta de desarrollo desde los años de 1960 se ha llegado a los SIG-Web con la actual tecnología de Internet y la Web [3].

Durante los 1960's y 1970's iniciaron nuevas tendencias en donde los datos acerca de los recursos naturales, del suelo y los sistemas de panorama empezaron a ser usados para: el gravamen de recursos, evaluación de propiedad de la tierra, la planeación y el uso del suelo. De esta manera la tecnología de la computación subraya la automatización de los SIG y sus principios y ciencia dieron pie a modelos espaciales geográficos de los SIG. El primer SIG operacional fue desarrollado en 1962 por Roger Tomlinson para el *Canada's Federal Department of Forestry and Rural Development*. Llamado *Canada Geographic Information System* (CGIS) [3], fue usado por el inventario de tierra y planificación geográfica de Canadá. El cual fue un primer paso en la cartografía computacional al incluir superposiciones de capas de mapa. Tomlinson ha llegado a ser conocido como el “padre” de los SIG por el desarrollo de CGIS y por promover este tipo de sistemas. De esta manera, a partir de los años de 1970's, los SIG con el desarrollo de las computadoras del tipo “mainframes” como: IBM 370, CDC 3100, HP 3000, CDC Cyber 74, PDP-10, entre otras<sup>3</sup>. siguieron una trayectoria de mayor automatización llevada por el crecimiento tecnológico de la computación.

Desde el enfoque de graficación por computadora y visualización los SIG producen mapas muy bonitos, en una gran diversidad de versiones usando diferentes escalas, temas y símbolos. Sin embargo, en los SIG los mapas son un medio donde el análisis espacial, éstos tienen funciones analíticas poderosas que convierten los datos en información muy útil. Éstos relacionan los datos dispersos sobre la base geográfica común, revelando relaciones ocultas, patrones y tendencias que no son fácilmente evidentes en las hojas de cálculo o paquetes estadísticos y crear nueva información que apoye en la toma de decisiones. Los SIG son una clase especial de sistemas de información cuya clave es la ubicación, para vincular la geometría y la toponimia, entonces, éste es el escenario donde transcurren y pasan los eventos, problemas, actividades y cosas. De esta manera, los problemas que implican un aspecto de ubicación, ya sea en la información utilizada para resolverlos, o en las propias soluciones, se denominan problemas geográficos y tienen una variedad de aplicaciones:

1. **Servicios de salud.**- Los directores de la salud cuando deciden donde crear nuevas clínicas y hospitales, requieren de un marco geográfico Lang L. (2000) *GIS and health organizations* with CDROM. ESRI Press, Redlands [31].

---

<sup>3</sup>El lector puede recurrir al artículo *Automated Geographic Information Systems on Landsat Data: A Survey* [1], de Ellen M. Knapp y Deborah, se expone un cuadro sumario de los sistemas de computadoras y organizaciones en contra de los estándares de las categorías y características.

2. **Logística y transportación.** Lang L. (1999) *Transportation GIS*. ESRI Press, Redlands/CA [32]. Las compañías de mensajería resuelven problemas geográficos cuando deciden rutas y horarios de sus vehículos, a menudo sobre una base diaria. Las autoridades de transporte requieren de un estudio espacial para seleccionar rutas para nuevas autopista.
3. **Sociología y demografía.** Los consultores de geodemografía<sup>4</sup> en el problema de crecimiento demográfico requieren de la planeación urbana y regional: Greene RW (2000) *GIS in public policy with CDROM*. ESRI Press, Redlands [33]; Martin D (1996) *Geographic Information Systems: socioeconomic applications*. Routledge, London [34].
4. **Gestión de recursos naturales.** Leward A.S, Valenzuela CR (eds) (1991) *Remote Sensing and Geographical Information Systems for Resource Management in Developing (Eurocourses: Remote Sensing 1)* [35]. Las compañías forestales que resuelven problemas geográficos cuando ellos determinan la mejor manera de como manejar los bosques, donde cortar, donde localizar caminos y donde plantar nuevos arboles, para causar el menor daño ecológico.
5. **Uso de suelo y planificación regional.**- Las autoridades de parques nacionales resuelven problemas geográficos cuando programan el mantenimiento y mejora de rutas recreativas. Los agricultores que requieren información geográfica para el uso de suelo e hidrología para ayuda a la toma de decisiones en los cultivos y producción agrícola: *Geographical Information Systems in hydrology* (Water Science and Technology Library 26) Kluwer, Dordrecht [36].

## 2.2. Conceptualización del mundo real en los SIG

Tradicionalmente las personas, las organizaciones y el gobierno han suministrado información que en computación ha sido codificada como “datos” y determinados como valores de una entidad como numérica (intervalo/razón) o categoría (nominal/ordinal). De esta manera los atributos son los que constituyen al objeto mismo, o bien como la característica de medición del mismo objeto.

Los SIG coleccionan información en los atributos n-dimensional pero vinculada a una georeferencia en un espacio geográfico donde suceden los eventos en el tiempo de tal forma que tenemos una base de de datos espacio-temporal (ver Figura 2.2).

La codificación de datos y procesamiento de entrada desde el surgimiento de los SIG fue uno de los problemas principales por atender, de como la información relevante para

---

<sup>4</sup>Geodemografía: Ciencia que une la geografía y la demografía, la cual busca entender los procesos por los que los asentamientos, ciudades y barrios se forman y evolucionan.

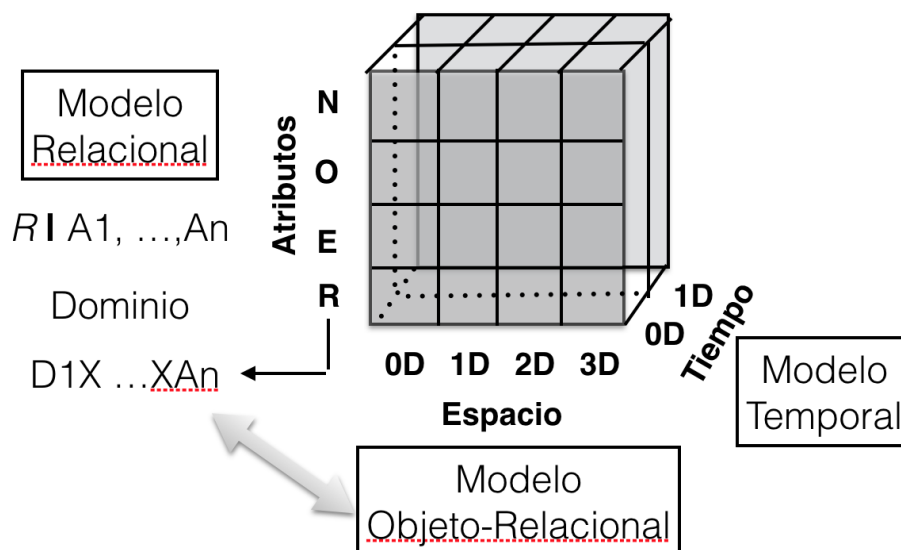


Figura 2.2: Cubo multidimensional del espacio, tiempo y atributos, donde cada tema apunta a un modelo de base de datos (De acuerdo con [2]).

cada tipo de dato debería ser obtenida, considerando la exactitud, la precisión y las características espaciales de los datos (por ejemplo, sistema georeferenciado, escala) [5]. Los involucrados en el SIG acerca de la captura de datos y la conversión son cubiertos por Hohl en su libro “GIS data conversion” [Hohl, P. 1998] [37]. De esta forma los tipos de datos espaciales para ser referidos como una capa de datos espaciales (ver Figura 2.3), donde cada capa de datos espaciales se tienen cuatro tipo de entidades geográficas representadas como puntos, líneas y polígonos. En este punto es cuando se hace referencia al uso de una malla de celdas como un identificador de localización donde las entidades podrán ser generalizadas y almacenadas. La retícula de celdas esta constituida áreas regulares, irregulares, triangulares irregulares o alguna otra representación de malla [5].

Por ejemplo, en la Figura 2.3 se ilustra como el mundo real abstrae en un número de capas de datos espaciales, incluyendo el uso del suelo, la elevación y algunos otros temas. Estas capas ciertamente se utilizan para crear mapas compuestos, pero también, se utilizan para generar una variedad de información útil a través de análisis del SIG. Esta información pueda ser usada para contestar preguntas tan cruciales como las siguientes: En un sistema geográfico en el caso de un siniestro de inundación nos podemos preguntar ¿Qué áreas serían afectadas?, ¿Qué hogares requerirán de reparación de servicios?, ¿Cómo llevar a cabo un plan de desastre DN-3?. En consecuencia, el uso que pueda hacer el gobierno de este sistema es como un sistema de información geográfica de riesgo o planeación

estratégica.

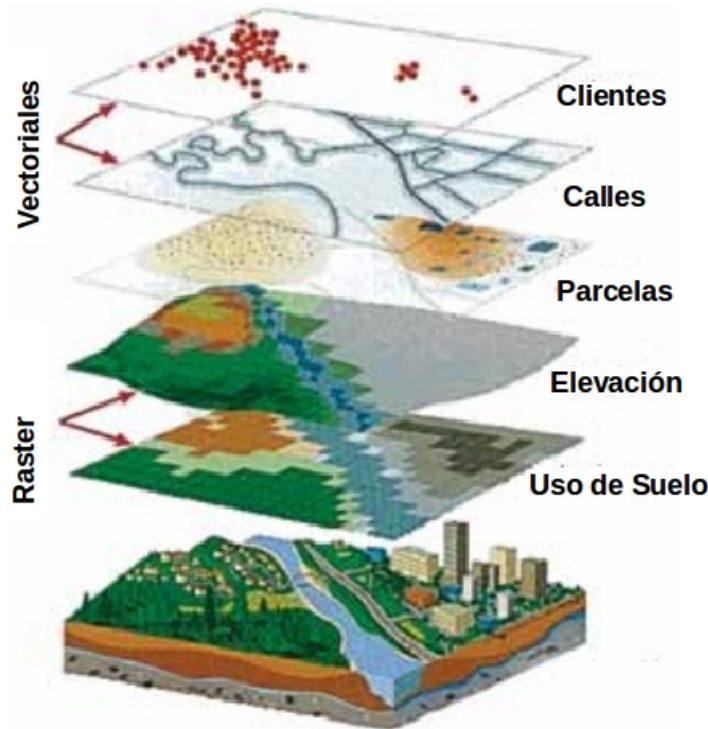


Figura 2.3: Capas de datos geográficos de acuerdo con [3].

Con lo anterior queremos mostrar de manera simplificada como en un SIG se conjuga un gran conjunto de datos de diferentes tipos, pero principalmente todos georeferenciados, donde la visualización y el análisis espacial ha sido un objetivo principal de los profesionales desarrolladores de SIG. De esta forma el modelado de los fenómenos del mundo real se convierte en la tarea principal y más difícil para la construcción de un SIG. Los modelos son eficaces representaciones significantes de construcciones mentales usadas como instrumentos para ciertos propósitos particulares. En este caso los modelos son representaciones de la realidad con un alto significado, útil, eficaz y que nos presentan ventajas por su visión unitaria.

Hesse Mary B., en la referencia “Models and Analogy in Science”, en P. Edwards (ed.), *Encyclopedia of Philosophy*, vol. 5, p. 356. [38] menciona:

“The theoretical models carries with it what has been called ‘open texture’ o ‘surplus meaning’, derive the familiar system. The theoretical model conveys associations and implications that are not completely specifiable and that may be transferred by analogy to the explanandum [the phenomenon to the explained]; further developments and modifications of the explanatory theory may therefore be suggested by the theoretical model. Because the theoretical model is richer than the explanandum, it imports concepts and

conceptual relations not present in the empirical data alone.”

En la geografía como una ciencia empírica, la concepción general de un modelo se restringe al hecho, donde la calidad del modelo deberá ser verificable por encontrar los sucesos empíricos (Hacke & Grünreich 1994) *Kartographic* [39]. Por la aplicación dentro de un SIG la definición deberá ser extendida al hecho, que un modelo representa una parte del mundo real, es representada por datos. Goodchild [40] por ejemplo define “un modelo de datos como una representación limitada de la realidad, construida por la naturaleza finita y discreta de los dispositivos informáticos” (Goodchild 1992: 401) [40]. Esta definición se dirige al hecho de que la complejidad del mundo real debe ser reducida, de modo que pueda ser almacenada en un SIG.

En relación con una complejidad reducida, el modelado del mundo real implica la pérdida de mucha información y significado durante el proceso. Tenemos que elegir qué información se incluye y cual es descuidada. La decisión de una pieza de información es importante o no. Aunque es descuidado, la decisión sobre la información a incluir se basa en nuestra opinión de si una información es importante o no. Al hacer esto, ejecutamos una reinterpretación distinta del mundo real. Por lo tanto, un modelo nunca es válido para todos los propósitos.

La reducción de la complejidad del mundo real durante el modelado de un SIG conduce a resultados diferentes. El propio proceso de modelado no sólo está relacionado con el mundo real, sino también con las limitaciones impuestas por la tecnología SIG. Por lo tanto, el resultado del proceso de implementación varían considerablemente y muchos errores son cometidos. La representación del mundo real pasa por alto los fenómenos importantes, o el modelado representa a el mundo real según los objetivos del SIG. Otro tipo de problemas son los conceptos técnicos erróneos, de modo que los datos en el SIG no procesan, analizan y presentarse como se desee. Por lo tanto, es altamente recomendable no construir un SIG basado en un análisis *ad-hoc* de los requisitos. Existen varias etapas de planificación que ayudan a prevenir errores hasta el fracaso completo del proyecto SIG (ver Figura 2.4).

Primero deben definirse los temas generales del SIG. Los problemas que se resolverán deben ser identificados y los objetivos deben ser declarados. El segundo paso en la construcción de un SIG es la conceptualización del mundo. Para esta etapa es necesaria una gran experiencia y conocimientos teóricos sobre la naturaleza de los fenómenos reales del planeta. A raíz de las decisiones sobre cómo y por qué entidades y datos debe representarse el mundo real en el SIG, desarrollar las estructuras de datos que almacenan la información pertinente.

Durante la siguiente fase, los datos deben ser recolectados y la base de datos ser construida. Además, los objetos de geometría tienen que ser digitalizados o importados al SIG. Este es un punto crítico, donde todas las decisiones tomadas antes deben ser reevaluadas



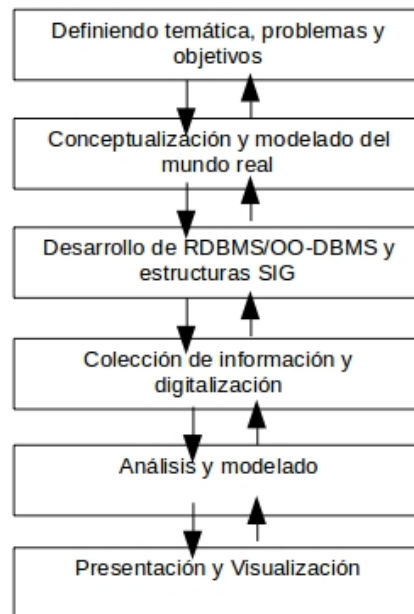


Figura 2.4: Pasos en el desarrollo de un SIG (adaptado de Hendrick & Ottens en 1997 [4])

cuidadosamente. Los datos recopilados para un SIG son a menudo la parte más cara del proceso de construcción.

El análisis y el modelado posterior son la parte más interesante del proceso de construcción. En ese momento, los errores o malas interpretaciones realizados antes tendrían impactos cruciales. Finalmente, los resultados se visualizan y presentan.

### Etapas de abstracción en el modelado de un SIG

El proceso de representación del mundo real en un SIG se separa en diferentes etapas, empezando por la conceptualización del mundo real (ver Figura 2.5, Peuquet 1990: 252 [41]). Como se discutió anteriormente al inicio de la sección 2.2, el mundo podría ser entendido como un continuo espacio-temporal. Nuestra comprensión y conceptualización depende de nuestra visión del mundo, nuestra experiencia, nuestros conocimientos teóricos y, por último, pero no menos importante, nuestro interés y nuestro objetivo.

Debido a que este tipo de modelado trata del significado, la noción y los términos, también es abordado como modelado semántico de la realidad [5]. El modelo espacial conceptual rara vez se aplica directamente para almacenar datos en un sistema de información. En una segunda etapa hay que construir elementos claramente definidos del mundo real. Estos elementos se denominan comúnmente entidades. La relación de tales elementos formales, por ejemplo es representada en un modelo de entidad-relación (modelo ER).

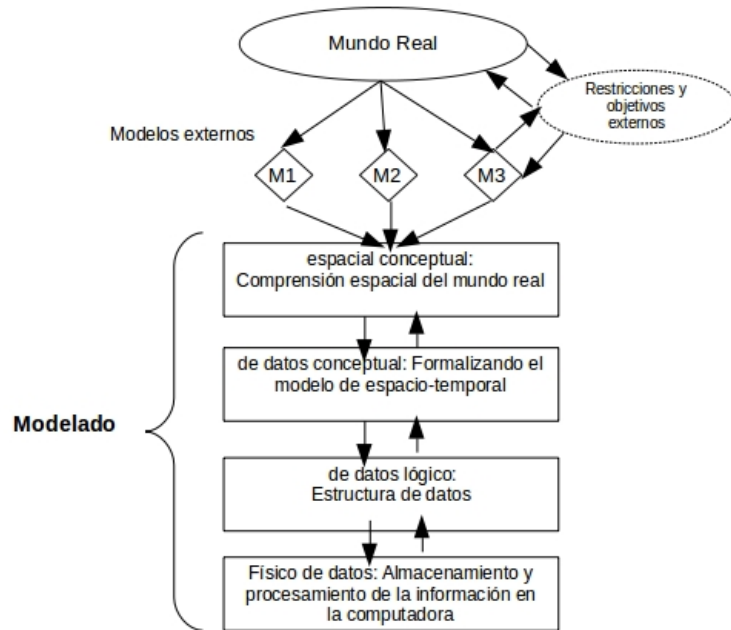


Figura 2.5: Conceptualización del mundo real en un SIG (adaptado de Hendrick & Ottens 1997 [4])

La información asimilada a tales objetos finales y discreta se almacena en un sistema de información. Esta etapa se llama el modelo lógico de datos. En este nivel, la información no tiene estructura inherente. Existen diferentes maneras en que los datos son estructurados lógicamente, dependiendo de la tecnología de base de datos que se esté utilizando (Laurini & thompson 1992 [8]). La transferencia del modelo de datos conceptuales al lógico de datos es un proceso intelectual. La estructura formal dada en el modelo conceptual guía este proceso. A menudo, el modelo ER se utiliza para el desarrollo de tablas en un sistema de gestión de bases de datos relacional.

La diferenciación entre el nivel conceptual y el nivel lógico del proceso de modelización no debe considerarse como un fin en sí mismo. El nivel conceptual fomenta el objetivo del modelado, la interpretación del mundo real y el conocimiento utilizado para definir las entidades que se utilizan para reducir la complejidad a un nivel que se almacene en una base de datos. El nivel lógico en contraste enfoca el interés en los métodos informáticos utilizados para facilitar estos objetivos [5].

El modelo físico, que marca el final del proceso de modelado, no es accedido por el usuario. Describe el almacenamiento físico de los datos en los medios de almacenamiento.

### Conceptualización del mundo real en un SIG

Este proceso es representado como anillos concéntricos con una etapa decreciente de abstracción y una etapa creciente de reducción (ver Figura 2.6). La conceptualización de un fenómeno del mundo debe restar la complejidad, de modo que se pueda distinguir un elemento o proceso distinto. Se utiliza una comprensión de la dimensión espacio-temporal de los fenómenos para derivar una representación espacio-temporal formalizada (ver Figura 2.7).

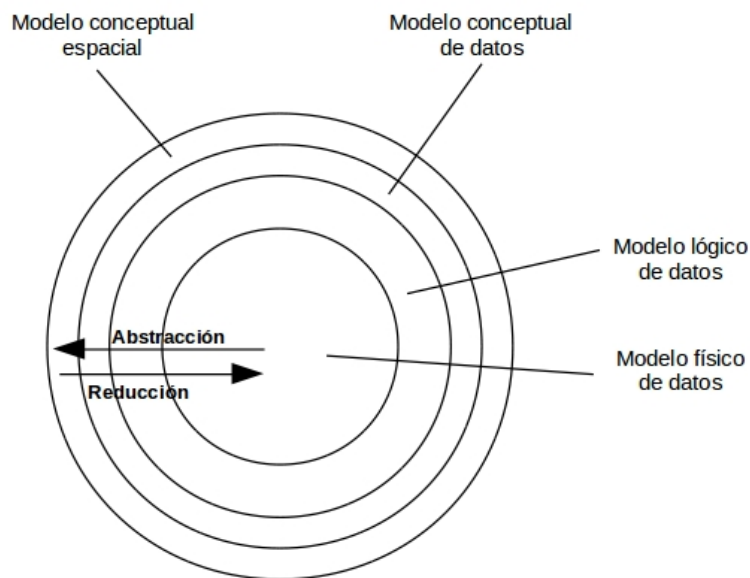


Figura 2.6: Etapas del modelado de datos (de acuerdo con [5])

Por ejemplo, un proceso está representado por puntos, indicando la ubicación de ciertos eventos. Otra visión del proceso conceptualiza la ocurrencia de los acontecimientos según su difusión espacio-temporal mediante líneas. Otra vista podría utilizar regiones para sumar el número de eventos en un área dada o simbolizar la densidad de tales eventos aplicando líneas de contador.

### Entidades y objetos

La terminología utilizada para los SIG es heterogénea. Los nombres utilizados para describir los elementos del mundo real y el elemento del SIG son engañosos. Algunas “características” y “objetos” se utilizan para diferenciar entre la esfera del mundo real y los datos utilizados en un SIG sin distinción y “característica” adicionales para los datos correspondientes almacenados en una base de datos. (Lange 1997 [42]).

Los objetos se definen por su ubicación y están relacionados con los datos de los atributos

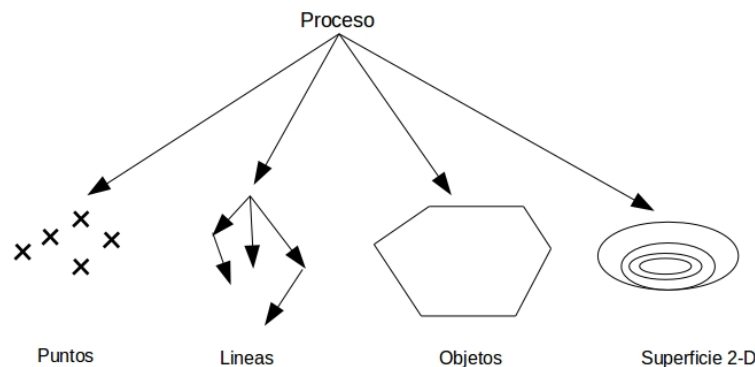


Figura 2.7: Diferentes conceptualizaciones espaciales de un proceso (de acuerdo con Raper (1999) [6])

que varían. Las características se definen por su contenido, pero la forma y la posición cambian con el tiempo. La entidad se aplica a elementos del mundo real, que se distinguen de otros elementos, usando características distintas del elemento. De acuerdo con Laurini & Thompson (1992) [8] “una entidad es una persona, un lugar, una cosa o un acontecimiento, la parcela del Sr. Green, el edificio 2314, el número 34HP89, el vuelo 123 son entidades en el mundo real”. Las entidades también consisten en otras entidades de otro tipo.

La representación espacio-temporal de una entidad en un SIG es un objeto. Los objetos consisten en una dimensión espacial, una dimensión temporal y otros datos de atributo (ver Figura 2.8). La dimensión espacial del objeto se modela en maneras diferentes. El objeto en SIG debe estar estructurado, para que el SIG pueda abordarlos. Están disponibles dos estrategias conceptuales: la organización de la información, ya sea a nivel de entidades u objetos, se hace algunas veces utilizando capas o un modelo ER.

### Modelo GIS de capa

Se aplica en la mayoría de los paquetes de GIS de escritorio. Usando capas, la información se estructura lógicamente según diferentes criterios. Las entidades u objetos se organizan de manera que cada capa conste de un solo tipo de información. El concepto de capa es similar con la superposición de transparencias utilizadas en la cartografía tradicional, donde se utiliza una capa para cada tipo de información en el mapa (ver Figura 2.9).

### Modelo Entidad-relación

En un modelo ER la relación entre entidades se estructura de acuerdo con la jerarquía de las entidades derivadas de la información de atributo relacionada (ver Figura 2.10). El modelo objeto-relacional-modelo de los objetos de los objetos-relacional del modelo. Cada objeto consiste en los datos de atributos y métodos de objetos relacionados de funciones.

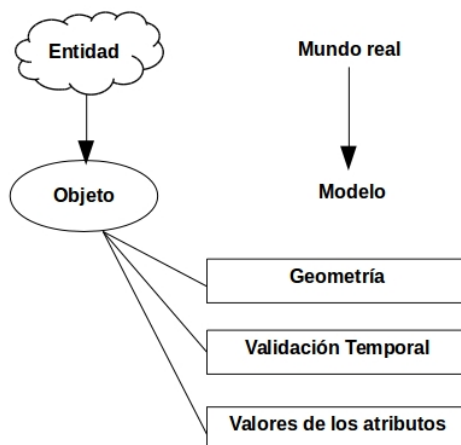


Figura 2.8: Dimensiones de una entidad representada por un objeto (de acuerdo con Behr (1998) [7] )

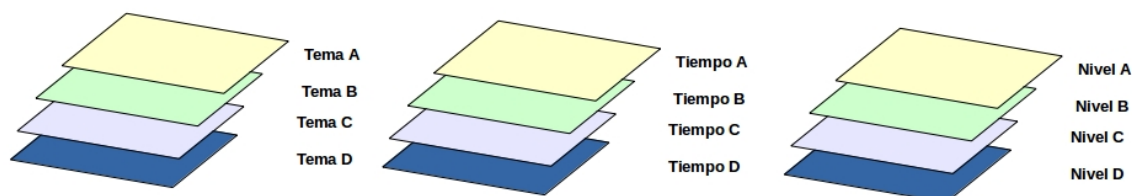


Figura 2.9: Modelo Capa-SIG (de acuerdo con Laurini & Thompson (1992) [8], [5])

## 2.3. Modelos de datos

Para visualizar un fenómeno natural, primero debemos determinar cual es la mejor manera de representar ese espacio geográfico. Los modelos de datos son un conjunto de reglas y/o constructores usados para describir y representar aspectos del mundo real en la computadora. Los datos espaciales en los SIG tienen dos principales formatos (a disposición de los datos para su almacenamiento o visualización). Los SIG actuales integran los formatos vectorial o raster (ver Figura 2.11). Cada formato tiene ventajas y desventajas.

### 2.3.1. Modelo de datos Vectorial

Es el modelo más comúnmente usado en la planificación. Los datos vectoriales usan puntos y están asociados a pares de coordenadas  $[x,y]$  para representarlos vértices de las características espaciales. Los atributos de los datos de estas características son almacenados en un sistema de gestión de base de datos independientes.

Existen tres tipos fundamentales de datos vectoriales: puntos, líneas y polígonos. Puntos son objetos de una sola dimensión que solo contienen un solo par de coordenadas. Los

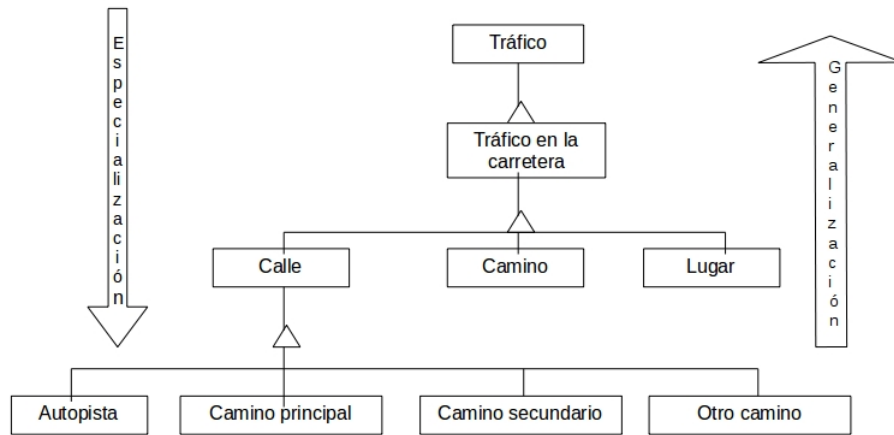


Figura 2.10: Modelo ER (de acuerdo con [5])

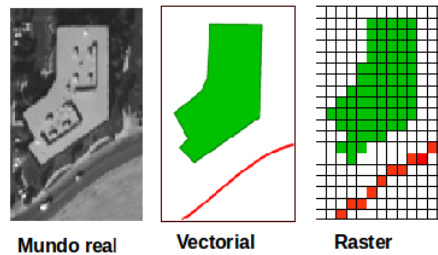


Figura 2.11: Formato raster y vectorial

puntos son típicamente usados para modelar características singulares y discretas, tales como edificios, pozos, postes de electricidad, los sitios de muestreo y así sucesivamente. Las líneas son unidimensionales compuestas de múltiples puntos conectados. Las líneas son usadas para representar características tales como caminos, arroyos, fallas, límites y así sucesivamente. Los polígonos son bidimensionales creados por múltiples líneas que se cierran para crear una figura o forma.

En el caso de los polígonos, el primer par de coordenadas (punto) sobre el primer segmento de línea es el mismo que el último par de coordenadas sobre el último segmento de línea. Los polígonos son usados para representar características tales como límites de la ciudad, formaciones geológicas, los lagos, las asociaciones de suelos, comunidades vegetales, etc. Los polígonos también son llamados áreas. Este formato de datos tiene diferentes ventajas que se describen en la Tabla 2.1.

Vector	
Ventajas	Desventajas
Los datos geográficos son más precisos y creíbles.	Análisis espacial y filtrado.
Estructuras de datos complejas.	Algunos formatos vectoriales son difíciles de manejar que los formatos raster.
Compacto.	
Fácil de entender.	
Datos son representados en su resolución original.	

Tabla 2.1: Ventajas y Desventajas del formato Vectorial

### 2.3.2. Modelo de datos Raster

El formato de datos Raster es ampliamente usado en aplicaciones que van mucho más allá de los SIG. Este formato es muy familiar con las fotografías digitales. La ubicuidad de los formatos de los archivos JPEG, BMP y TIFF están basados en el formato de datos raster. Los datos raster para los SIG son obtenidos de fotos aéreas o satelitales. Este formato consiste de filas y columnas de igual tamaño de píxeles interconectados para formar una superficie plana. Estos píxeles se utilizan como bloques de construcción para la creación de puntos, líneas, áreas, redes y superficies. A pesar que los píxeles son triángulos, hexágonos, o aún octógonos, los píxeles cuadrados representan la forma geográfica más simple a utilizar. La mayoría de los datos raster en los SIG es construida sobre píxeles cuadrados. Estos cuadros son típicamente usados para formar rectángulos de varias dimensiones si el modelo de datos es transformado de una proyección a otra. La principal diferencia entre el modelo raster y vectorial es reflejo de la “pixelación” de un raster, lo cual serían puntos, líneas y polígonos en un modelo de datos vectoriales. Los datos raster son particularmente adecuados para ciertos tipos de análisis, tales como geoprocésamiento básico, análisis de superficies y mapeo de terrenos. Los datos raster simplifican muchos tipos de análisis espacial, que serían más complicados de realizar con conjuntos de datos vectoriales. En la Tabla 2.2 se muestra la comparativa entre las ventajas y desventajas de este formato.

### 2.3.3. Estándar de tipos de datos y funciones geográficas

Han existido muchos intentos de definir un superconjunto de tipo de datos geográficos para representar y procesar datos geográficos en bases de datos. La comunidad de los SIG han trabajado bajo los auspicios de la ISO (*International Standards Organization*) y OGC (*Open Geospatial Consortium*) [9], los cuales han definido los principales tipos y funciones geográficas para ser usados en un *DBMS* y ser accedidos usando el lenguaje Structured

<b>Raster</b>	
Ventajas	Desventajas
Estructuras de datos simples.	Imprecisión espacial.
Incrustación y combinación de mapas.	Necesitan mayor espacio de almacenamiento.
Simulaciones fáciles, debido a que las celdas tienen el mismo tamaño y forma. Fáciles de entender.	Las imágenes raster no se escalan de forma idónea.  Están basadas en píxeles cuadrados.
Las imágenes de teledetección son típicamente en formato raster.	Las imágenes raster son menos “bonitas”, esto si se realiza un aumento, la imagen pierde detalle y coherencia.
	Problemas al cambiar la proyección de la imagen.

Tabla 2.2: Ventajas y Desventajas del formato Raster

Query Language (SQL). Los tipos de datos geográficos son mostrados en la Figura 2.12, en un diagrama UML estándar, para detallar cada tipo en el idioma Inglés. La clase “geometry” es la clase raíz. Las clases *Point*, *Curve*, *Surface* y *GeometryCollection* son todos los subtipos de *Geometry*, los cuales se encuentran en la publicación *OpenGIS Implementation Specification for Geographic information-Simple feature access-Part 1: Common architecture* [9].

Acorde a el estándar ISO/OGC [9] [10], existen 9 métodos para probar las relaciones espaciales entre estos objetos geográficos. Cada uno toma como entrada dos geometrías (colecciones de uno o más objetos geométricos) y evalúa si la relación es verdadera o no. La geometría base es un superconjunto de la geometría de la comparación. El conjunto completo de los operadores Booleanos para probar las relaciones espaciales entre geometrías son:

- Equals. Las geometrías son iguales.
- Disjoint. Las geometrías comparten un punto en común.
- Intersects. Las geometrías se intersecan.
- Touches. Las geometrías se intersecan en sus límites o fronteras.
- Crosses. Las geometrías se solapan o se superponen.



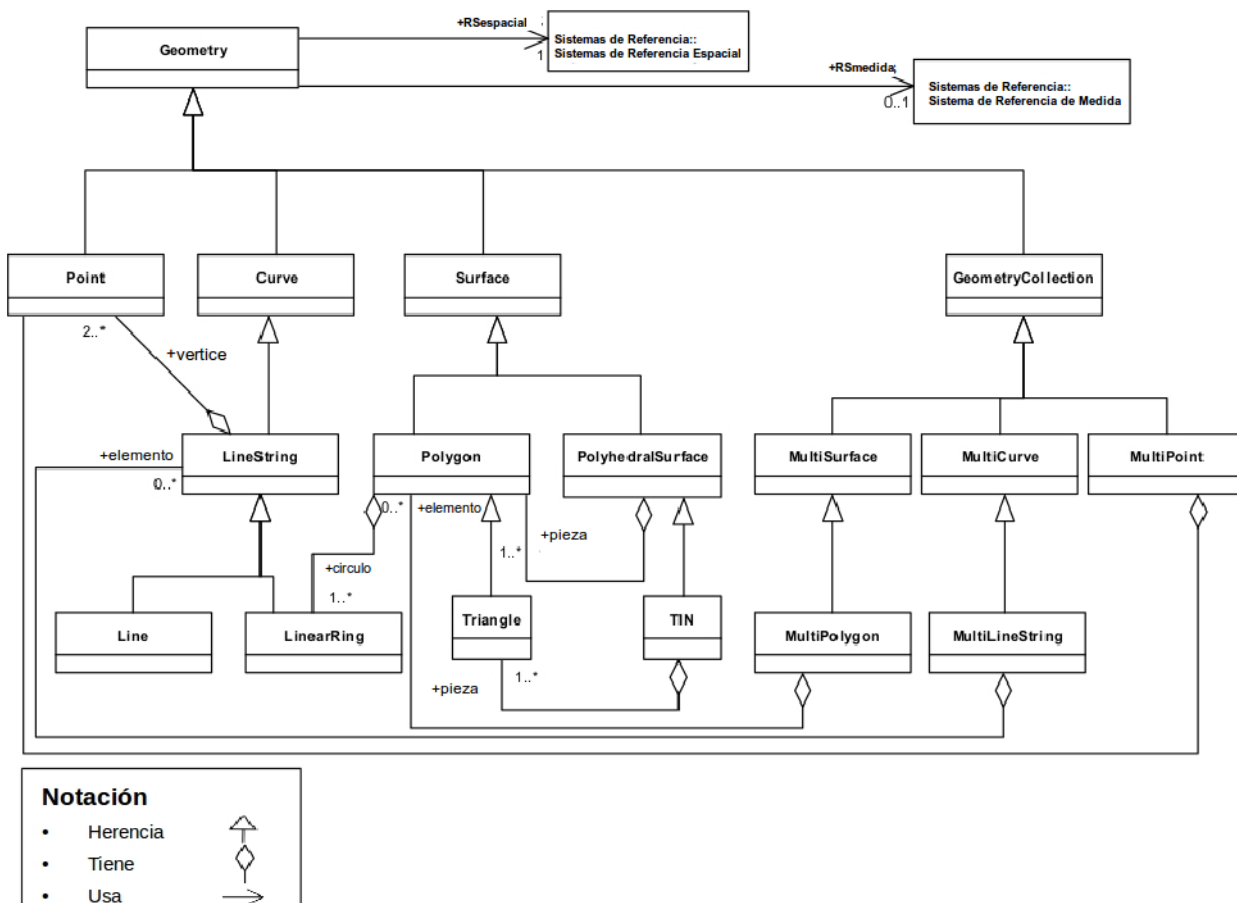


Figura 2.12: Modelo de los tipos de datos geográficos (de acuerdo a la OGC [9])

- Within. Esta la geometría dentro de otra.
- Contains. La geometría contiene completamente a otra.
- Overlaps. Las geometrías se superponen
- Relate. Existen intersecciones entre el interior, el limite, o el exterior de las geometrías.

Existen también siete metidos [9] [10] que apoyan el análisis espacial para las geometrías anterior mente mencionadas.

- Distance. Determina la distancia más corta entre dos puntos cualquiera en dos geometrías.

- **Buffer.** Devuelve una geometría que representa todos los puntos cuya distancia de la geometría es de menos o igual a una distancia definida por el usuario.
- **ConvexHull.** Devuelve una geometría que representa la envolvente convexa de una geometría (un *convex hull* (Cápsula convexa) es el polígono más pequeño que encierra otra geometría sin ningún tipo de zonas cóncavas).
- **Intersection.** Devuelve una geometría que contiene solamente los puntos en común con las dos geometrías de entrada.
- **Union.** Devuelve la geometría que contiene todo los puntos de las geometrías de entrada.
- **Difference.** Devuelve una geometría que contiene los puntos que son diferentes entre dos geometrías.
- **SymDifference.** Devuelve una geometría que contiene los puntos que se encuentran en cualquiera de las geometrías de entrada, pero no ambos.

## 2.4. Bases de datos espaciales

Las bases de datos espaciales son un tipo especial de BD, donde lo importante es la información espacial. Por lo tanto es esencial observar la definición de las BD, para entender por que este tipo de BD se diferencia de los demás sistemas de datos comunes.

### 2.4.1. Sistemas de Gestión de bases de datos (DBMS)

Una BD es definida como una colección grande de datos almacenados interrelacionados dentro de un entorno computacional. En este entorno los datos son persistentes, lo cual significa que sobreviven a problemas de software o hardware inesperados. Las características más importantes de las bases de datos son un gran volumen de datos y persistencia. Las aplicaciones tradicionales de bases de datos incluyen el manejo de personal, valores, reservaciones de viaje y manejo bancario.

Una base de datos es vista como uno o demasiados archivos sobre algunos dispositivos de memoria externos, tales como un disco. A pesar que es posible escribir aplicaciones que accedan directamente a los archivos, como una arquitectura podría plantear una serie de problemas relacionados con la seguridad, concurrencia y la complejidad de la manipulación de datos. DBMS (Database management system) es una colección de *software* que maneja la estructura de la base de datos y controla el acceso a los datos almacenados en una BD. Un DBMS facilita el proceso de:

- definiendo una BD, que especifica los tipos de datos, estructuras y restricciones a ser tomadas en cuenta.
- construcción de la BD, que almacena los datos por si mismas en un almacenamiento persistente.
- manipulación de la BD.
- consulta de la BD para conseguir datos específicos.
- actualización de la BD (cambio de valores).

Un DBMS gira en torno a el concepto fundamental de la *independencia de los datos*. Los usuarios interactúan con una representación de los datos de forma independiente del almacenamiento *físico* y el DBMS es el encargado de traducir las acciones del usuario en las operaciones eficientes de estructuras de datos físicos. Lo anterior es muy diferente del procesamiento de archivos, en el que la estructura de un archivo, junto con las operaciones de este fichero, se integran en un programa de acceso. Este mecanismo es factible mediante el uso de diferentes *niveles de abstracción*.

Se distinguen 3 niveles en un entorno de de BD. El *nivel físico* trata las estructuras de almacenamiento; el *nivel lógico* define la representación de datos propuestas por el usuario; y el *nivel exterior* corresponde a una vista parcial de la base de datos provista en una aplicación particular.

### 2.4.2. Sistema de gestión de bases de datos relacionales(RDBMS)

Un Sistema de gestión de bases de datos relacionales (RDBMS) es una colección de tablas que están conectadas de tal manera que los datos son accedidos sin reorganizar las tablas. Las tablas son creadas de tal forma que cada columna representa un atributo particular (por ejemplo, apellido, número de teléfono, entre otros) y cada fila contiene una instancia única de datos para cada atributo.

En el modelo relacional, cada tabla es enlazada a cada una de las otras tablas, por medio de claves predeterminadas. La clave principal representa el atributo (columna), la cual tiene un valor único para identificar un registro en particular en la tabla.

### 2.4.3. Bases de datos espaciales

Una de las características de los SIG son los datos y donde deben ser almacenados. Desde 1990 se han incrementado las aplicaciones SIG que son usadas para el análisis espacial. BDE almacenan representaciones geográficas del mundo real para ser usadas por un SIG. Estas BDE utilizan técnicas adicionales diferentes de tablas para almacenar estas representaciones espaciales. Cabe señalar que una base de datos espacial no es lo mismo que un SIG, a pesar que ambos sistemas comparten varias características. Las cuales incluyen

las funciones para una BD en general: concurrencia, almacenamiento, integridad y consulta, pero no sólo datos espaciales. Un SIG por otra parte, se adapta para operar datos espaciales.

Una base de datos espacial o geográfica define los tipos de datos espaciales para los objetos geométricos y permite el almacenamiento de datos geométricos en tablas regulares de una BD. BDE proporcionan funciones y índices especiales para consultar, manipular y actualizar los datos geoespaciales utilizando el lenguaje SQL. Una BDE es a menudo un recipiente de almacenamiento para los datos espaciales. Aunque una BDE no tiene por qué ser de naturaleza relacional [43]. Además, una base de datos espacial proporciona herramientas de almacenamiento y de análisis.

Entre los sistemas de gestión de BD, que cuentan con soporte de datos espaciales, están las versiones recientes de Oracle y Microsoft SQL Server. Estos DBMS ofrecen datos espaciales nativos, sus cadenas contienen cadenas de caracteres, números e incluso líneas, polígonos y puntos. Los datos espaciales en una BDE son almacenados en una columna especial conocida como “*geometry column*”.

Una BDE permite a una amplia variedad de usuarios acceder a grandes conjuntos de datos (sean geográficos o alfanuméricos) y el manejo de sus relaciones, garantizar su integridad. La Open Geospatial Consortium (OGC) [44] ha liberado una serie de estándares relacionados a las bases de datos geográficas que definen:

- Qué tablas deben estar presentes en una BDE.
- Los formatos de datos, denominados “características simples”.
- Un conjunto de instrucciones SQL para el análisis geográfico.

La arquitectura de una base de datos espacial se diferencia de un RDBMS estándar no sólo porque maneja datos de la geometría y gestionar proyecciones, sino también por un mayor conjunto de comandos que se extienden del lenguaje SQL estándar (por ejemplo, los cálculos de distancia, superposición, la conversión entre sistemas de coordenadas, etc.).

## SQL Espacial

La implementación del OGC “Características sencillas para SQL” ofrece nuevas y poderosas funciones de GIS para la gestión, recuperación y análisis de datos geoespaciales. El dominio espacial introduce un nuevo conjunto de funciones en el lenguaje SQL [45]. El prefijo ST suele estar reservado para las funciones SQL / MM en otras bases de datos espaciales. La Figura 2.13, muestra de forma gráfica el funcionamiento de funciones espaciales.

## Métodos espaciales

La mayoría de todos los métodos espaciales se agrupan en una de las cuatro categorías siguientes:

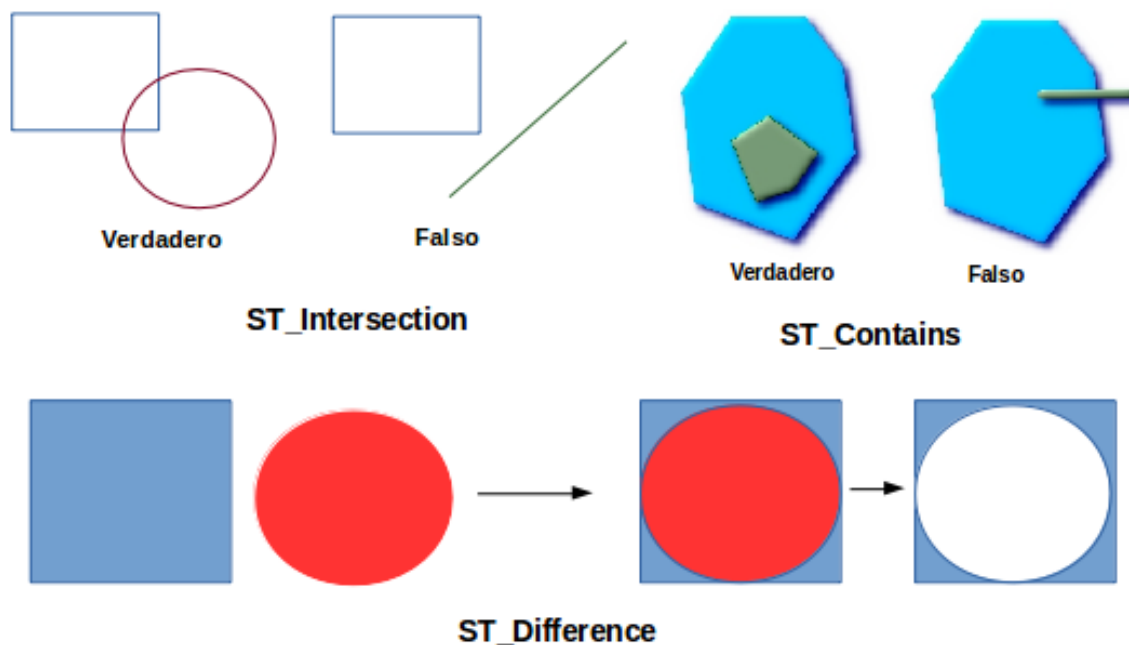


Figura 2.13: Funciones SQL Espacial: ST\_Intersects, ST\_Contains y ST\_Difference

- Convertir entre geometrías y formatos de datos externos.
- Recuperar propiedades o medidas de una geometría.
- Comparar dos geometrías con respecto a su relación espacial.
- Generar nuevas geometrías de otras.

### Comparar dos geometrías

Una parte muy interesante en las consultas espaciales viene de la comparación de geometrías. Las preguntas como “qué edificios están en una zona de inundación” o “dónde están las intersecciones de carreteras y calles ferroviarias” sólo son contestadas si se comparan entre sí las geometrías que representan los edificios, las zonas de inundación, las carreteras de ferrocarril y las calles.

La norma define el siguiente conjunto de métodos para comparar geometrías de varias maneras.

- ST\_Equals prueba la igualdad espacial de dos geometrías.
- ST\_Disjoint prueba si dos geometrías no se cruzan.

- ST\_Intersects, ST\_Crosses y ST\_Overlaps prueban si los interiores de las geometrías interceptarse.
- ST\_Touches prueba si dos geometrías tocan sus fronteras, pero no se cruzan en sus interiores.
- ST\_Within y ST\_Contains prueba si una geometría está completamente dentro del otro.

### Generar nuevas geometrías

El último conjunto de métodos permite al usuario generar nuevas geometrías a partir de las ya existentes. Una geometría recién generada puede ser el resultado de una operación de conjunto en el conjunto de puntos representados por cada geometría, o puede ser calculada por algún algoritmo aplicado a una sola geometría. Los siguientes métodos son ejemplos para ambos.

- ST\_Buffer genera un buffer a una distancia específica alrededor de la geometría dada.
- ST\_ConvexHull calcula el casco convexo para una geometría.
- ST\_Difference, ST\_Intersection y ST\_Union construyen la diferencia, intersección o unión entre los conjuntos de puntos definidos por dos geometrías.

## 2.5. Arquitectura de los SIG

Los SIG no solo se componen de datos geográficos, la arquitectura de estos sistemas contiene diferentes componentes para hacer al sistema funcional. Los SIG tienen cinco componentes esenciales en su arquitectura, como el hardware, los conjuntos de módulos de aplicación del software, datos requeridos, las personas que van a manejar el sistema y un buen diseño del método de implementación [46].

### Hardware

Hardware se refiere a los componentes físicos de un sistema computacional sobre el cual un SIG opera. Un SIG opera sobre una computadora para almacenar y procesar datos. El tamaño del sistema computacional dependerá sobre el tipo y la naturaleza del SIG.

### Software

El software SIG provee las funciones y herramientas necesitadas para almacenar, analizar y desplegar información geográfica. EL núcleo de cualquier sistema SIG recae en el mismo

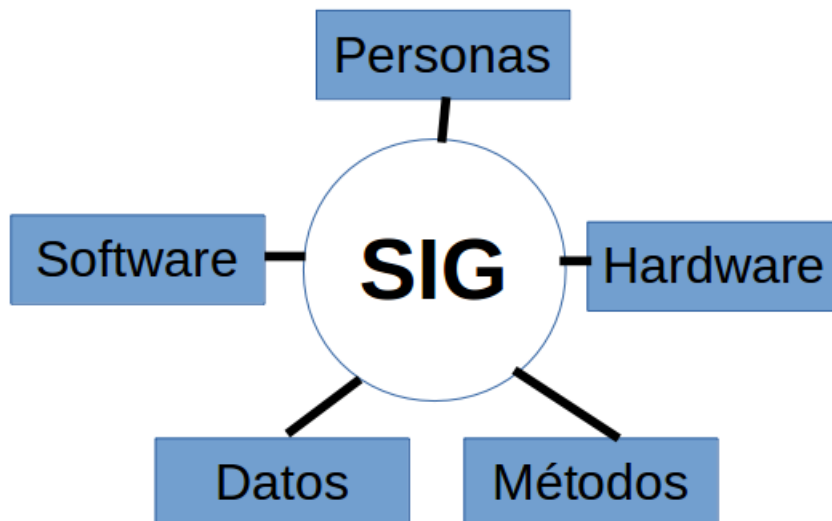


Figura 2.14: Componentes de un SIG.

software SIG, el cual ofrece funcionalidad para guardar, administrar, enlazar, consultar y analizar datos geográficos.

## Datos

El componente más importante de un SIG son los datos. Los datos geográficos y tablas relacionadas, son colectados en casa, compilados para especificaciones y requerimientos personalizados, o comprados de un proveedor de datos comerciales. Un SIG integra datos espaciales con otras fuentes existentes de datos a menudo almacenadas en un *DBMS* corporativo. La integración de datos espaciales (a menudo propiedad del software SIG) y tablas relacionadas en un *DBMS* es la funcionalidad clave proporcionada por un SIG.

Los datos para un SIG vienen en dos formas, datos geográficos o espaciales y atributos de datos espaciales. Los datos espaciales son datos que contienen una ubicación geográfica explícita en la forma de un conjunto de coordenadas. Los datos de atributos son conjuntos descriptivos que contienen diversa información relevante a una ubicación particular, por ejemplo profundidad, altura y cifras de ventas. Estos datos son enlazados a una ubicación particular por medio de un identificador, por ejemplo dirección y código postal. La exactitud de los datos determinan la calidad del SIG.

## Personas

El principal objetivo de un SIG es apoyar a los usuarios con los datos apropiados y las herramientas en la ayuda de toma de decisiones. De esta manera cuidar las necesidades

particulares, que los usuarios deben de dar en los estados de diseño del sistema, tal que cada grupo de usuarios tendrá acceso a los datos y a la funcionalidad del sistema, de manera apropiada. Las personas definen los métodos a emplearse al SIG, entonces, se consideran parte esencial de los SIG.

## Métodos o Procedimientos

Un SIG opera acorde al plan de implementación y a las reglas de negocios. Los cuales son los modelos y las practicas de operación para cada organización. Nuevas herramientas solo se utilizan con eficiencia si se integran adecuadamente a la estrategia empresarial y de operación. Los métodos incluyen como se recuperan los datos, se almacenan, se gestionan, se transforman, se analizan y se presentan en un a salida final.

Existen muchos proveedores de software SIG, tales como *Environmental Systems Research Institute(ESRI)* [47], la cual distribuye ArcGIS. Pitney Bowes distribuye MapInfo GIS. A pesar de los servicios de mapeo en línea e interfaces son proporcionadas por compañías como Google [48], Yahoo! [49] y Microsoft [50], tales servicios no son considerados como plataformas SIG, por no presentar análisis espacial y estar enfocados en la ubicación de lugares y personas. También existen opciones SIG de libre distribución tales como GRASS [51] y QGIS [52] las cuales cuentan con una gran comunidad para su mantenimiento.

## 2.6. Proyección de mapas

El problema más básico de la proyección de mapas es la representación de una superficie curvada en un plano. Adicionalmente, es a menudo el problema de presentación de la tierra o la Luna sobre un mapa plano. La forma de la Tierra es usualmente representada por una sólida revolución, ya sea una elipsoide o una esfera, depende del caso que podría ser, lo cual es considerado como una superficie de referencia para la cual todos los puntos físico esta relacionados.

La tierra tiene una forma esférica redonda grande llamada esferoide. Un globo es una representación muy común y muy correcta de la tierra en forma tridimensional (esferoide). Pero la representación de la tierra en forma de esferoide, tiene sus respectivos problemas como no ser muy portátil y su pequeña escala hace de su uso en la practica muy limitada. Para superar los diversos problemas, es necesario transformar la forma tridimensional de la tierra a una superficie bidimensional como una superficie plana de un papel, una pantalla de una computadora o un dispositivo móvil para obtener las formas más útiles de mapas y escalas de mapa.

Una **proyección cartográfica o de mapas** es una técnica matemáticamente descrita de como representar la superficie de la curvatura de la tierra sobre un mapa plano [44].



Las proyecciones de mapas se refieren a los métodos y procedimientos que se utilizan para transformar la tierra esférica en superficies planas de dos dimensiones [53]. Específicamente, las proyecciones cartográficas son fórmulas matemáticas que son usadas para trasladar latitud y longitud sobre la superficie de la tierra a coordenadas  $x$  y  $y$  sobre un plano (ver Figura 2.15). Desde que existen muchas maneras de realizar esta transformación, existen demasiados números de proyecciones geográficas.

No existe una solución perfecta para el problema de proyección de mapas, pero se podría argumentar que el problema es totalmente irrelevante, ya que es posible limitarnos a modelos tridimensionales de la elipsoide o la esfera, ya sea física construida, como en el caso del globo, o el color del pelo, la religión, entre otras. Para facilitar la creación de un esquema de clasificación para las proyecciones del mapa compuestas de clases y variedades ciertos factores específicos que deben ser considerados:

- La superficie del objeto o datum proyectado
- La superficie de proyección sobre la que se proyecta la superficie de referencia.
- La proyección de la representación “per se” (por si mismo o de por si).

La superficie de proyección es considerada como el problema extrínseco y el procedo de la proyección o representación como el problema intrínseco.

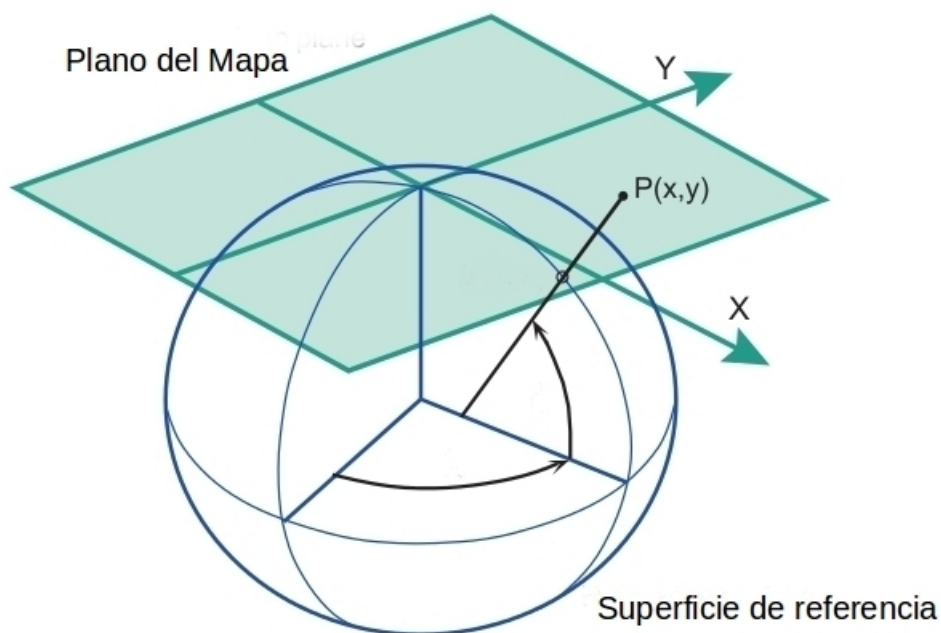


Figura 2.15: Ejemplo de una proyección cartográfica.

### 2.6.1. Sistemas de coordenadas geográficas

Un sistema de coordenadas es un sistema de referencias usado para representar las ubicaciones de elementos geográficos, imágenes y observaciones, como el Sistema de Posicionamiento Global (GPS), ubicaciones dentro de un marco geográfico común. Un sistema de coordenadas es una forma para referenciar, o localizar, todo sobre la superficie de la tierra en un espacio  $x$  y  $y$  [54]. El método usado para representar o describir una parte de la tierra esférica sobre una superficie plana es llamada proyección cartográfica. Los sistemas de coordenadas permiten conjuntos de datos geográficos para usar lugares comunes para la integración.

Un sistema de coordenadas geográficas usa una superficie esférica tridimensional para definir ubicaciones sobre la tierra. Un *GCS* (*geographic coordinate system*) es a menudo llamado incorrectamente un datum, pero un datum es solamente parte de un GCS. Un GCS incluye una unidad angular de medida, un meridiano principal y un datum (basado sobre un esferoide). El esferoide define el tamaño y la forma del modelo terrestre, mientras el datum conecta el esferoide a la superficie terrestre. Los sistemas de coordenadas son necesarios para la expresión de la posición de puntos sobre las diversas superficies, ya sea un elipsoide, una esfera o un avión.

### 2.6.2. Escala

La escala es la relación entre la distancia en un mapa y la distancia correspondiente en el mundo real [55]. Por ejemplo si se tiene una escala de 1:24,000 entonces cada pulgada en el mapa es igual a 24,000 pulgadas (2,000 pies) sobre el terreno. La escala del mapa también es expresada como un condición de equivalencia usando diferentes unidades por ejemplo una pulgada = 2 mil pies. La escala de datos refleja la resolución y su precisión relativa en el mapa, entre más grande la escala, se presenta un conjunto de datos más detallados.

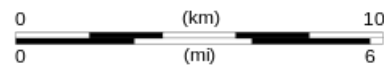
Los números de la escala del mapa son contrarios a la intuición: número más bajo, mayor resolución (y las escala más grande). Así tenemos que una escala de 1:6,000 es significativamente con mayor resolución que una 1: 1000,000. Todos los mapas modelan representaciones del mundo real y, por tanto las características se reducen en tamaño cuando la escala es asignada. En otras palabras, la escala es la medición de la cantidad de reducción de una función de mapeado, tiene una característica asignada que tiene su contra parte real sobre el terreno.

Características extraídas en gran escala (e.g. 1:6,000 ) muestran con mayor detalle el mapa debido a que representan más de cerca las características del mundo real. En un mapa a pequeña escala (e.g. 1:100,000) de las características suelen ser generalizados o agregados. Ambas son útiles y tienen un propósito. Las escalas son factores críticos en el

diseño de los SIG. Si los datos principales de la escala es muy grande, el volumen de los datos podrían superar los recursos computacionales de procesamiento. En caso contrario si la escala es tan pequeña, la base de datos no podría contener los detalles necesarios y eso no permitirá realizar un mejor análisis de forma fiable al no existir materiales de información específicos.

Todos los mapas tendrán un indicador de la escala del mapa. Un mapa que no se ajusta a una escala específica será indicado por las palabras “no a escala”. Esta notación se encuentra más comúnmente en los mapas gráfico del estilo tales como el “estamos aquí.” “cómo llegar” mapas de estilo utilizadas en las invitaciones. Desde SIG se basa en un umbral mínimo de exactitud y precisión, todos los mapas GIS basada tendrán una escala. Existen tres maneras para mostrar la escala de los mapas: gráfica, verbal y fracción representativa. Las escalas gráficas son también conocidas como barras de escalas, siguiendo este nombre, muestran la escala gráficamente.

La escala verbal es basada en texto, con la escala que se muestra como un número y el tipo de unidad de medida igual a una unidad de medida especificada en el suelo. La escala de fracción es basada también en texto, pero las unidades no son mostradas.



**Escala gráfica**

1 cm = 100,000 km

**Escala Verbal**

1:100,000

**Escala Fracción Representativa**

Figura 2.16: Notaciones de las escalas

### 2.6.3. Datum

La superficie de referencia o superficie del datum es usualmente una elipsoide de revolución, a pesar que algunas veces es aproximado por una esfera. Todos los sistemas de coordenadas están atados a un datum. Un datum define el punto inicial desde el cual las coordenadas son medidas [54]. Por ejemplo las coordenadas de latitud y longitud son determinados por su distancia del ecuador y del primer meridiano que cruza a través de Greenwich, Inglaterra. El datum más recientemente desarrollado y ampliamente usado es el WGS (World Geodetic System) 1984. Este sirve como el marco de trabajo para la localización por medio de todos los dispositivos GPS de todo el mundo.

### Datum WGS84

El datum WGS 84 (*World Geodetic System 1984*) es también un sistema de referencia global obtenido a partir de observaciones satelitales, sin embargo como marco de referencia no ofrece la disponibilidad de estaciones materializadas para la propagación de coordenadas mediante posicionamiento. El datum WGS 84 es el sistema de referencia que utiliza el GPS, es por ello que en el posicionamiento GPS absoluto, con receptores tipo navegador o con otro tipo de receptor usado de manera autónoma, no se obtienen posiciones ligadas a un marco de referencia materializado y las referencias o “puntos de control” son las órbitas o posiciones predichas de los satélites en WGS 84, que proporcionan los datos transmitidos, por lo tanto las posiciones obtenidas están ligadas también al WGS 84. Sin embargo, las posiciones obtenidas del posicionamiento GPS de manera autónoma, en WGS 84, tienen una incertidumbre de hasta 15 metros.

### 2.6.4. El sistema de coordenadas UTM

Ha sido generalmente aceptado para el desarrollo natural, que cada país elija una proyección de mapa que se acomode mejor para sus fines geodésicos y cartográficos considerando el tamaño y forma del área para que sean representados y las características más deseadas en la representación. En muchos casos las proyecciones básicas fueron modificados para servir a sus propios propósitos. Obviamente ninguna proyección de mapas podría ser universalmente aceptada para una aplicación global.

El sistema de coordenadas UTM (*Universal Transverse Mercator*) (ver Figura 2.17), es un sistema que cubre gran parte del globo terráqueo [10, 56]. El sistema UTM no es una sola proyección de un mapa, esta fue desarrollada con el objetivo de crear aplicaciones globales. En la actualidad UTM es una de la proyecciones más extendidas en todos los ámbitos. Divide a la Tierra en una serie de zonas rectangulares mediante una cuadrícula y se aplica una proyección y unos parámetros geodésicos concretos a cada una de dichas zonas. Con este sistema las coordenadas de un punto no se expresan como coordenada terrestres absolutas, sino mediante las zona correspondiente y las coordenadas relativas ala zona UTM en la que se trabaje.

La cuadrícula resultante de UTM alrededor de la Tierra son un total de 60 husos o zonas numerados de 1 al 60. En latitud cada huso se divide en 20 zonas. Una zona UTM se localiza, con un número y una letra. Estas coordenadas se expresan en metros y expresan la distancia entre el punto y el origen de la zona UTM en concreto.

Una posición en la Tierra está dada por el número de zona UTM y el par de coordenadas del este y de la zona norte. El punto de origen de cada zona UTM es la intersección del ecuador y el meridiano central de la zona. Para evitar tratar con números negativos, el meridiano central de cada zona se define, para que coincida con 500000 metros Este.

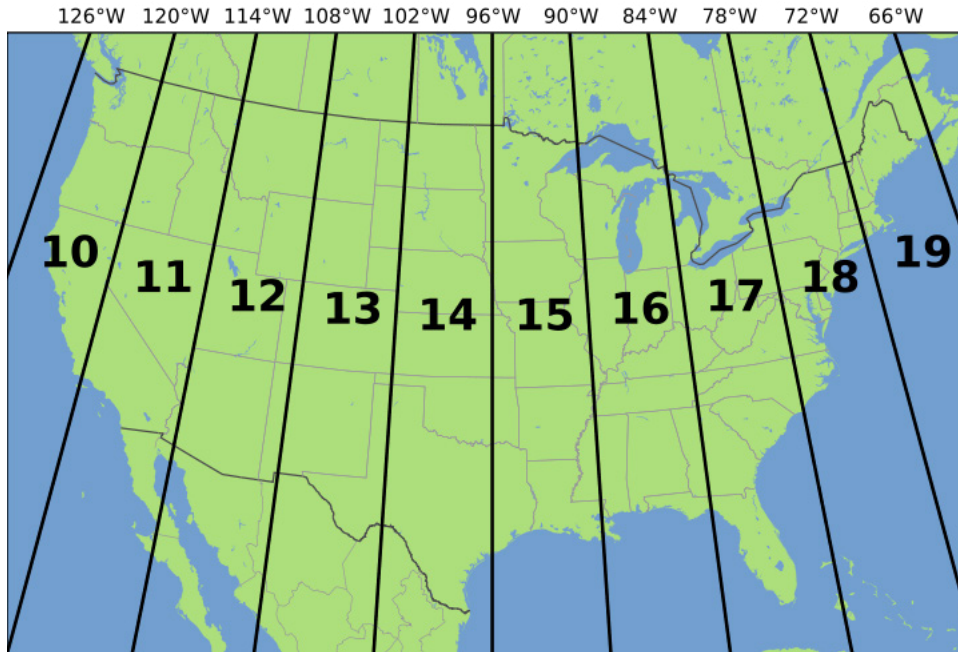


Figura 2.17: Zonas UTM de Estados Unidos de América [10].

En cualquier zona, un punto tiene un este de 400000 metros, el cual está a unos 100 km al oeste del meridiano central.

El sistema de referencia espacial WGS 84 describe la Tierra como un esferoide a lo largo del eje norte-sur con un radio ecuatorial de.  $a = 6378,137 \text{ km}$  y un aplamamiento inverso de  $\frac{1}{f} = 298,257, 223, 563$ . Tomemos un punto de latitud  $\varphi$  y otro de longitud  $\lambda$  y calculemos sus coordenadas UTM como también el factor de escala puntual  $k$ , la convergencia meridiana  $\gamma$ , la cual usa un meridiano de referencia de longitud  $\lambda_0$ . Por convención, en el hemisferio norte  $N_0 = 0 \text{ km}$  y en el hemisferio sur  $N_0 = 10000 \text{ km}$ . Y también por convención  $k_0 = 0,9996$  y  $E_0 = 500 \text{ km}$ .

En las siguientes formulas de acuerdo con [56], las distancia son en kilómetros. De antemano se calcula los siguientes valores preliminares:

$$n = \frac{f}{2 - f}, \quad A = \frac{a}{1 + n} \left( 1 + \frac{n^2}{4} + \frac{n^4}{64} + \dots \right),$$

$$\alpha_1 = \frac{1}{n}n - \frac{2}{3}n^2 + \frac{5}{16}n^3, \quad \alpha_2 = \frac{13}{48}n^2 - \frac{3}{5}n^3, \quad \alpha_3 = \frac{61}{240}n^3,$$

$$\beta_1 = \frac{1}{2}n - \frac{2}{3}n^2 + \frac{37}{96}n^3, \quad \beta_2 = \frac{1}{48}n^2 + \frac{1}{15}n^3, \quad \beta_3 = \frac{17}{480}n^3,$$

$$\delta_1 = 2n - \frac{2}{3}n^2 - 2n^3, \quad \delta_2 = \frac{7}{3}n^2 - \frac{8}{5}n^3, \quad \delta_3 = \frac{56}{15}n^3.$$

Las formulas para convertir coordenadas en latitud y longitud  $(\varphi, \lambda)$  a coordenadas UTM (E,N)

Las formulas finales son:

$$t = \sinh(\tanh^{-1} \sin \varphi - \frac{2\sqrt{n}}{1+n} \tanh^{-1}(\frac{2\sqrt{n}}{1+n} \sin \varphi)),$$

$$\xi' = \tan^{-1}\left(\frac{t}{\cos(\lambda - \lambda_0)}\right), \eta' = \tanh^{-1}\left(\frac{\sin(\lambda - \lambda_0)}{\sqrt{1+t^2}}\right),$$

$$\sigma = 1 + \sum_{j=1}^3 2j\alpha_j \cos(2j\xi') \cosh(2j\eta'), \tau' = \tanh^{-1}\left(\frac{\sin(\lambda - \lambda_0)}{\sqrt{1+t^2}}\right)$$

$$t = \sinh(\varphi)(\tanh^{-1}\left(\frac{2\sqrt{n}}{1+n} \sin(\xi)\right))$$

Las formulas finales son:

$$E = E_0 + k_0 A (\eta' + \sum_{j=1}^3 \alpha_j \cos(2j\xi') \sinh(2j\eta')),$$

$$N = N_0 + k_0 A (\xi' + \sum_{j=1}^3 \alpha_j \sin(2j\xi') \cosh(2j\eta')),$$

$$k = \frac{k_0 A}{a} \sqrt{\left\{1 + \left(\frac{1-n}{1+n} \tan \varphi\right)^2\right\} \frac{\sigma^2 + \tau^2}{t^2 + \cos^2(\lambda - \lambda_0)}},$$

$$\gamma = \tan^{-1}\left(\frac{\tau\sqrt{1+t^2} + \sigma t \tan(\lambda - \lambda_0)}{\sigma\sqrt{1+t^2} - \tau t \tan(\lambda - \lambda_0)}\right)$$

Para coordenadas UTM (E,N,Zona, Heml) que sean convertidas a latitud y longitud  $(\varphi, \lambda)$ , los valores para el norte de  $Hemi = +1$  y  $Hemi = -1$  para el sur. Primero se calcula los valores intermedios:

$$\xi = \frac{N - N_0}{k_0 A}, \eta = \frac{E - E_0}{k_0 A},$$

$$\xi' = \xi - \sum_{j=1}^3 \beta_j \sin(2j\xi) \cosh(2j\eta), \eta' = \eta - \sum_{j=1}^3 \beta_j \cos(2j\xi) \sinh(2j\eta),$$

$$\sigma' = 1 - \sum_{j=1}^3 2j\beta_j \cos(2j\xi) \cosh(2j\eta), \tau' = \sum_{j=1}^3 2j\beta_j \sin(2j\xi) \sinh(2j\eta),$$

$$\chi = \sin^{-1}\left(\frac{\sin \xi'}{\cosh \eta'}\right).$$

Las formulas finales son:

$$\varphi = \chi + \sum_{j=1}^3 \delta_j \sin(2j\chi)$$

$$\lambda_0 = Zona \times 6^\circ - 183^\circ$$

$$\lambda = \lambda_0 + \tan^{-1}\left(\frac{\sinh \eta'}{\cos \xi'}\right),$$

$$k = \frac{k_0 A}{a} \sqrt{\left\{1 + \left(\frac{1-n}{1+n} \tan \varphi\right)^2\right\} \frac{\cos^2 \xi' + \sinh^2 \eta'}{\sigma'^2 + \tau'^2}}$$

$$\gamma = Hemi \times \tan^{-1}\left(\frac{\tau' + \sigma \tan \xi' \tanh \eta'}{\sigma' - \tau' \tan \xi' \tanh \eta'}\right)$$

Con las formulas anteriormente descritas se transforman coordenadas UTM a coordenadas WGS 84, esto con el fin de poder cambiar la localización de la información geográfica y trabajar con esta en diferentes proyecciones dependiendo de la área geográfica de estudio, la cual depende si esta en el hemisferio sur o norte. El sistema WGS 84 es el elegido por ser un estándar par América del Norte y el cual es ampliamente utilizado por herramientas o dispositivos.

## 2.7. SIG-Web

La Web o la *World Wide Web* es un espacio de información en el cual los objetos de interés, referidos como recursos, están identificados por identificadores globales llamados *Uniform Resource Identifiers* (URI) [57] o es definido también como un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet. Fue creada por Tim Berbers-Lee durante el periodo de 1989 a 1991, con el objetivo de encontrar una manera más fácil de compartir e intercambiar documentos con sus colegas. Berners-Lee invento HTTP(Hypertext Transfer Protocol), HTML(*Hypertext Markup Language*)y la URL(*Uniform Resource Locator*). Berners-Lee es considerado como el “padre de la Web”.

### 2.7.1. Web 2.0

El concepto de la Web 2.0 empezó con una conferencia de intercambio de ideas entre O'Reilly y MediaLive International. Dale Dougherty, pionero de la Web y O'Reilly VP, observaron que lejos de haberse “estrellado”, la Web era más importante que nunca, con nuevas e interesantes aplicaciones y sitios apareciendo con sorprendente regularidad. Lo que es más, las empresas que habían sobrevivido al colapso parecía tener algunas cosas en común. Podría ser que el colapso de las punto-com marcó una especie de punto de inflexión para la Web, de tal manera que una llamada a la acción como “Web 2.0”. Estamos de acuerdo en que lo hizo y así nació la Web 2.0 [58].

La Web 2.0 es la red como plataforma, que abarca todos los dispositivos conectados. Las aplicaciones Web 2.0 son las que realizan la mayoría de las ventajas intrínsecas de esa plataforma: la entrega de software como servicio continuamente actualizada que mejora el mayor número de personas que lo utilizan, consumiendo y mezclando datos de múltiples fuentes, incluyendo usuarios individuales, al tiempo que proporciona sus propio datos y servicios en una forma que permite mezclar por otros, la creación de efectos de red a través de una “arquitectura de participación”, que van más allá de la metáfora de la página de la Web 1.0 para ofrecer experiencias interactivas.

Las tecnologías de los navegadores Web del lado del cliente, utilizados en el desarrollo de la Web 2.0 son Asynchronous JavaScript y XML (AJAX), Adobe Flash y el framework Adobe Flex y frameworks JavaScript / Ajax como Yahoo!Library VI, ExtJs y jQuery.

### 2.7.2. Fundamentos Web

Una aplicación Web básica es esencialmente una arquitectura cliente-servidor (C/S), en la cual el cliente es específicamente llamado cliente Web y el servidor es llamado servidor Web. La aplicaciones Web usualmente tienen una arquitectura de tres capas las cuales son la capa de datos, capa lógica y la capa de presentación (ver Figura 2.18) [58]. El cliente Web es típicamente un navegador Web, la arquitectura Web básica es también referida como la navegador/servidor (N/S).

Las tres tecnologías principales de la *World Wide Web*(WWW) son:

- HTTP :Hypertext Transfer Protocol.
- URL : Uniform Resource Locator.
- HTML : Hypertext Markup Language.

Estos protocolos fueron inventados por Tim Berners-Lee en 1990 [3].Estas tecnologías son ahora estándares internacionales y sus especificaciones son manejadas y mantenidas por W3C(World Wide Web Consortium).



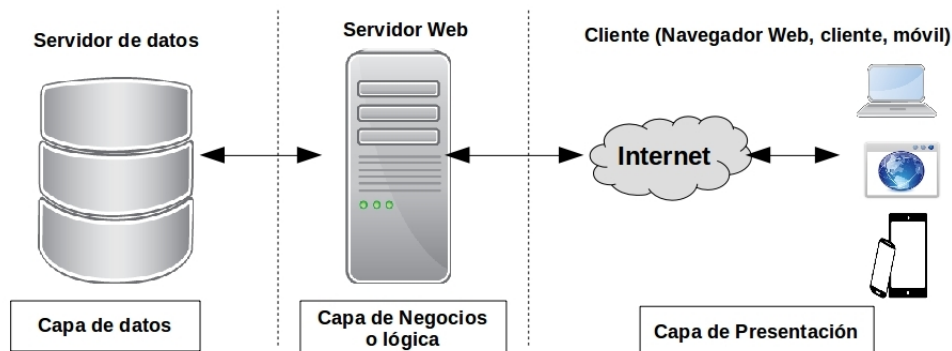


Figura 2.18: Arquitectura básica de una aplicación Web incluye tres capas.

## HTTP

La Web necesita de un protocolo que define las reglas y procedimientos para todas las partes lo sigan. Como un protocolo, HTTP (*Hypertext Transfer Protocol*) define un conjunto de reglas y procedimiento que clientes Web y servidores Web utilizan para comunicarse entre sí.

HTTP define ocho métodos de petición *GET*, *POST*, *HEAD*, *PUT*, *DELETE*, *TRACE*, *OPTIONS* y *CONNECT*. *GET* y *POST* son dos de las peticiones más comúnmente utilizadas para peticiones. Las peticiones HTTP y los mensajes de respuesta tienen encabezados y cuerpos.

## URL

Una URL (*Uniform Resource Locator*) es un subconjunto de una URI (*Uniform Resource Identifier*) que especifica donde un recurso identificado está disponible en la Internet y el mecanismo por el cual es recuperado. Una URL es a menudo llamada dirección Web. Cada página Web tiene una única URL. Justo como una dirección normal es utilizada para localizar casas en el mundo real, una dirección Web es utilizada para localizar una página Web sobre la WWW, la cual contiene miles de millones de páginas Web.

## HTML

HTML (*Hypertext Markup Language*) es el principal lenguaje para la creación de páginas Web y es el principal recurso para las páginas Web. HTML contiene contenidos, un diseño e información de formato. Cuando una página Web es cargada en un navegador Web, el navegador Web interpreta el código HTML y muestra la página Web de la manera en que el diseñador deseaba ser vista.

## Servidor Web

Los servidores Web hospedan los sitios Web que se leen y sirven datos a otras aplicaciones Web. Son los responsables por aceptar peticiones HTTP de los cliente y entregarles respuestas HTTP. Los servidores Web son conocidos por comunicar con los clientes. Los servidores también son contenedores que permite *scripts* tales como JSP(JavaServer Pages), JavaServlet y ASP.NET para ejecutarse internamente.

### 2.7.3. Sistemas de Información Geográfica Web

La Web cambio todo, los SIG no son la excepción. Los SIG Web son el resultado de la combinación de la *Web* y de los sistemas geográficos de información, los SIG han crecido hasta convertirse en una disciplina en rápido desarrollo desde su creación en 1993. SIG se ha convertido en una aplicación convincente de Internet que ha llevado a muchas personas a tomar ventaja de la *Web*. La gran mayoría de los usuarios de Internet utilizan mapas simples o otras aplicaciones habilitadas espacialmente a través de Internet en algún momento, aunque muchos no son conscientes de esto.

En 1993 el *Xerox Corporation Palo Alto Research Center*(PARC) desarrolla el primer visor de mapas Web, naciendo los SIG Web. El visor de mapas de Xerox PARC fue un experimento para permitir la recuperación de información interactiva sobre la Web. El sitio Web proveía de un mapa simple con funciones de ampliación, selección de capas y conversión de proyecciones.

La distribución de la información geoespacial en Internet es un factor de ejecución por los proveedores de información. Internet permite a todos los niveles de la sociedad acceder a la información geoespacial y proveer un medio para el procesamiento de información georeferenciada sin ninguna restricción de ubicación. Los SIG basados en la Web evolucionaron a partir de diferentes mapas Web y arquitecturas cliente-servidor para distribuirlos. Como tal, Internet cambia la forma de todas las funciones de los sistemas de información que incluyen: la recopilación, almacenamiento, recuperación, análisis y visualización de datos.

Un SIG Web o SIG basado en Web es un tipo de sistema de información distribuido. La forma más simple de un SIG Web debería tener al menos un servidor y un cliente, donde el servidor es un servidor de aplicaciones Web y el cliente es un navegador Web, una aplicación de escritorio, o una aplicación móvil (ver Figura 2.19). EL servidor tiene una URL para que los clientes puedan encontrar lo en la Web. Luego, el cliente se basa en las especificaciones HTTP para enviar solicitudes al servidor. El servidor realiza la operación que el SIG ha solicitado y envía una respuesta al cliente, de nuevo a través de HTTP. El formato de la respuesta es un archivo HTML que es utilizado por el cliente del explorador

Web, pero también están otros formatos como imagen binaria, XML (Extensible Markup Language), o JSON (JavaScript Object Notation).

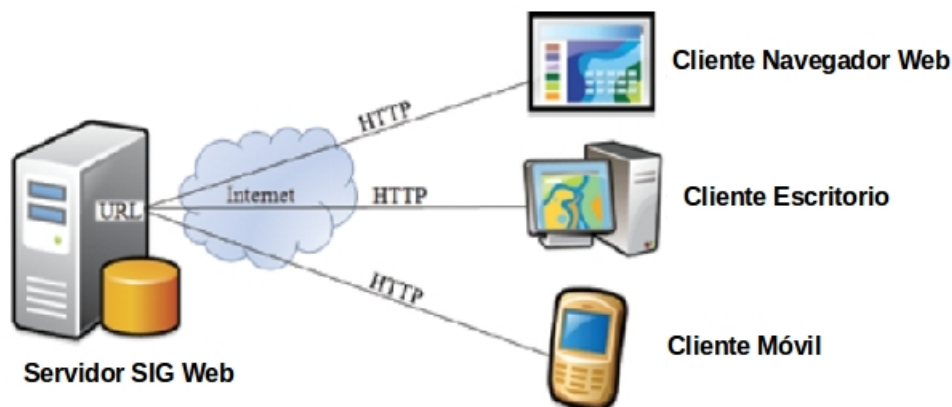


Figura 2.19: Modelo Típico de un SIG Web

El desarrollo de la Web y la expansión del Internet proveen dos capacidades claves que ayudan enormemente a los geocientíficos. Primero la Web permite la interacción visual con los datos. Segundo, gracias a la naturaleza del internet, los datos geoespaciales son ampliamente accesibles.

Los SIG Web no están libres de fallas. El problema principal es la velocidad, los SIG generalmente hacen uso extensivo de los gráficos. Las velocidades de la conexión al Internet hacen un uso intensivo de gráficos y hacer la conexión más lenta para los usuarios.

Un SIG Web extiende una aplicación Web básica con capacidades de los SIG. La arquitectura básica de los SIG Web son similares a cualquier aplicación Web, pero añadiendo componentes de los SIG. En un flujo de trabajo básico, un usuario usa una aplicación de SIG Web a través de un cliente, el cual es una navegador Web, una aplicación de escritorio o una aplicación móvil. El cliente envía una petición a el servidor Web sobre Internet a través HTTP. El servidor Web envía peticiones relacionadas del SIG a el servidor SIG. El servidor SIG regresa los datos necesitados de la base de datos SIG y procesa la petición, la cual genera un mapa, realizar una consulta, o realizar un análisis. La información, mapas o otros resultados son enviados por el servidor Web a el cliente en respuesta, por medio de HTTP. El cliente muestra el resultado a el usuario, lo cual completa el ciclo de petición y respuesta. HTML como un lenguaje de marcas de hipertexto es esencialmente texto plano marcado por un conjunto de etiquetas que son definidos en hojas de estilo(CSS), un lenguaje de tipo *script* es usado para dar estilo a las paginas Web.

### Características de un SIG-Web

El Internet y la Web eliminaron la restricción de la distancia desde el ciberespacio, lo que permite el acceso instantáneo a la información a través de Internet sin tener en cuenta lo lejos que el usuario y el servidor podría estar el uno del otro. Esta cualidad de los SIG-Web dan ventajas inherentes sobre los SIG de escritorio tradicional, incluyendo los siguientes:

- **Alcance global:** Un usuario accede a aplicaciones SIG-Web desde su hogar por medio de un celular o su computadora.
- **Gran número de usuarios:** Los SIG-Web son usados por docenas o cientos de usuarios de manera simultánea. Ésto requiere mejor desempeño y escalabilidad para los SIG-Web que para los SIG de escritorio
- **Mejor Capacidad cross-platform:** La mayoría de los clientes son navegadores Web. Existen diversos navegadores Web para diversos sistemas operativos. Los SIG Web soportan diferentes sistemas operativos.
- **Menor costo promedio por número de usuarios:** El SIG es compartido a diversos usuarios, sin tener que comprar o instalar la aplicación.
- **Uso fácil para los usuarios finales:** Los SIG Web son diseñados para simplicidad, intuición y conveniencia, haciendo los típicamente mucho más fáciles para los usuarios finales que un SIG de escritorio.
- **Actualización unificada:** Los SIG de escritorio necesitan ser actualizados e instalados en cualquier computadora, Pero los SIG Web con una sola actualización es suficiente para que todos los usuarios posean una versión actualizada.
- **Aplicaciones Diversas:** Las aplicaciones son de ámbito empresarial como públicos.

### Funciones de un SIG Web

Los SIG Web son capaces de realizar todas las funciones que implican la información espacial de los SIG, que incluyen capturar, almacenar, editar, manipular, manejar, análisis, compartir y visualizar. Algunos de las fortalezas de los SIG Web incluyen lo siguiente:

- **Visualización y consulta.** La cartografía Web, como la cara de los SIG Web, es la función más usada. Los datos de los SIG y el análisis de resultados son usualmente presentados como mapas.
- **Coleccionar información espacial.** Los SIG Web deben de poder obtener datos de BD, o mediante otras fuentes por medio de Servicios Web.

- **Diseminación.** Los SIG Web son sistemas distribuidos y estos están en múltiples equipos (e.g. computadoras personales, celulares, tabletas, entre otros) para compartir información geográfica.
- **Análisis geoespacial.** A partir de información geográfica por medio de su análisis se puede determinar caminos mas cortos, peligros potenciales, zonas con inundaciones, donde crear una nueva tienda.

### Mapas Web Interactivos

Existen varios niveles de tecnologías para publicar datos de los mapas en la Web, que van desde los sitios que simplemente publican mapas Web estáticos, a sitios más sofisticados que soportan mapas dinámicos, mapas personalizados de forma interactiva y múltiples plataformas de computadoras y sistemas operativos. En términos de los SIG Web, el desafío más grande es la interacción. La Open GIS Consortium un grupo de interés especial por los mapas Web están trabajando sobre los problemas de la publicación de SIGs basados en Web [59]. Ellos desarrollaron un modelo esencial de representación interactiva.

Este modelo es una herramienta muy útil para analizar y comparar diferentes arquitecturas para los Servidores de Mapas en Internet y otras aplicaciones SIG de Internet. El modelo de representación interactiva tiene cuatro niveles:

- El proceso de selección recupera información de una fuente de datos geoespacial de acuerdo a las restricciones de la consulta tales como una área de búsqueda o selecciones temáticas.
- El *Display Element Generator* procesa turnos de los datos geoespaciales seleccionados en una secuencia de elementos mostrados. Se usan estilos tales como símbolos, estilos de línea, estilos de relleno con el ámbito espacial genera la anotación de atributos alfanuméricos, ordena los elementos de la pantalla en un cierto orden y hace otro procesamiento gráfico.
- El *Render* toma los elementos mostrados y genera un mapa renderizado. Ejemplos de mapas renderizados son las listas de visualización *In-memory* (En memoria).
- El proceso *Display* hace al mapa mostrado visible para el usuario de un catálogo de disposición.

### Servicios Web Geoespaciales

La tecnología de los servicios Web ha tenido un efecto dramático en la evolución de los productos SIG Web, permitiendo colocar un nicho de productos tales como productos de servidor, geonavegadores y portales Web que correspondan a funciones de servidor, cliente e intermediario en una arquitectura Web. Los servicios Web se están incrementando

solo con la construcción de bloques en una arquitectura orientada al servicios (SOA) o en un arquitectura orientada a Web (WOA), los cuales permiten a los SIG ser integrado principalmente en los sistemas de información. El uso de los servicios Web proveen un nuevo enfoque a la interoperabilidad. La tecnología de los servicios Web tienen una gran importancia hoy y en el futuro.

“Un servicio Web es un programa que se ejecuta sobre un servidor Web y expone interfaces de programación a otros programas en la Web” [3]. Los servicios Web representan una evolución importante para la computación distribuida. Esta reemplaza una función local con una función que es provista remotamente por un servidor.

Con base en los servicios Web se crearon los servicios Web geoespaciales, los cuales permiten fácilmente proveer información o servicios de mapas, poder conseguir herramientas geoespaciales, catalogar la información de cierta región.

### **Tipos de Servicios Web**

La comunicación entre el servicios Web y el cliente envuelve el envío de peticiones del cliente a el servicio Web y el regreso de respuestas del servicio Web a el cliente. Los servicios Web se categorizan en dos tipos principales, servicios Web basados en SOAP o en servicios Web *REST-style*.

### **Servicios Web basados en SOAP**

SOAP es un protocolo de especificación para el intercambio de información estructurada en formato XML. SOAP llego a ser una recomendación de la W3C en 2003, pero el acrónimo “Simple Object Access Protocol” fue considerado erróneo y desechado por la W3C en 2007. Los servicios Web basados en SOAP usan *HTTP Post* para enviar las peticiones. Las peticiones y respuestas están en encapsulados XML SOAP (ver Figura 2.20).

### **Servicios Web basados en el estilo REST**

REST (*Representational State Transfer*) es un estilo de arquitectura de software introducida en el 2000 por Roy Fielding. Este estilo de arquitectura es diseñado para tomar ventaja de HTTP, para reducir la complejidad de sistemas y mejorar la escalabilidad del sistema. Los servicios Web de estilo REST son servicios Web que transmiten información sobre HTTP sin una capa adicional de mensajería como SOAP. Lo más común en una arquitectura de servicios Web REST, es que los clientes envían todos los parámetros en la URL solicitado. Es decir todas las peticiones son realizadas por un URL y todos los parámetros están en el URL. REST utiliza comúnmente estándares como JSON y XML (ver Figura 2.21).

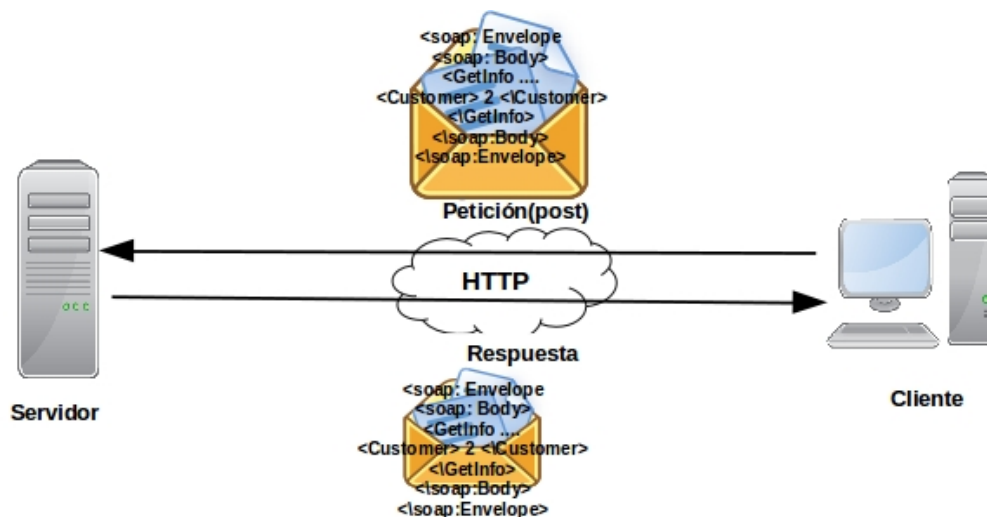


Figura 2.20: Servicios Web basados en SOAP.

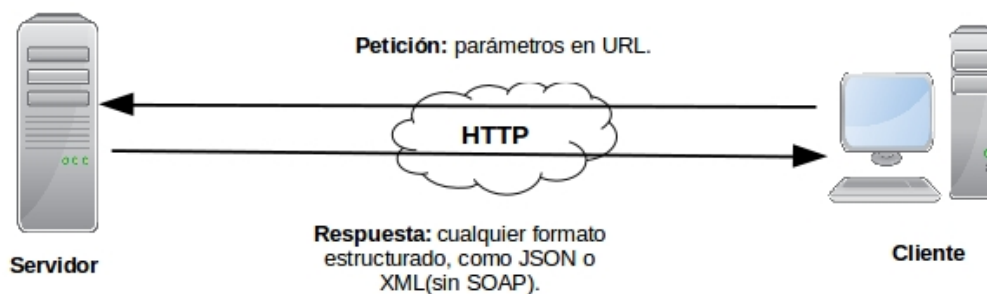


Figura 2.21: Servicios Web basados en REST.

## Estándares de Servicios Web Geoespaciales

### WMS

WMS (Web Map Service) es un estándar OGC e ISO para la petición y servicio de mapas sobre Internet. Los mapas producidos por WMS son generalmente renderizados por un formato de imágenes como PNG, GIF o JPEG. Las operaciones que maneja WMS son:

- GetCapabilities, pide los servicios de metadatos desde el servidor. Los metadatos se encuentran en un formato XML.
- GetMap, Pide un mapa del servidor, incluye parámetros como capa geográficas, áreas de interés, coordenadas y formato.
- GetFeatureInfo, recupera información para una ubicación sobre un mapa.

## WFS

WFS(Web Feature Service) es un estándar de la OGC para ofrecer servicios Web para la lectura y escritura de características geográficas. Con, WFS los clientes realizan operaciones, tales como inserción, eliminación y consulta de información geográfica que se encuentre en un servidor. WFS define las siguientes operaciones:

- GetCapabilities, solicita el servicios de metadatos.
- DescribeFeatureType, solicita la estructura del tipo de características.
- GetFeature, recupera una característica geográfica y sus atributos para realizar una consulta filtrada.
- LockFeature, solicita al servidor bloquear una o más característica durante una transacción.
- Transaction, solicita a el servidor, crear, actualizar y eliminar características geográficas.

Este estandar tiene dos principales clases son definidas:

- WFS Básico, implementa las funciones GetCapabilities, DescribeFeatureType y GetFeature. Este es considerado como WFS de solo lectura.
- WFS Transaccional(WFS-T), implementa todas las capacidades de WFS básico, más la operación de transacción. Es considerado como un WFS de lectura y escritura.

## WCS

WCS(Web Coverage Service) soporta la recuperación de datos geoespaciales como “*coverages*(coberturas)”, incluyendo imágenes satelitales, fotos aéreas, datos de elevación digital, entre otros. WCS regresa información sin tratar y es utilizado para información de tipo raster. Sus funciones son:

- GetCapabilities, pide el servicio de metadatos XML.
- DescribeCoverage, pide todas las descripciones de una o más coberturas servidas por un servidor WCS.
- GetCoverage, solicita una cobertura con la área geográfica seleccionada, dentro de un rango de tiempo, en un sistema de coordenadas y en un formato como GeoTIFF, HDF-EOS y NITF.



## 2.8. Arquitectura y componentes básicos de un SIG Web

Los SIG Web son similares a las arquitecturas típicas de cliente/servidor de tres niveles. El geoprocesamiento se descompone en tareas del lado del servidor y lado del cliente. Un cliente típico es un navegador Web. El lado del servidor consiste de un servidor Web, software SIG Web y bases de datos (ver Figura 2.22).

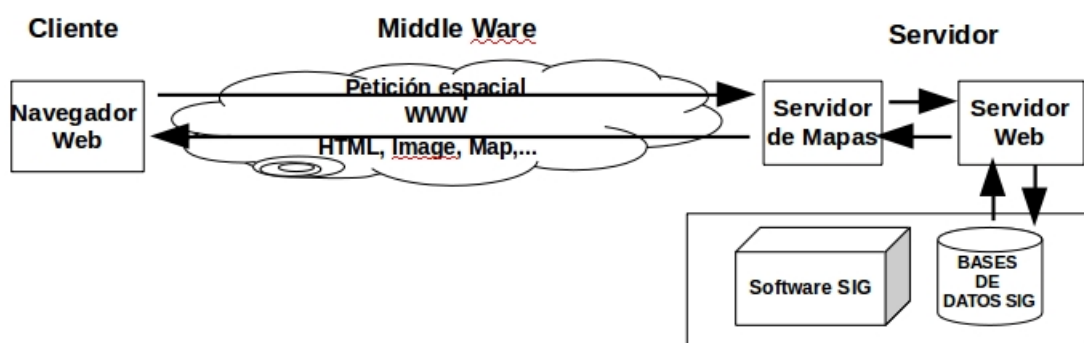


Figura 2.22: Modelo Típico de un SIG Web

Este modelo de red existe ampliamente dentro de las empresas, en las que algunos ordenadores actúan como servidores y otros actúan como clientes. El Servidor simplemente tiene el SIG en funcionamiento y añade una interfaz de cliente en el lado del cliente y un middleware en el lado del servidor para la comunicación entre el cliente y el software GIS. El desarrollo en la programación orientada a objetos hace posible producir componentes de software y enviarlos al cliente antes de ejecutarse en la máquina del cliente. Ésto sale al thick. La arquitectura del cliente thick permite a la computadora del cliente realiza la mayoría del procesamiento localmente. Los SIG Web están conformados por los siguientes componentes básicos: servidores Web, bases de datos SIG, clientes de un SIG Web, servidor de mapas.

### Servidor SIG Web

El servidor SIG Web es el componente más importante en un SIG Web. Su funcionalidad, capacidad, para ser personalizado, escalable y rendimiento son fundamentales para el éxito de la aplicación SIG Web. La capacidad y calidad de una aplicación SIG Web son determinadas por el servidor SIG Web que se utilice.

### Base de datos SIG

Las bases de datos SIG son el *framework* para almacenar información y manejar un SIG Web. Estas soportan una colección de conjuntos de datos geográficos de varios tipos, tales

como datos vectoriales (puntos, líneas, polígonos) y datos raster (imágenes satelitales e imágenes aéreas). Las bases de datos SIG son el soporte subyacente para las aplicaciones SIG Web. Las respuestas son enviadas desde una aplicación SIG Web, la cual sólo es tan buena como la calidad de la información contenida en la base de datos GIS. Aunque la aplicación casual confía en fuentes de datos ocasionales, las aplicaciones profesionales tienden a necesitar información geográfica de alta calidad, autorizada y actualizada.

## Cliente de un SIG Web

El cliente en una aplicación Web desempeña dos roles. Primero representa la interfaz de usuario final para el sistema entero. Este interactúa con el usuario, colecta todas las entradas del usuario, envía las peticiones a el servidor y presenta los resultados al usuario. Lo que necesita conocer el usuario acerca del sistema es que tan rápido, estable, confiable y que tan amigable es al cliente. Segundo, el cliente especialmente un cliente *thick*, realiza algunas tareas de procesamiento geoespacial, como clasificación dinámica para el análisis de mapas temático, de agrupación y mapas de calor. Los clientes SIG Web son típicamente navegadores Web pero que también son aplicaciones escritorio, aplicaciones móviles, o incluso aplicaciones del servidor.

## Cliente del navegador Web

Los navegadores son el principal tipo de cliente del navegador Web. Los navegadores Web actualmente no son estáticos y con tecnologías como AJAX, Flex, Silverlight y JavaFX crean interfaces dinámicas, ricas y amigables conocidas como RIA. Si se utilizan diferentes APIs en el desarrollo de aplicaciones, realizan diferentes funciones SIG.

Existen diferentes tipos de arquitecturas para el desarrollo de un SIG Web, las dos principales son la arquitectura *Thin Client* y *Thick Client*. Estas arquitecturas fueron propuestas dependiendo de la consideración del diseño que deba tener la aplicación, la carga de trabajo, el diseño de l sistema cliente/servidor, si la carga de trabajo es distribuida.

## Servidor de mapas

Los servidores de mapas o servidores de cartografía digital (IMS) permiten que las bases de datos SIG sean fácilmente accesibles a través de una interfaz del navegador. Éstos proveen información vectorial e imágenes georeferenciadas. Usualmente los servidores de mapas trabajan con estándares de la OGC.

### 2.8.1. Arquitectura *Thin Client*

La arquitectura *Thin Client* se utiliza en la arquitectura típica. En un sistema *Thin Client*, los cliente solo tienen interfaces de usuario para comunicarse con el servidor y mostrar resultados. Todo el procesamiento es realizado sobre el servidor como es mostrado en la Figura 2.23. Los servidores usualmente tienen más poder que la del cliente y manejan los recursos centralizados [59]. Además, la principal funcionalidad es sobre el lado del servidor en la arquitectura thin hay también la posibilidad para programas de utilidad en el lado del servidor para ser enlazadas en el software del servidor.

La figura 2.23 muestra la comunicación esquemática entre el navegador Web, el servidor Web y el servidor SIG. Sobre el lado del servidor Web, existen algunas posibilidades para realizar la conexión del SIG a la World Wide Web o Web. Como CGI, Web Server Application Programming Interface (API), Active Server Pages, Java Server Pages (JSP) Y Java Servlet.

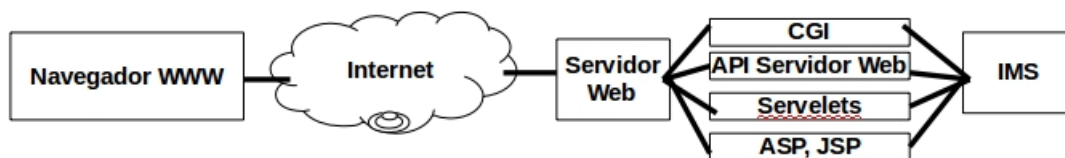


Figura 2.23: Aplicaciones del lado del Servidor

### 2.8.2. Arquitectura *Thick Client*

En general un navegador Web maneja documentos HTML e imágenes raster embebidas en formatos estándar. Para manejar otros tipos de formatos como datos vectoriales, vídeos o archivos de música, la funcionalidad de los navegadores ha sido extendida. Usando la arquitectura thin client, los archivos vectoriales podrían no ser usados. Para superar este problema la mayoría de los navegadores ofrecen un mecanismo que permiten a programas de tercer nivel trabajar juntos con el navegador como un Plug-in [59].

La funcionalidad de las interfaz de usuario ha progresado de la atracción de documentos simples a aplicaciones más atractivas. Este progreso es como sigue: HTML, CGI, usando HTML forms y CGI, Java script para incrementar las capacidades de interfaz de usuario (ver Figura 2.24).

## 2.9. Resumen

En este capítulo se describió el significado de SIG, estos sistemas de información ofrecen datos geográficos. Los SIG esta compuestos de elementos: hardware, software, personas,

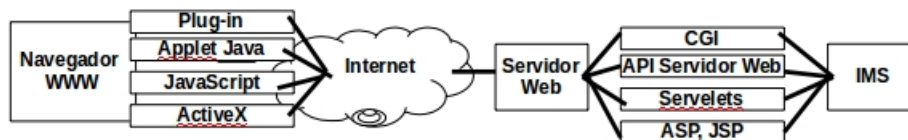


Figura 2.24: Aplicaciones del lado del Cliente

datos y métodos. Siendo los datos la parte más importante de los SIG. Los datos son almacenados en imágenes en los formatos vectoriales o raster, o otros sistemas optan por una BDE para guardar la información geográfica. Los datos manejan diferentes tipos de proyecciones dependiendo del país o región estudiada, estos tienen su propio sistema de coordenadas. Los SIG Web son definidos como la unión de SIG y Web para usar tecnologías Web como HTTP, URL, HTML. Estos sistemas aumentan la funcionalidad de los SIG volviéndolos por naturaleza en sistemas distribuidos.

# Capítulo 3

## Estado del arte

En este capítulo se listaran los trabajos relacionados a este proyecto de tesis, los trabajos están enfocados en el área de Sistemas de Información Geográfica Web (SIG Web) y de la geovisualización. Estos trabajos proponen diferentes arquitecturas para SIG's-Web, la mayoría de estas en herramientas de libre distribución para crear aplicaciones SIG-Web. Algunos trabajos estudiados tratan sobre la geovisualización de elementos vectoriales y raster en una pared de vídeo.

### **3.1. MapBeing: Una arquitectura para la manipulación y publicación de datos vectoriales en un SIG basado en Web**

MapBeing [60] es una arquitectura propuesta para la manipulación y publicación de datos vectoriales, la cual provee de características para manipular y publicar inmensos volúmenes de datos, sin perder desempeño. Proyectos anteriores como TerraFly [61] y GeoHosting [62] se enfocan únicamente a la manipulación y el análisis de datos vectoriales. Ninguna contribución anterior considera la manipulación de datos vectoriales. El principal objetivo de esta arquitectura es la manipulación de datos vectoriales, lo cual es muy difícil para los usuarios editar o eliminar formas geoespaciales en los dispositivos cliente, ya que sólo se proporcionan las imágenes parciales de los datos totales del vector.

La arquitectura de MapBeing ésta dividida en tres secciones y una base de datos:

- Navegador Web.
- Servidor MapBeing.
- Geoserver.

- Base de datos PostGIS.

EL navegador Web tiene la tarea de responder a la interacción del usuario sobre el mapa. La tarea es completada por la biblioteca OpenLayers. OpenLayers recibe las peticiones del usuario y las envía a el servidor MapBeing. Otra vez la respuesta a el servidor MapBeing es mostrada a el usuario. El servidor MapBeing es utilizado para la manipulación de los datos vectoriales.

La sección del servidor MapBeing es la responsable en la conexión de OpenLayers con GeoServer. Como un servidor Web, tiene la tarea convencional de proveer contenido Web a el navegador a través de internet; y procesar las instrucciones del servidor y enviando una respuesta apropiada a la petición del usuario. Sin embargo la principal responsabilidad del servidor MapBeing aquí es para procesar las peticiones de manipulación de usuario y enviar estas a GeoServer. GeoServer es la parte más importante de este *framework*. Éste provee tres servicios, los cuales pueden ser revisados en la SubSección 2.7.3:

- Web Feature Service (WFS).
- Web Map Service (WMS).
- Web Coverage Service (WCS).

WFS provee una interfaz para realizar peticiones y obtener características geográficas. WFS no solo permite realizar consultas y obtener características, también se cuenta con un WFS transaccional (WFS-T), éste permite crear, eliminar y actualizar características. WFS es un protocolo estandarizado que sirve imágenes de mapas geoferenciadas en el Internet. El navegador Web se comunica directamente con GeoServer a través del servicio WMS. Las imágenes obtenidas de este servicio son mostradas en el navegador. No se puede manipular imágenes, a causa de esto se utilizaron el servicio WFS-T, para permitir la manipulación directa con los datos vectoriales.

La arquitectura MapBeing (ver Figura 3.1) ésta separada en tres capas:

- Capa User End.
- Capa Request Handler.
- Capa Service .
- Data provider.

La capa **User End** la cual es representada por OpenLayers, interactúa con los usuarios que visualizan ubicaciones en el mapa. Es responsable de tomar las instrucciones de usuario y mostrar los respectivos cambios en el mapa. La capa **Request Handler** procesa las demandas y peticiones de la capa **Service** para solicitar los servicios requeridos. La

### 3.1. MAPBEING: UNA ARQUITECTURA PARA LA MANIPULACIÓN Y PUBLICACIÓN DE DA

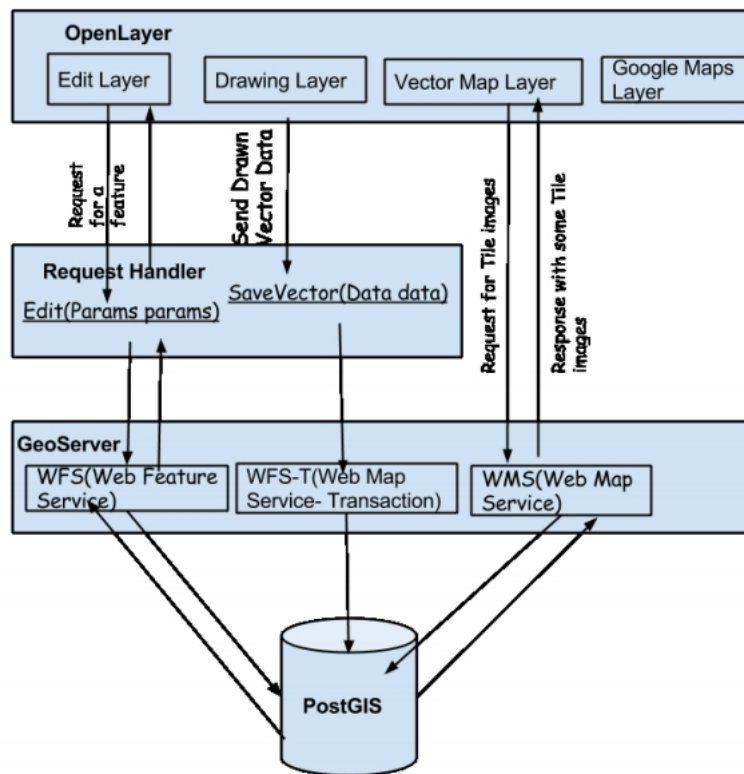


Figura 3.1: Arquitectura de MapBeing

capa **Service** provee los servicios geográficos al usuario final a través de la capa **Data provider**, conocida como capa GeoServer.

La arquitectura propuesta ésta basada en software de libre distribución. MapBeing utilizo OpenLayers como cliente, GeoServer como un servidor Web y PostGIS para el almacenamiento de datos espaciales. MapBeing tiene un servidor para tomar peticiones XML-HTTP de OpenLayers. GeoServer realiza peticiones a PostGIS por datos vectoriales dependiendo de los parámetros de la petición y crea imágenes en mosaicos de los datos vectoriales. MapBeing fue probado en un equipo con un sistema operativo Windows 7, un procesador Intel Core 2 Duo E8400@3.00GHz y 2.00 GB RAM fueron usados como una plataforma experimental. En este equipo se realizaron pruebas, donde MapBeing descarga diferentes archivos de tipo ESRI's *shape*.

## 3.2. Diseño e Implementación de un Modelo, basado en Web, SIG de un atlas de Cáncer

En este trabajo se desarrolló una herramienta de visualización basada en Web para la exploración de datos de cáncer a nivel de condado para el estado de Pensilvania [63], esta herramienta tiene el nombre de *Pennsylvania Cancer Atlas* (PA-CA). El público de este SIG incluye profesionales en los departamentos de públicos locales de salud quienes tienen un enfoque particular sobre la vigilancia y control del cáncer.

El sistema PA-CA utiliza un conjunto de datos del Registro de Cáncer en Pensilvania, donde se contiene información de las tasas de incidencia de cáncer colorrectal y de próstata de 1994-2002 estratificada<sup>1</sup> con la edad, el género, la raza y el estadio al momento del diagnóstico.

El atlas creado de este trabajo es un prototipo, el cual fue diseñado como un modelo para la implementación de atlas, que apoyen actividades del gobierno en el control de cáncer. Este modelo integra principios de simbolización y diseño de cartografía impresa, estrategias de interacción de geovisualización exploratoria y avances en servicios de *Web-map/Web-feature* en SIG.

El objetivo de su SIG Web es combinar el potencial de combinar mapas impresos o a papel, para impulsar una visión acerca de las distribuciones espaciales y las relaciones con la capacidad del entorno digital, para apoyar el análisis exploratorio, la generación de hipótesis, comprobación de hipótesis estadística y computacional, la toma de decisiones políticas y la difusión de la información. De forma más específica una integración de los avances en servicios distribuidos basados en mapas SIG Web, atlas interactivos y la geovisualización exploratoria añaden un valor sustancial a los repositorios desarrolladas de forma sistemática de los datos relacionados con cáncer- siendo construidos por las agencias gubernamentales estatales y nacionales.

El modelo del atlas diseñado e implementado usa una estructura modular que provee un *framework* para extender la funcionalidad de los SIG y cartográfica a través del tiempo, incluyendo la capacidad de sustituir un cliente de base de datos, o cualquier servidor diferente sin ninguna modificación sustancial de los otros componentes.

La arquitectura de este SIG (ver Figura 3.2), muestra que el sistema realiza ésta compuesto por un cliente, el cual envía peticiones (*requests*) al servidor de mapas Geoserver, el cual es el intermediario entre el cliente y la base de datos. La base de datos del atlas

---

<sup>1</sup>estratificada



### 3.3. INTEGRACIÓN DEL ENFOQUE AJAX DE SERVICIOS WEB EN SIGS DE VISUALIZACIÓN

es PostgreSQL con el paquete adicional de PostGIS, ambos de los productos son de libre distribución. Los datos de cáncer provienen de un conjunto de datos de la incidencia de cáncer para Pensilvania, EUA. Utilizaron el software SEER\*stat para calcular las estadísticas mostradas en el atlas. Para este atlas, los datos de tipo de cáncer a ser mostrados son colorectal y de próstata.

El servidor para enlazar la base de datos a el cliente es GeoServer. Éste es usado para manipular los datos geoespaciales. Utilizaron el servicio WFS (Web Feature Service) para permitir la actualización dinámica de mapas en respuesta de las acciones de los usuarios. como también conexiones dinámicas entre múltiples vistas en la pantalla del cliente.

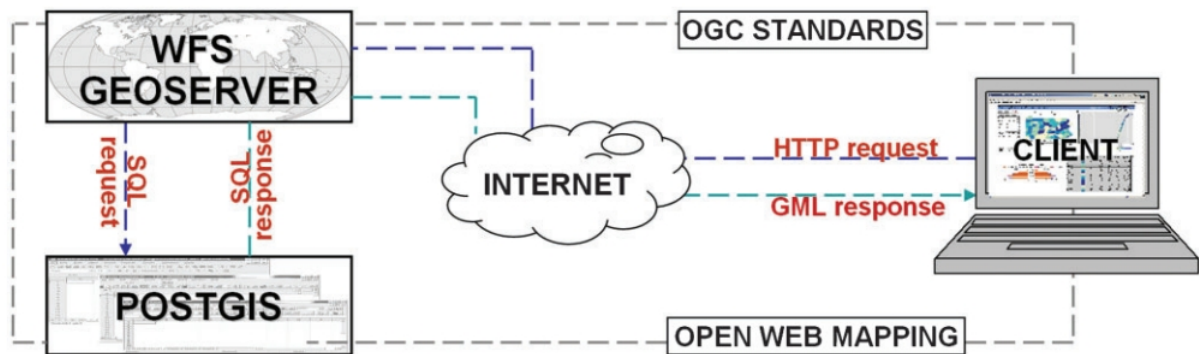


Figura 3.2: Arquitectura del sistema PA Cancer Atlas

Ellos construyeron su propio cliente en Flash. Flash tiene la habilidad de cargar y analizar gramaticalmente los datos en formato XML. Flash también tiene herramientas sofisticadas de dibujo y diseño usadas para personalizar las capacidades de interacción y la estética de las representaciones gráficas basadas en vectores dentro de la interfaz de usuario de una aplicación Web. Cuando un cliente es inicialmente cargado en el navegador Web, entonces Flash envía una solicitud URL a GeoServer para obtener características del estado de Pensilvania. La respuesta de GeoServer es enviado a el cliente en la forma de un documento GML (Geographic Markup Language) conteniendo la geometría de cada condado de Pensilvania. GML es el estándar XML para expresar características geográficas.

### 3.3. Integración del Enfoque AJAX de Servicios Web en SIGs de visualización

En este trabajo [64] se presenta un marco genérico y el rendimiento eficiente para la integración de modelos de AJAX en los sistemas de servicios Web SIG visualización basados

en navegador [65]. La arquitectura de integración se formó mediante el uso de AJAX y Servicios Web en el dominio de los SIG. Hay dos actores principales en la arquitectura de integración - El cliente de visualización y *GIS Web Services*. Este *framework* está diseñado para aplicaciones Web basadas en servicios Web. La arquitectura del *framework* permite la integración de los servicios Web y AJAX.

El *framework* tiene que el cliente de visualización SIG utiliza los servicios Web de la aplicación del navegador de escritorio y por medio de los servicios Web son capaces de responder lo suficientemente rápido. Utilizando el modelo de AJAX para llamar a Servicios Web dio aumentos de rendimiento. Dado que tanto AJAX y los Servicios Web utilizan protocolos basados en XML, para la solicitud y las respuestas, permitió a los desarrolladores integrar fácilmente las aplicaciones de navegador basadas en servicios Web de AJAX. La figura 3.3 muestra como se realiza la petición de dos capa a dos servidores distintos las peticiones del usuario son síncronas y por medio de AJAX se realiza la petición de información geográfica por medio de Servicios Web.

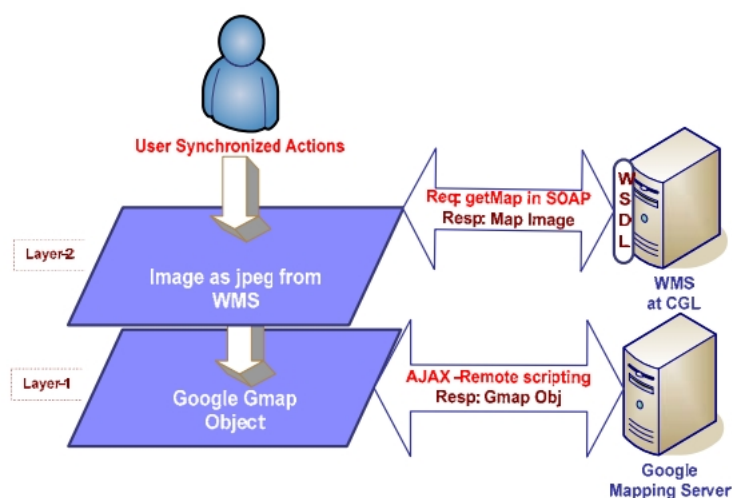


Figura 3.3: Integración de Google Maps con OGC WMS

El uso de Google Maps sólo tiene algunas desventajas en la extracción y visualización de la información acerca de la característica específica seleccionada haciendo clic en el mapa. Mediante el uso de servicios Web del SIG en la misma aplicación y la asignación de esta parte de WMS. Ya que el API de Google Map utiliza el análisis sintáctico DOM, si el tamaño de los datos es de gran tamaño para el servidor es imposible analizar y obtener información sobre las características de un gran conjunto de datos geográficos representado en datos XML estructurados como GML. Mediante la integración de servicios Web de SIG en la visualización la aplicación y el uso de técnicas de extracción de análisis sintáctico resolvieron el inconveniente anteriormente mencionado.

### 3.4. Desarrollo de una aplicación SIG Web para asignación espacial en un inventario de emisiones basado en herramientas *opensource*

Este trabajo [66], se examina el desarrollo de una aplicación SIG Web como parte de un sistema de inventario de emisiones que produce mapas y archivos con las emisiones asignadas espaciales en un formato de cuadrícula. El estudio no se limita en los mapas producidos sino que también presenta las características y capacidades de una aplicación Web. El desarrollo de la aplicación se basa en herramientas de software de código abierto, como MapServer para las funciones de los SIG, PostgreSQL y PostGIS para la gestión de datos y HTML, PHP y Javascript como lenguajes de programación. La figura 3.4, muestra la arquitectura del SIG Web a detalle de los componentes que utiliza.

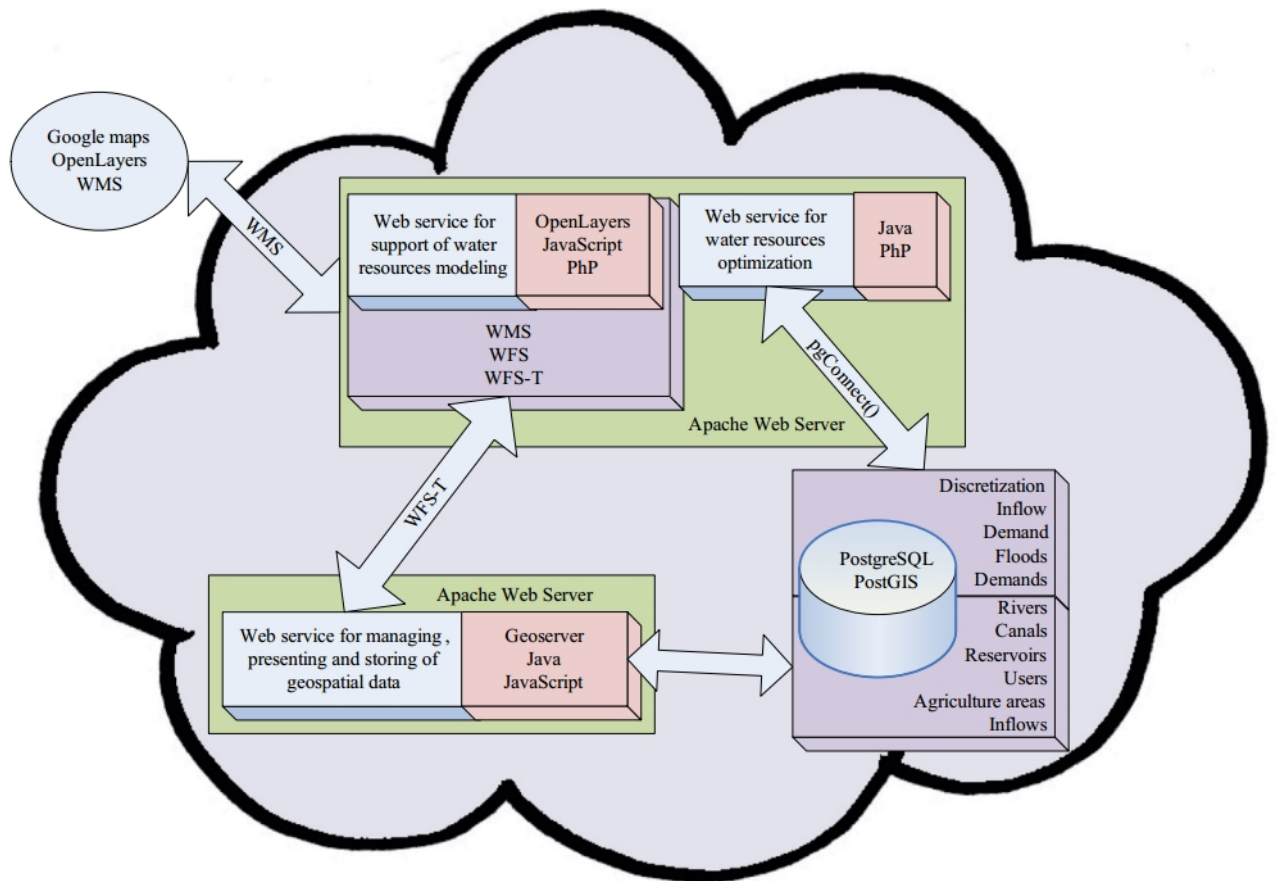


Figura 3.4: Arquitectura del SIG Web para el inventario de emisiones

Además, los procesos de fondo se utilizan de una forma innovadora para manejar el consumo de procedimientos costosos y computacionales de la aplicación en tiempo. Por

otra parte, un servicio de mapas Web fue creado para proporcionar mapas para otros clientes como la versión 3 del API de Google Maps que se utiliza como parte de la interfaz de usuario. La salida de la aplicación incluye mapas en formato raster y vectoriales, mapas con resolución temporal.

El objetivo principal de este desarrollo fue la creación de una aplicación SIG Web. El primer objetivo de este estudio fue crear una herramienta que calcule las emisiones de todas las principales fuentes de Chipre, aparte del transporte por carretera. Esta herramienta es referido como CTEE. La aplicación resultante crea dos tipos de mapas. El primero de éstos es considerado como un vector de asignación de emisiones. El usuario selecciona un tipo de fuente específica, un contaminante, un día del año examinado y hora. A continuación, el nivel mínimo y máximo de las emisiones se calcula y un número de clases se crean con rangos que dependen de estos niveles.

El segundo tipo de mapa se considera como una distribución de las emisiones en un formato raster. El usuario tiene que seleccionar uno o más tipos de fuentes, un contaminante y un tamaño de celda. A continuación, la aplicación divide el mapa en una cuadrícula de acuerdo con el tamaño de la celda seleccionada. Para cada célula de la red, se encuentran las características de cada tipo de fuente, incluso si estas características se incluyen parcial o totalmente en la célula. Cada característica tiene un valor de emisión, que se añade al valor total de células de acuerdo con su tipo.

### **3.5. Diseño e Implementación de una plataforma SIG Web Basada en un *Middleware* SIG de libre distribución**

Este trabajo [67], se propone un diseño de una plataforma SIG Web, la cual es implementada sobre *middle-ware* de libre distribución, se expone en detalle el flujo de trabajo de cada *middle-ware* y como cada *middle-ware* trabaja de forma colaborativa y finalmente verifica la plataforma con un trabajo práctico.

#### **Motor del SIG Web**

El motor de un SIG Web es el núcleo de un sistema de información geográfica basado en Web. Un buen motor de mapas es crucial en mejorar la habilidad de respuesta de un SIG Web. Los mapas en línea se comunican a través de la Internet, lo cual es inevitablemente influenciado por la velocidad de transmisión de la red.

### 3.5. DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA SIG WEB BASADA EN UN MIDDLEWARE

#### Tecnología *Middleware*

Un *middleware* SIG es una especie de bloque de software reusable. El cual provee de interfaces predefinidas por el usuario, estas encapsulan varios servicios del SIG, así los usuarios se enfocan en integrar *middle-wares* a sus aplicaciones con con medios potentes y flexibles en lugar de aplicar funciones más básicas.

Los principios de los SIG Web son mostrados en la figura. Las aplicaciones SIG Web consisten en tres partes: el servidor de bases de datos (capa de datos), el servidor de aplicación (capa de aplicación) y el navegador (capa de presentación de mapas).

La arquitectura diseñada e implementada de esta plataforma SIG Web fue basada sobre un *middleware* SIG de libre distribución acorde a los principios de los SIG Web. La visualización del mapa usa la biblioteca de libre distribución OpenLayers. La capa de aplicación usa TF.NET y la biblioteca FDO como la biblioteca principal de análisis SIG. La capa de datos incluye datos geográficos, datos temáticos, red de mapas y datos vectoriales de otros formatos SIG.

Los datos básicos de mapas son proporcionados por datos bidimensionales, modelos de datos tridimensionales de Google. Los datos temáticos contienen datos almacenados en SQL Server 2005 Express.

Acorde a los principios de diseño y de la arquitectura, el sistema se divide en tres capas en estructura 3.5, el navegador, el servidor de aplicación y el servidor de bases de datos. Correspondientemente, su función es dividida en tres capas, capa de presentación de mapas (cliente), capa de servicios de aplicaciones y capa de servicio de datos.

#### La capa de presentación de mapas

- Ajax y SIG Web. Ajax es la abreviacion de “Asynchronous JavaScript and XML”, el suso de Ajax es necesario para la respuesta de tiempo real c
- El visualizador de mapas ésta basado en OpenLayers. OpenLayers es uno framework bastante popular para la visualización de mapas en línea. OpenLayers accede a todos los datos geospaciales respetando los estándares de la industria. El proceso es: recibir una petición del cliente, convertir a una consulta estándar, construir objetos espaciales,interactuar con la base de datos y regresar los resultados, procesar los resultados a el formato solicitado, regresar el resultado final al cliente.

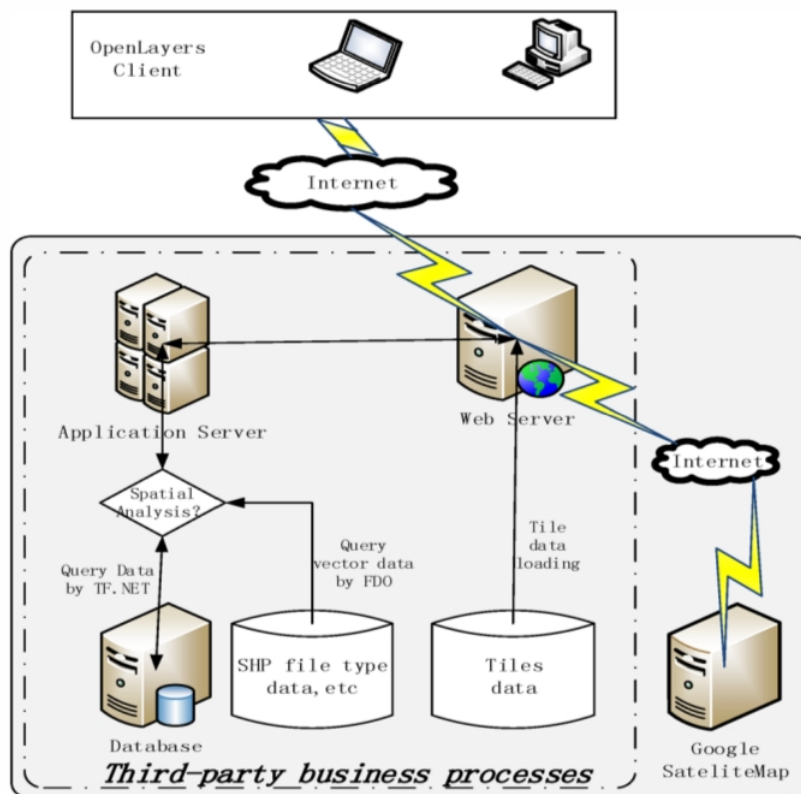


Figura 3.5: Arquitectura del SIG Web basado en software de libre distribución

## Capa de aplicación de servicios basada en .NET

La capa de aplicación de servicios está ubicada en el medio, ésta incluye el servidor Web, el servidor SIG y las extensiones de la aplicación. El servidor está encargado de editar, construir relaciones topológicas, mantenimiento automático de las relaciones de los objetos y del trazo de mapas.

## La capa de datos espaciales basados en múltiples fuentes de datos espaciales

- Motor de acceso a datos espaciales
- Biblioteca SIG y biblioteca de acceso a datos espaciales.
- Diseño del motor de datos espaciales. El diseño del motor de datos espaciales de este trabajo fue implementado en TF.NET y FDO. En el motor de datos espaciales

TF.NET es usado para filtrar los datos geográficos y FDO para acceder a otros formatos de datos SIG.

La implementación de este SIG incorporó AJAX para poder mostrar información sin la necesidad de recargar la página. Este *middleware* desarrollado con software de libre distribución tiene ventajas como una fácil implementación, configuración simple, instalación gratuita y eficiencia alta.

### 3.6. Trabajos SIG, SIG Web y de visualización

En esta sección se mencionan trabajos enfocados en el área de la visualización, paredes de vídeo y su uso en los SIG. La productividad es mejorada especialmente en la navegación en entornos 3D, en la construcción de mejores mapas cognitivos del mundo virtual (Czerwinski et al. [68]). Los siguientes trabajos a enunciar, en el área de Visualización, Navrátil [69] enuncia que las pantallas grandes ofrecen mejorar la usabilidad y el desempeño en el análisis de imágenes de alta resolución. Los casos de estudio para estas pantallas son en la interacción humano computadora, en la computación científica, simulaciones y modelos crean imágenes de alta resolución.

En estado del arte tomamos en cuenta diverso trabajos que se realizaron sobre geovisualización usando herramientas de visualización en alta resolución como son las paredes de vídeo. El trabajo de Ball et al. [70], menciona principalmente dos experimentos con participantes con visión normal. Éstos consistieron en visualizar mapas con diferentes configuraciones de pantallas, de uno, de cuatro y de nueve monitores. El primer experimento está enfocado en tareas de navegación de mapas. El segundo experimento fue en como maximizar la efectividad de una técnica interactiva sobre demanda con gran recuento de pantallas con muchos píxeles. Principalmente en la configuración de 9 monitores se encontró que los participantes de estos dos experimentos, mostraron que los usuarios tuvieron 70 % clics menos, 90 % menos uso del manejador de ventanas y en general una mejora en la precisión y desempeño sobre el monitor.

El hardware utilizado consistió en nueve monitores usaron una Dell Optiplex GX270 con 2.66 GHz con 2 GB de RAM ejecutándose sobre Windows XP, los monitores fueron colocados para formar una matriz de monitores de 3x3. El trabajo concluye que el uso de múltiples pantallas conlleva a un mejor desempeño en las tareas de interpretación geoespacial. Ésto indica que las grandes pantallas de altos números de píxeles mejora la visión y entendimiento de los datos.

Ebert et al. [71] presenta Tiled++, un nuevo enfoque escalable a la solución del problema del bisel que se presenta en paredes de vídeo basadas en *LCD*. Tiled++ usa proyectores para mejorar los marcos del monitor con la información que la imagen perdió o fue de-

formada. Se realizaron prototipos con 3x3 y 5x5 con paneles *LCD*, la combinación de proyectores y paneles *LCD* resulta efectiva en el problema de los biseles. Ésto mejora la usabilidad de los sistemas de múltiples pantallas. Los problemas principales con los biseles normalmente se tratan de resolver con ignorar los biseles y su efecto sobre la continuidad de la escena (*offset approach*), o tratando de compensar el problema del bisel pretendiendo que la malla sea cubierta por la imagen (*overlay approach*).

El trabajo de Sakuraba et al. [11], ellos proponen un sistema de presentación de desastres a gran escala de nombre LIVEDWall (*Larger-Disaster Interactive Visualization Environment for Tiled Display Wall*). El objetivo principal de este sistema es ofrecer soporte para la toma de decisiones en situaciones de desastre a la sede de respuesta a emergencias, desplegando información de los daños. Proponen un SIG basado en la representación y el sistema de compartimiento de la situación de daños entre sedes de manejo de desastres. Por sus siglas en Inglés (*CDS (Counter Disaster Site)*), en su traducción al español son las sedes de manejo de desastres(SMD). Cada *SMDs* usara un entorno con *tiled display wall (TDW)*. Este sistema usa el modelo cliente-servidor en un servidor principal y muchos clientes en cada SMDs. El modelo del sistema LIVEWall se observa en la Figura.3.6. Ellos han construido dos entornos de paredes de vídeo con diferentes especificaciones. La primero con 16 mosaicos con una resolución máxima de 6,400 x 4,800 píxeles, con un sistema Ubuntu Linux 12.04. La asegundo con 27 mosaicos con una resolución máxima de 12,294 x 2,304 píxeles, con un sistemas Ubuntu Linux 11.10.



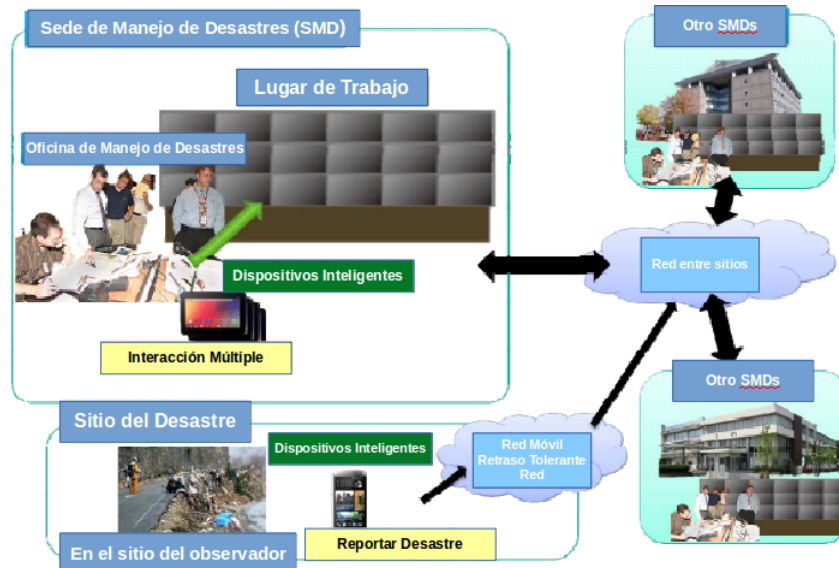


Figura 3.6: El Concepto LIVEWall propuesto por Sakuraba et al. [11]

En el trabajo de Sorokine [72], menciona una extensión del SIG GRASS GIS, permite a los usuarios la realización de tareas de visualización geográfica. Esta extensión recibe el nombre de PD-GRASS para la realización de visualización paralela en pantallas de mosaico. En la operación de la extensión requirieron de un clúster de computadoras y dispositivos de despliegue, que juntos producen un solo campo de visión combinado por varios pantallas. La Fig.4, se muestra el funcionamiento de PD-GRASS en un sistemas de una sola pantalla, pero mostrando como se divide en varias ventanas. PD-GRASS consiste de varios nodos de *renderizado* y un nodo central. Cada uno de los nodos de *renderizado* está conectado a uno más dispositivos físicos de despliegue y todos los nodos están controlados por un nodo central.

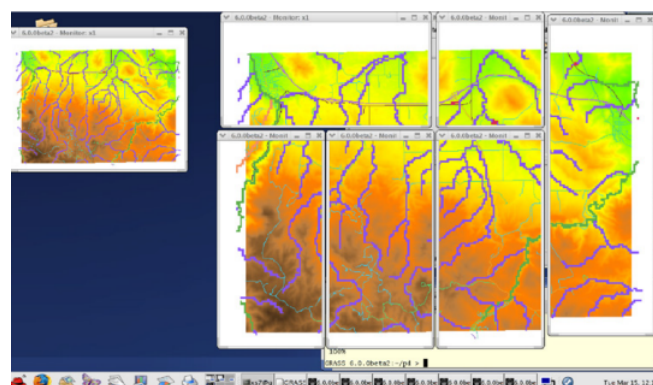


Figura 3.7: Prueba PD-GRASS en un sistema local

Fjukstad et al. [73] presenta un prototipo de un sistema interactivo de nombre Wall-Weather, para el modelado numérico del clima. Haciendo uso de visualización sobre demanda de alta resolución en un muro pantalla y se enfoca de modelos meteorológicos numéricos usando un arreglo de pantallas en alta resolución, donde se despliegan los pronósticos en alta resolución de áreas pequeñas a estudiar. El muro de pantalla consistió de 28 proyectores por 28 computadoras organizadas en una malla de 7x4 para tener una resolución de 7168x3072 píxeles. En la figura 3.8 se muestra el sistema de visualización WallGlobe funcionando.



Figura 3.8: Uso de la Pared de Vídeo con WallGlobe

En el artículo de Yin et al. [74], ellos investigaron en varios algoritmos de renderizado de terrenos en tiempo real y arquitecturas de sistemas paralelos de renderizado. Proponen un sistema de renderizado paralelo para un gran conjunto de datos sobre un terreno. El sistema se compone por 15 nodos de renderizado y 15 proyectores, los nodos son independientes y sincronizados para generar una sola imagen proyectada, los nodos están conectados por Gigabits Ethernet, el sistema operativo de cada computadora es Windows 2003 Server. Se enfatiza en la calibración geométrica como lo más importante en la pared de vídeo, la Figura 3.9 muestra un terreno renderizado y visualizado en la pared de vídeo.

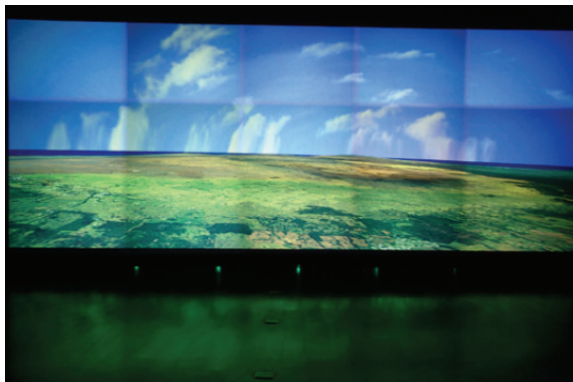


Figura 3.9: Visualización del terreno

### 3.7. Resumen

Este capítulo mostró diferentes arquitecturas de SIG Web, las cuales utilizan elementos comerciales y de libre distribución como son GeoServer, PostgreSQL, PostGIS. La mayoría de estas arquitecturas utiliza el servicios WMS y WFS, WFS-T para el despliegue de mapas Web en un navegador. Las arquitecturas mostradas nos dan una idea de como podemos crear nuestro ambiente Web por medio de estas herramientas y como mejorar las aplicación en un futuro. La parte de proyectos de visualización se enfoca en los SIG con la paredes de vídeo. Esta sección es importante para trabajos futuros por que muestran como se puede adaptar los SIG para ser utilizados en visualizaciones de inmensas cantidades de información.



# Capítulo 4

## Diseño del SIG Web

En este capítulo se describe el diseño del SIG creado con tecnologías Web con cada uno de los componentes de la arquitectura propuesta para la aplicación, para ofrecer servicios Web de mapas. Se menciona el diseño de BDS y su contenido basado en cartas topográficas. Como interactúan las diferentes capas de la arquitectura del SIG. La estrategia para visualizar las cartas topográficas de diferentes zonas geográficas UTM de México.

El núcleo de una aplicación Web es la lógica del lado del servidor. La capa de la aplicación Web por sí mismas puede comprender por muchas capas distintas. El ejemplo típico es una arquitectura de tres capas compuesta por; presentación, negocios y datos.

### 4.1. Diseño por capas lógicas

El concepto de una arquitectura por capas es donde se considera diferentes partes de un sistema de software como tener roles diferentes y separados. Las capas describen las agrupaciones lógicas de la funcionalidad y de los componentes en una aplicación; mientras que los niveles describen la distribución física de la funcionalidad y componentes en servidores separados, computadoras, redes o ubicaciones remotas. Aunque las capas y los niveles utilizan el mismo conjunto de nombres (presentación, negocios, servicio y datos), recordando que sólo los niveles implican una separación física.

Independientemente del tipo de aplicación que se diseñe y si se cuenta con una interfaz de usuario o bien de una aplicación de servicios que sólo expone los servicios (no confundir con la capa de servicios) se puede descomponer el diseño en partes lógicas llamadas capas [12]. Las capas ayudan a diferenciar los diferentes tipos de tareas realizadas por los componente, por lo tanto es más fácil para crear un diseño que apoya la reutilización de componentes. Cada capa lógica contiene un número de tipos de componentes discretos agrupados en subcapas; con cada subcapa realiza un tipo específico de tarea. Solo con identificar los tipos de componentes genéricos que existen en la mayoría de las

soluciones, se puede construir un mapa significativo de una aplicación o servicio.

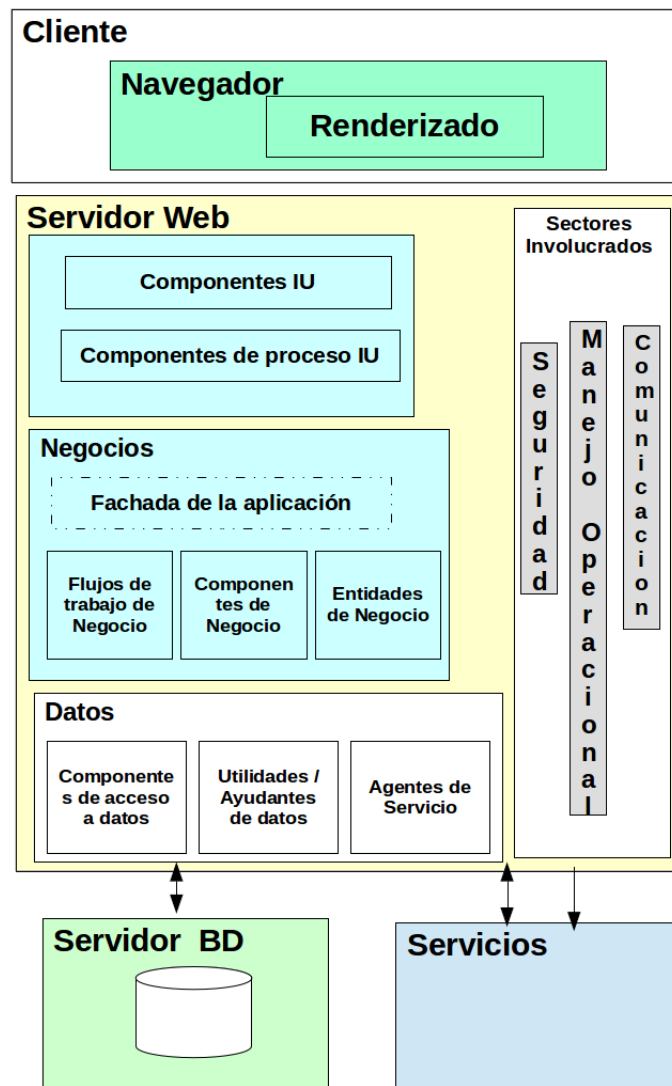


Figura 4.1: Arquitectura de tres capas para aplicaciones Web (de acuerdo a [12]).

En el nivel más alto y abstracto, la vista de la arquitectura lógica de cualquier sistema puede considerarse como un conjunto de componentes de cooperación agrupados en capas. La Figura 4.1 muestra de forma simplificada, el nivel más alto de representación de esas capas y sus relaciones con los usuarios, otras aplicaciones que requieren servicios implementados dentro de la capa de negocios, fuentes de datos, tales como bases de datos relacionales o servicios Web proporcionan acceso a los datos y servicios externos o remotos que son consumidos por la aplicación.

Explicando la Figura 4.1, se aprecian tres niveles pero sin el nombre correspondiente, el primer nivel representa al cliente o navegador que corresponde a la capa de presentación, el segundo nivel es mucho más complejo, con componentes de la Interfaz de Usuario (IU), es en este nivel donde se crean las reglas de negocio para realizar la comunicación entre el cliente por medio de la IU y la base de datos utilizando componentes de acceso a datos, o si se requiere información por medio de servicios se utilizan agentes de servicio para obtenerla. El último nivel es la base de datos y servicios, éstos últimos se utilizan para obtener información de fuentes de externas. Los sistemas Web, involucran otros sectores a considerar como son la seguridad, la forma de comunicación y el manejo operacional.

Una aplicación puede consistir de un número determinado de capas básicas. El diseño más común es tres y consiste de:

- **Capa de presentación.** Esta capa contiene la funcionalidad orientada al usuario, la cual es responsable de manejar la interacción del usuario con el sistema y generalmente consiste de componentes que proveen de un puente común con el núcleo de la lógica de negocios, encapsulado en la capa de negocios. En el SIG que se diseñó la capa de presentación provee de una interfaz para que los usuarios puedan acceder, procesar e interactuar con mapas y datos espaciales.
- **Capa de negocios.** Esta capa implementa la funcionalidad básica del sistema y encapsula la lógica de negocios relevante. Generalmente consiste en componentes, algunos de los cuales exponen las interfaces de servicio que otros usuarios puedan utilizar. La capa de negocios es usada para recibir y procesar las peticiones del usuario por medio de la capa de presentación, producir mapas y manejar el sitio.
- **Capa de datos.** Esta capa provee acceso a los datos alojados dentro de los límites del sistema y los datos expuestos por otros sistema en la red; tal vez se acceda a través de servicios. La capa de datos expone interfaces genéricas que los componentes en la capa de negocios puedan consumir. La capa de datos incluye los datos geográficos y los datos del servidor.

#### 4.1.1. Capa de Presentación

Cualquier sistema de información necesita comunicarse con entidades externas; las cuales son seres humanos o otras computadoras. Una gran parte de esta comunicación involucra la presentación de la información a esas entidades externas y permitir les actuar recíprocamente con el sistema por medio de operaciones, para conseguir respuestas. Los componentes de un sistema de información que son fieles a estas tareas forman la capa de presentación. Esta capa podría ser implementada por medio de una interfaz de usuario

gráfica, o podría ser un módulo que da formato a un conjunto de datos de una representación sintáctica determinada.

La capa de presentación usualmente es conocida como el cliente de un sistema de información lo cual no es exactamente correcto. Todos los sistemas de información tienen clientes, los cuales son entidades que usan los servicios proveídos por el sistema. Los clientes son externos e independientes del sistema. En ese caso, éstos no son parte de la capa de presentación.

La capa de presentación contiene los componentes que implementan y muestra la interfaz de usuario para controlar la interacción de usuario. Esta capa incluye controles para las entradas de usuario y el despliegue, además de los componentes que organizan la interacción del usuario. Una aplicación típica suele incluir los siguientes componentes en la capa de presentación:

- **Componentes de la Interfaz de usuario.** Éstos son los elementos visuales utilizados para mostrar la información que el usuario introduce.
- **Componentes de la Lógica de presentación.** La lógica de presentación es el código de aplicación, el cual define el comportamiento lógico y la estructura del uso en un camino, que es independiente de cualquier puesta en práctica de interfaz de usuario específica.

Para la creación de la capa de presentación, se utilizan herramientas para controlar el comportamiento de la página Web del SIG Web. Esta capa esta compuesta por AJAX, jQuery, HTML, JavaScript y la herramienta OpenLayers 3.0 para controlar la visualización de los elementos vectoriales.

### OpenLayers 3.0

El nacimiento de OpenLayers fue en 2006, esta herramienta de libre distribución aparece como una alternativa a Google Maps y otros proveedores de APIs. OpenLayers gana más atención en 2007, cuando adopta el proyecto OpenStreetMap para su sitio Web.

OpenLayers 3 (OL3) es un rediseño fundamental de la biblioteca de mapeo Web Open Layers. La version 2 es ampliamente utilizada, desde los primeros días del desarrollo de JavaScript, pero cada vez se muestra más ineficiente. OL3 ha sido reescrito desde cero para utilizar patrones de diseño moderno [75]. Esta diseñado también para nuevas características, tales como el despliegue de mapas 3D, o el uso de WebGL para rápidamente mostrar una inmensa cantidad de conjuntos de datos vectoriales, las cuales son añadidas en futuras versiones.

OL3 esta basada sobre Google Closure tools, la cual permite utilizar un conjunto de



herramientas para la construcciones de ricas aplicaciones Web con JavaScript. El uso de éste para manejar lo básico como DOM o el control de eventos significa que los desarrolladores se concentran en la funcionalidad del mapeo y asegurarse de que el software subyacente es bien probado y multi-navegador. Closure Library está especialmente diseñada para ser optimizado por el Closure Compiler.

La clase `ol.Map` es el elemento principal de la biblioteca OpenLayers 3.0, en el apéndice A se describe a detalle los elementos principales de esta biblioteca y cual es la función de cada elemento que la conforma. Por mencionar la clase `ol.Map` hace uso sus subclases `ol.layer`, `ol.View`, `ol.Source`, para poder desplegar un mapa Web simple. Para el despliegue de diferentes capas vectoriales es necesario combinar diferentes tecnologías principalmente de AJAX y jQuery con OpenLayers, para poder realizar peticiones asíncronas al servidor, estas peticiones contienen valores que representan la zona seleccionada o estado, el tipo de capa solicitada.

## JavaScript

JavaScript fue inventado por Netscape en 1995, es un lenguaje de tipo *script* que se ejecuta principalmente en los navegadores Web para crear paginas dinámicas e interactivas. Es el lenguaje de tipo script mas popular, utilizado por millones de paginas Web. Javascript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora [76]. Una de las innovaciones que ayudó a cambiar el modo en que vemos Javascript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código Javascript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejora de capacidad permitió superar viejas limitaciones de rendimiento y confirmar al lenguaje Javascript como la mejor opción para la Web.

## jQuery

jQuery es una biblioteca rápida y concisa de JavaScript creada por John Resig en 2006 con un lema: “Escribir menos, hacer más”. JQuery simplifica el desplazamiento de documentos HTML, el manejo de eventos, la animación y las interacciones Ajax para un rápido desarrollo Web. jQuery es un conjunto de herramientas JavaScript diseñado para simplificar varias tareas escribiendo menos código. Aquí está la lista de características principales importantes soportadas por jQuery:

- Manipulación de DOM: jQuery facilita la selección de elementos DOM, negociarlos y modificar su contenido mediante el uso del motor selector de fuente abierta multi-navegador llamado Sizzle.

- Manejo de eventos: jQuery ofrece una manera elegante de capturar una gran variedad de eventos, como un usuario haciendo clic en un enlace, sin la necesidad de enturbiar el código HTML con los manejadores de eventos.
- Soporte de AJAX: jQuery le ayuda mucho a desarrollar un sitio con una gran cantidad de funciones usando la tecnología AJAX.
- Animaciones: jQuery viene con un montón de efectos de animación incorporados.
- Peso ligero: jQuery es una biblioteca muy ligera - de unos 19KB de tamaño.
- Soporte Cross Browser: jQuery tiene soporte para varios navegadores y funciona bien en IE 6.0+, Mozilla FireFox 2.0+, Safari 3.0+, Chrome y Opera 9.0+.
- 

## AJAX

AJAX es un acrónimo de *Asynchronous Java Script + XML* [77], representa una pieza fundamental en el desarrollo de aplicaciones Web. Ajax no es una tecnología por si misma. Es realmente muchas tecnologías, cada una floreciendo por su propia cuenta, las cuales se unen en nuevas y poderosas formas. Describe como usar tecnologías como HTML, XHTML, CSS, JavaScript, DOM, XML y XMLHttpRequest. AJAX incorpora:

- Presentación basada en estándares utilizando XHTML y CSS.
- Visualización y interacción dinámica con *Document Object Model*(DOM).
- Intercambio de datos y gestión por medio de XML y XSLT.
- Recuperación de datos asíncrona utilizando XMLHttpRequest.
- JavaScript conjunta todo.

Con la combinación de las tecnologías anteriormente mencionadas, es posible lograr aplicaciones Web capaces de actualizarse continuamente sin tener que volver a cargar la página completa. En las aplicaciones Web tradicionales las acciones del usuario en la página provocan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario. AJAX mejora la interacción del usuario con la aplicación Web, evitando recargar constantemente la página y permitir el intercambio de información con el servidor en un segundo plano.

XHTML (*Extensible Hypertext Markup Language*), se basa en la sintaxis XML. CSS (*Cascading Style Sheets*) es un lenguaje de hoja de estilo, que se utiliza para describir el aspecto de un documento Web. DOM especifica la forma de referirse a elementos XML

o HTML como objetos, esto facilita la actualización del contenido de las páginas Web. XMLHttpRequest es una API DOM que puede ser llamada por un *script* del lado del cliente para enviar una solicitud HTTP a un servidor Web, para obtener la respuesta y para manejar contenido devuelto dentro del lenguaje de scripting.

De las características de AJAX anteriormente mencionadas, se observa las siguientes ventajas:

- La interacción Web se vuelve más eficiente mediante el uso de XMLHttpRequest y DOM de forma asíncrona. Las aplicaciones Web recuperaran los datos en forma de XML desde el servidor de forma asíncrona en el *backend* sin interferir con la visualización y el comportamiento de la página existente ya que la información sólo se actualiza en los objetos DOM asociados.
- Los usuarios finales tienen una experiencia Web más rica a través de las funciones implementadas por las bibliotecas JavaScript basadas en el *framework* AJAX.
- XML facilita la comunicación entre el lado del cliente y del servidor.

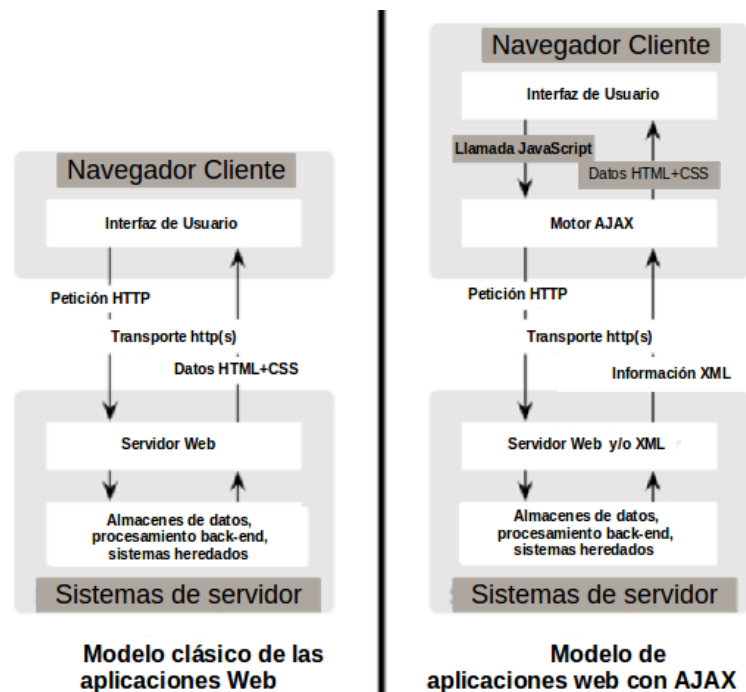


Figura 4.2: Comparación del modelo clásico y Ajax.

La Figura 4.2 muestra una comparativa entre el modelo clásico y Ajax de aplicaciones Web, se observa que en las aplicaciones tradicionales, la mayoría de las acciones del usuario

en la interfaz dispara una petición HTTP de vuelta a un servidor Web. El servidor realiza algo de procesamiento como: recuperación de datos, procesar una gran cantidad de datos, conversación con varios sistemas heredados y luego devuelve en una página HTML al cliente. Es un modelo adaptado del uso original de la Web como un medio de hipertexto. El problema de las aplicaciones tradicionales es que mientras el servidor esta haciendo algo, el usuario esta esperando.

Las aplicaciones Ajax eliminan la naturaleza *start-stop-start-stop* de la interacción en la Web mediante la introducción de un intermediario -un motor Ajax- entre el usuario y el servidor. En lugar de cargar una página, al inicio de la sesión, el navegador carga un motor Ajax, escrito en JavaScript y normalmente escondido en un marco oculto. Este motor es responsable de representar la interfaz que el usuario ve y de comunicarse con el servidor en nombre del usuario. El motor de Ajax permite que la interacción del usuario con la aplicación de forma asíncrona, independientemente de la comunicación con el servidor.

La capa de presentación esta integrada por diferentes tecnologías y bibliotecas para poder realizar la visualización de mapas Web. La figura 4.3 muestra los componentes que integran la capa de presentación, siendo el lenguaje JavaScript esencial al ser utilizado por jQuery, AJAX y OpenLayers 3.0. La combinación de estas tecnologías en la capa de presentación permite la visualización de múltiples capas sobre un mapa sin tener que recargar o actualizar la página.

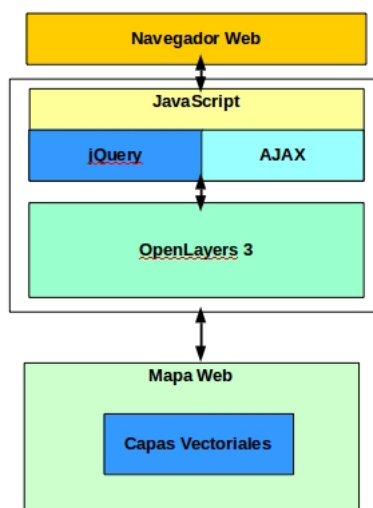


Figura 4.3: Diseño de la capa de presentación con sus componentes.

#### 4.1.2. Diseño de la interfaz

Para el diseño de la interfaz del SIG Web, se propuso una interfaz con dos mapas, el primer mapa sirve para visualizar los elementos vectoriales, el segundo mapa permite la navegación y selección de zonas o estados de la república Mexicana (ver Figura 4.4).

La Figura 4.4 a, muestra el menú de control de capas, por medio de este menú el internauta podrá mostrar o ocultar capas que tenga el mapa de visualización y éste se actualizará cada vez que se agregue una nueva capa. En 4.4 b, corresponde al mapa de visualización donde se desplegarán los mapas generados o agregado de archivos geográficos, este compartirá coordenadas con el mapa de navegación. El mapa de visualización contará con herramientas para hacer mediciones y realizar anotaciones. La parte de la interfaz 4.4 c, es el menú de selección de capas, donde se muestra las capas que se podrán generar a partir de la selección de estados o zonas de México en el mapa de navegación. Por último la parte 4.4 d corresponde al mapa de navegación, éste es el más importante por que en él se realiza la selección de los estados o de las zonas geográficas. Éstos son los componentes principales de la aplicación.

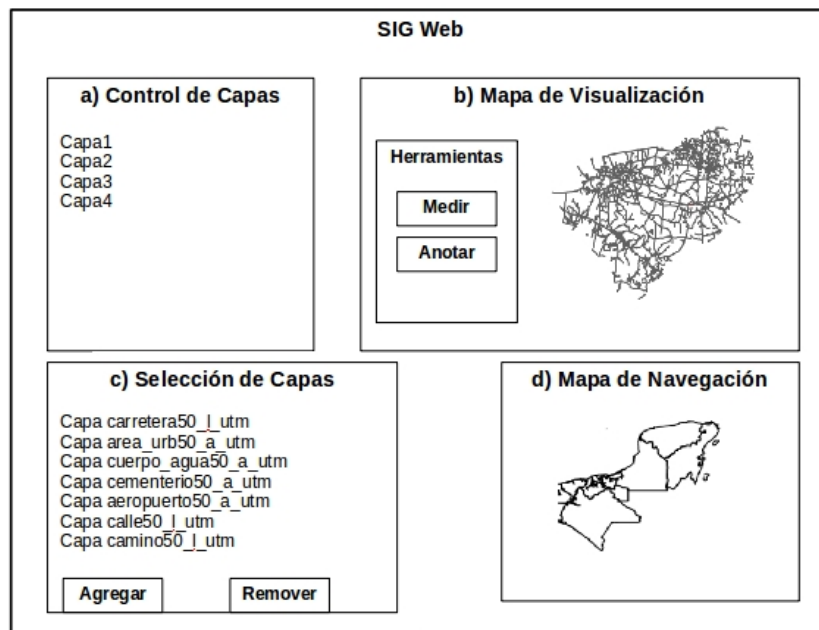


Figura 4.4: Diseño de la interfaz del SIG-Web.

### 4.1.3. Capa de Negocios

La capa de negocios, la capa de lógica de aplicación o la capa de lógica de negocios. La gran mayoría de los sistemas realizan algo del procesamiento de datos detrás de los resultados que se están entregando. Este procesamiento consiste en un programa que implementa la operación actual solicitada por el cliente a través de la capa de presentación. Nosotros nos referimos a todos esos programas como la capa negocios. También estos programas son referidos como los servicios que el sistema ofrece.

La capa de negocios y los componentes que podría tener esta capas usualmente son las siguientes.

- **Componentes de la lógica de negocios.** La lógica de negocios esta definida como cualquier lógica de aplicación que se encarga de la recuperación, el tratamiento, la transformación y la administración de datos de la aplicación y asegurar la coherencia de datos y validez. Para maximizar las oportunidades de reutilización, los componentes de lógica de negocio no deben contener ninguna lógica de comportamiento o aplicación que es específica para un caso de uso o historia de usuario. Los componentes de lógica de negocios se subdivide en las siguientes dos categorías:
  - Componentes del flujo de trabajo de negocios.
  - Componentes de entidad de Negocios.

La capa de negocios utiliza las herramientas Open Source como son Apache Tomcat Server y un Servidor Apache. El primero es de utilidad para instalar herramientas SIG como Geoserver, el cual es un servidor de mapas, éste permite la consulta de datos geográficos realizando un puente con una base de datos geográfica. El servidor Apache es de utilidad para poder instalar PHP y por medio de este lenguaje realizar consultas SQL a la capa de datos y para crear módulos que permitan la comunicación entre la capa de presentación y la capa de negocios.

### Servidor Web Apache

La fundación Apache tiene el proyecto del Servidor Apache HTTP un servidor HTTP de alto desempeño y escalable. El servidor Web Apache o Apache esta optimizado para servir contenido con alto desempeño sobre el protocolo HTTP. Es posible integrar un servidor Apache y Apache Tomcat para beneficiarse del contenedor de servlet. Apache es el servidor Web utilizado con más frecuencia en los sistemas Linux. Los servidores Web se emplean para entregar las páginas que los equipos cliente solicitan. Los clientes normalmente solicitan y visualizan las páginas Web mediante aplicaciones de navegación como Firefox, Opera, Chromium e Internet Explorer.

Existen varias razones por las que se considera integra Tomcat con Apache, la primera razón ha sido el desempeño. Apache es un servidor Web de alto desempeño que sirve contenido estático como archivos html, imágenes y vídeos. La segunda razón es la configurabilidad del servidor Apache. Con 15 años de existencia, el servidor Apache, tiene una alta popularidad y una gran comunidad de desarrolladores con diferentes conocimientos en lenguajes de programación. Miles de módulos esta disponibles como extensiones para los servidores Apache. Los módulos van desde las extensiones específicas de cada lenguaje de programación (para PHP, Pearl, Python, entre otros).

### PHP

El lenguaje de PHP (acrónimo recursivo *PHP Pre Hypertext -processor*) se ha convertido en unos de los lenguajes de programación más populares, el cual es utilizado para el desa-

rollo de aplicaciones Web. El lenguaje ha evolucionado, para permitir al programador desarrollar rápidamente programas libres de errores y bien contruidos utilizando técnicas de programación orientada a objetos y de procedimientos. Proporciona la capacidad de utilizar muchas bibliotecas preexistentes de código que, o bien vienen en una instalación básica o se instalan dentro del entorno PHP. Ésto le da múltiples maneras de completar una tarea en particular. Es más flexibilidad que muchos otros idiomas. La facilidad con la que las bibliotecas de código adicionales son añadidos al medio ambiente es una de las muchas fuerzas impulsoras de su popularidad.

PHP es un lenguaje del tipo *script*. Un lenguaje script es diferente de los lenguajes de programación actuales. Estos lenguajes no necesitan de un compilador. La primera vez que se accede a el código, éste es interpretado línea por línea a medida que se ejecuta el programa. Una vez que el código se ha ejecutado una vez, el código interpretado permanece en la memoria del ordenador, o servidor, para otras ejecuciones.

Al servidor Web Apache se le instaló un módulo de PHP para permitir interpretar código PHP. Apache permite la instalación de módulos, los cuales permiten utilizar diferentes lenguajes programación en el servidor, PHP es necesario para la implementación de la capa de negocios .

### Módulo PostgreSQL para PHP

Este paquete provee de un módulo para la conexión directa con bases de datos PostgreSQL a partir de código PHP. También se incluye el módulo `pdo_pgsql` para el uso con la extensión de PHP Data Object.

Los componentes de la capa de negocios se aprecian en la Figura 4.5, estos elementos en conjunto funcionarían para la creación de consultas cuyo resultado será almacenado en un archivo GeoJSON, existirán diversos archivos para generar las diferentes capas vectoriales . Los archivos GeoJSON proporcionarían un tipo de servicio Web de estilo REST (ver subsección 2.7.3), los cuales solo necesitan de la URL para enviar parámetros a la capa de negocios.

#### 4.1.4. Capa de datos

La mayoría de las aplicaciones que deben acceder a datos que son almacenados en bases de datos, las cuales son a menudo bases de datos relacionales. Los componentes de acceso de datos en esta capa son responsables de enviar los datos almacenados en las bases de datos a la capa de negocios.

Esta capa incluye todos los mecanismos de persistencia de datos (servidores de base, archivos compartidos, etc) y la capa de acceso de datos encapsula los mecanismos de persisten-

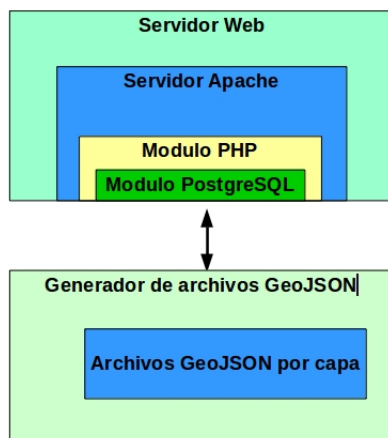


Figura 4.5: Componentes de la Capa de Negocios.

cia y expone los datos. La capa de datos provee acceso a la base de datos ejecutando un conjunto de sentencias SQL o procedimientos almacenados. Esta capa solo puede recibir peticiones de la capa de negocios por medio de diferentes métodos que requieran datos por medio de la ejecución de consultas. Esta capa usualmente tiene un Servidor SQL el cual provee mecanismo de almacenamiento, recuperación de datos y conceder diferentes permisos a los usuarios dependiendo del administrador del sistema.

## PostgreSQL y PostGIS

### PostgreSQL

PostgreSQL es un sistema de gestión de base de datos Objeto-Relacional (*object-relational database management system* (ORDBMS)) basado en POSTGRES, desarrollado en la Universidad de California en el departamento de ciencias de la computación de Berkeley [78].

POSTGRES fue pionero en muchos conceptos que sólo estuvo disponible en algunos sistemas de bases de datos comerciales mucho más tarde. PostgreSQL creado en 1996 es un descendiente de código abierto del código original de Berkeley. El cuál soporta una gran parte del estándar SQL y ofrece muchas características modernas:

- consultas complejas.
- llaves externas.
- vistas actualizables.
- integridad transaccional.



- control de concurrencia multiversión.

## PostGIS

PostGIS es una extensión de base de datos espaciales para la bases de datos objeto-relacional PostgreSQL( SQL siendo el lenguaje de consulta estructurado) [43]. PostGIS añade tipos de datos geométricos y funciones espaciales a la base de datos PostgreSQL. Los tipos de datos geométricos soportados son “Points”, “LineStrings”, “Polygons”, “MultiPoints”, “MultiLinesStrings”, “MultiPolygons” y “GeometryCollections”.

PostGIS es un proyecto encabezado por Refrations Research. PostGIS proporciona más de 300 operadores espaciales, funciones espaciales, tipos de datos espaciales y mejoras en la indexación espacial. Si agrega a la mezcla las características complementarias que PostgreSQL y otros proyectos relacionados con PostgreSQL proporcionan, entonces tendrá a su disposición una potencia potente que es muy adecuada para el trabajo duro y una valiosa herramienta de entrenamiento para conceptos espaciales.

Es necesario para la creación de la capa de datos un ORDBMS, se selecciono PostgreSQL por ser de libre distribución. PostgreSQL permite utilizar su extension PostGIS, la cual ofrece funciones geográficas. PostgreSQL y PostGIS permitirán utilizar la base de datos MapasMexico, la cual contiene las cartas geográficas de la INEGI.

Las tablas de la base de datos MapasMexico, contienen un campo especial denominado “geom”, que contiene información del tipo geográfica como son coordenadas, las cuales forman puntos, líneas y polígonos, sin PostGIS no es posible que PostgreSQL interprete correctamente esta información. PostGIS permitirá realizar consultas espaciales a esta base de datos, para resolver nuestros problemas de visualización.

## Cartas Topográficas

Las cartas topográficas son documentos que representan la concepción gráfica del inventario de la infraestructura, orografía, hidrografía y de población del país, así como de su distribución geográfica. Las cartas topográficas de la INEGI son de una escala de 1:50,000, las cuales utilizan un formato regular de quince minutos de latitud por veinte minutos de longitud. Las cartas cubren áreas de aproximadamente 960 kilómetros cuadrados y representan gráficamente las principales características del medio físico. Estas cartas para su publicación utilizan colores, líneas, áreas, símbolos y textos. La figura 4.6, muestra una carta topográfica con diferentes capas y se enfoca en una zona específica.

La capa de datos incorpora información obtenida de cartas Topográficas, estas contienen información específica de los Estados Unidos Mexicanos. La capa de datos contiene 97 tablas que representan diferentes capas de estas cartas. La tabla 4.1 muestra algunas tablas de la base de datos, que tienen diferentes datos de interés como son caminos, calles, áreas urbanas, cuerpos de agua entre otras, Estas tablas se seleccionaron sobre las demás, por poseer muchos o pocos polígonos esto ayuda a evaluar el desempeño de la aplicación

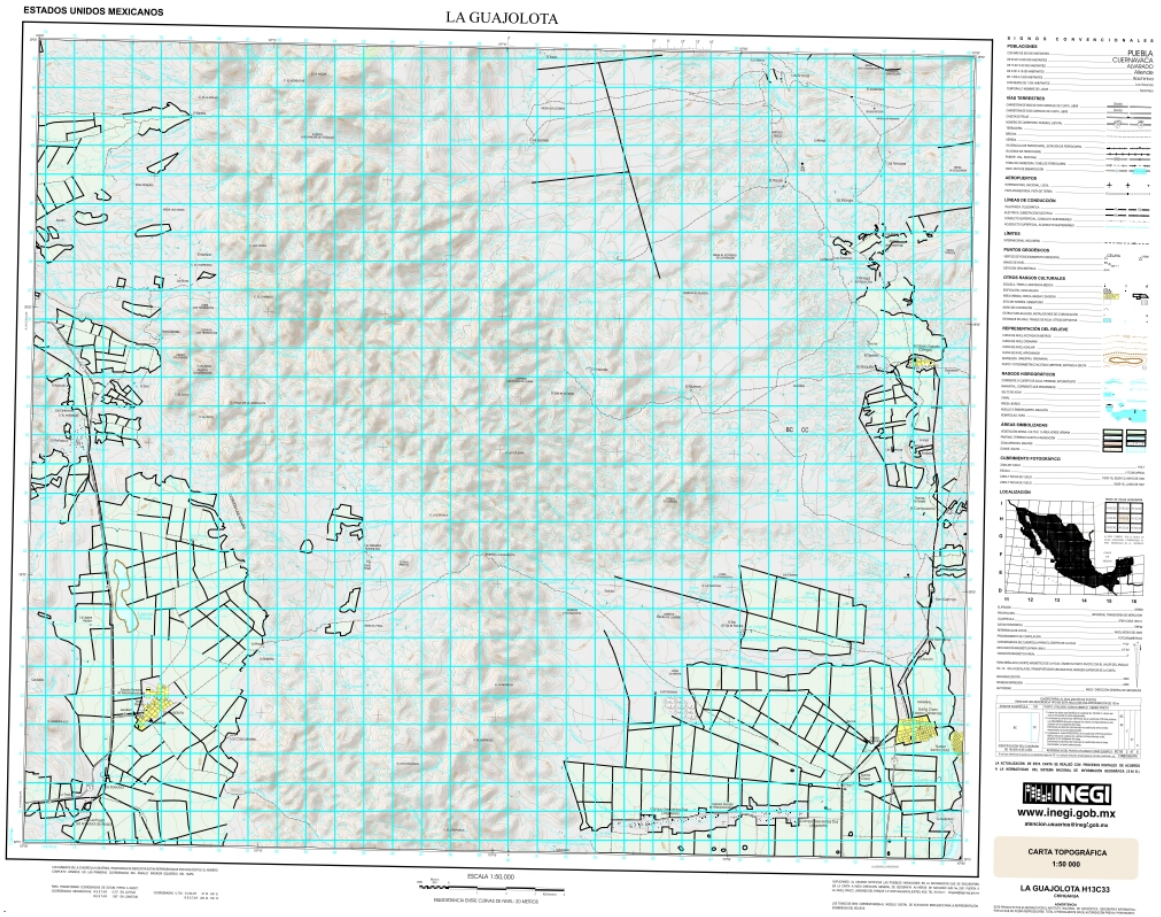


Figura 4.6: Ejemplo de Carta Topográfica, Zona La Guajolota Clave:H13C33.

para ubicar si los lugares están ubicados correctamente de acuerdo a sus coordenadas. Además, estas tablas también poseen campos de información nominal diferentes al resto de las tablas que no se seleccionaron.

<b>Tablas de la base de datos</b>
carretera50_l_utm
area_urb50_a_utm
cuerpo_agua50_a_utm
cementerio50_a_utm
aeropuerto50_a_utm
area_ver_ur50_a_utm
calle50_l_utm
camino50_l_utm
pista_aviac50_a_utm
via_ferrea50_l_utm
vegeta_dens50_a_utm
localidad50_p_utm

Tabla 4.1: Tablas obtenidas por medio de cartas topográficas.

#### 4.1.5. Interacción entre capas

La arquitectura basada en tres capas o niveles tiene la característica que siempre existe una comunicación de forma bidireccional y siempre se realiza por medio de la capa de en medio o capa de negocios. En la creación de sistemas Web es usual separar el sistema en *front-end* y *back-end*, para mantener partes del sistema separadas.

La aplicación cliente (Navegador Web) que forma parte de la capa de presentación recibe entradas de los usuarios y envía una petición a la capa de negocios. La aplicación servidor recibe y procesa la petición, pasándola a la capa de datos. La capa de datos recibe la petición y ejecuta consultas SQL y envía la información a la capa de negocios. La arquitectura propuesta es mostrada en la Figura 4.7.

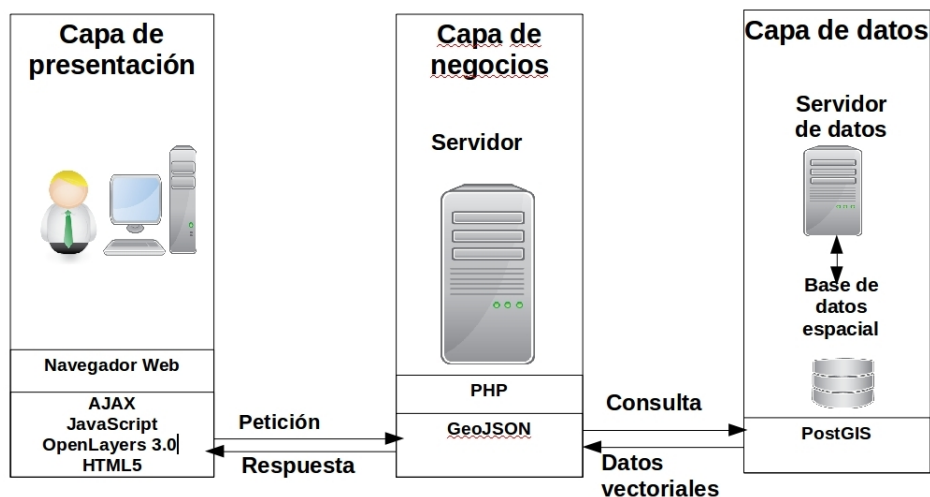


Figura 4.7: Arquitectura de tres capas del SIG.

Como es visible en la figura 4.7, las capas no se encuentran necesariamente en un mismo equipo sino que están repartidas en diferentes equipos servidores, computadoras personales, para lograr un mejor rendimiento y funcionamiento. Ésto para poder aprovechar al máximo la arquitectura de tres capas, que permiten fácilmente aprovechar el consumo de información desde diferentes equipos. Este tipo de arquitectura se puede implementar diferentes formas de seguridad en cada una de las capas, para prevenir ataques. Pero para el desarrollo de el ambiente de visualización se realizaran las tres capas de manera lógica. Las actividades principales se ven en la siguiente Figura 4.8, donde se aprecia con claridad que actividades realizara cada una de las capas que componen al ambiente de visualización.

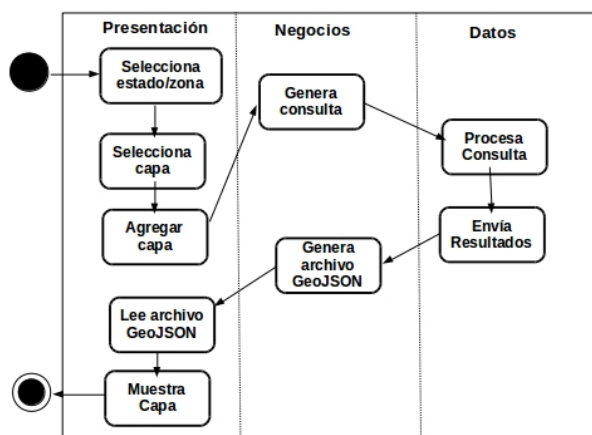


Figura 4.8: Diagrama de actividades por capa.

## 4.2. Casos de Uso del SIG Web

Para el SIG Web, se requirieron diferentes casos de uso, en la Figura 4.9 se aprecia las actividades principales del ambiente de visualización Web. El único actor que compone este ambiente es el Internauta, el puede realizar las siguientes actividades en el sistema: Agregar Capa (ver Tabla 4.2), Generar Capa (ver Tabla 4.3), Clasificar por medio de datos nominales (ver Tabla 4.4), Mostrar u ocultar capas (ver Tabla 4.5) y por último la actividad Mostrar información de los objetos geográficos (ver Tabla 4.6). Estas actividades son las contempladas a implementar en el sistema Web.

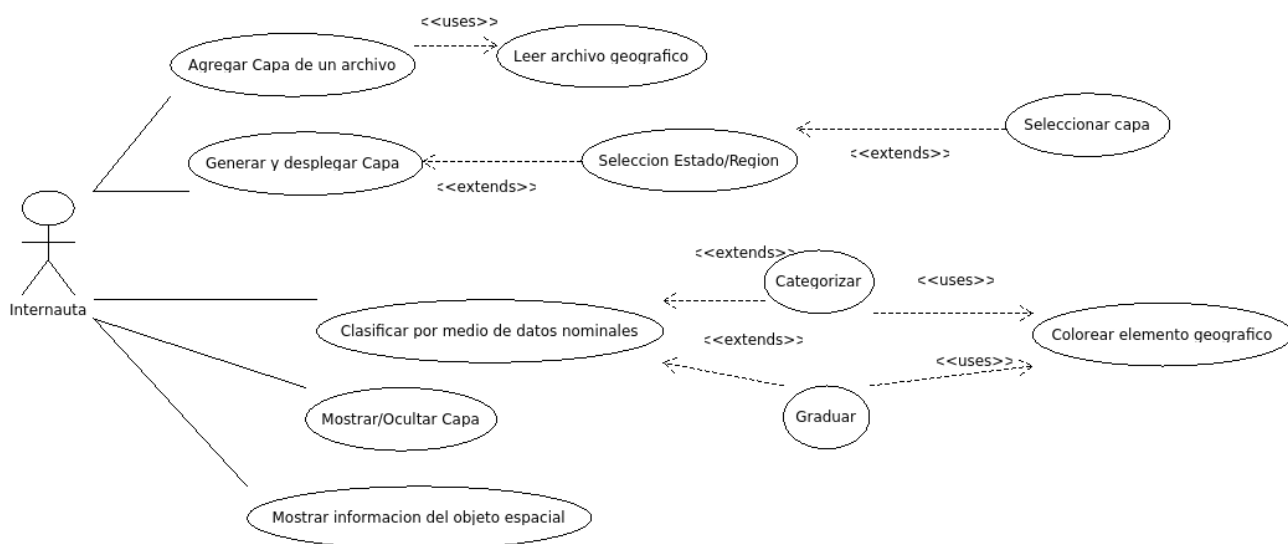


Figura 4.9: Caso de uso del ambiente de visualización.

<b>Caso de Uso</b>	AgregarCapa
<b>Actor</b>	Internauta
<b>Propósito</b>	Agregar una nueva capa a partir de un archivo y mostrarla en el mapa de visualización. La capa tendrá sus opciones en el menú de control de capas.
<b>Resumen</b>	El internauta en el sitio Web, selecciona la opción de agregar capa por medio de un archivo. El internauta selecciona un archivo de formato GeoJSON o KML (archivos con información geográfica). El sistema generara una nueva capa de este archivo y se mostrara en mapa de visualización
<b>Tipo</b>	Primario

Tabla 4.2: Caso de Uso: Agregar Capa.

<b>Caso de Uso</b>	Generar Capa
Actor	Internauta
Propósito	Generar y mostrar una nueva capa en el mapa de visualización y desplegar sus opciones en el menú de control de capas.
Resumen	El internauta en el sitio Web selecciona un estado o zona geográfica de la República Mexicana del mapa de navegación y la capa que desea visualizar. El internauta agrega la capa selecciona, entonces se genera la consulta para generar la nueva capa. La capa generada se mostrara en el mapa de visualización.
Tipo	Primario

Tabla 4.3: Caso de Uso: Generar y desplegar Capa.

<b>Caso de Uso</b>	Clasificar por medio de Datos Nominales
Actor	Internauta
Propósito	Clasificar las capas vectoriales por a partir de sus datos nominales. El internauta podrá clasificar por medio de categorización o graduación de los datos
Resumen	El internauta en el menú de categorización o graduación podrá realizar una clasificación a partir de los datos nominales de las capas vectoriales. El internauta seleccionara la capa y una de las características para realizar la clasificación. La clasificación se observara en el mapa de visualización al cambiar de color los elementos geográficos.
Tipo	Primario

Tabla 4.4: Caso de Uso: Clasificar por medio de datos nominales.

<b>Caso de Uso</b>	Mostrar o Ocultar Capa
Actor	Internauta
Propósito	Mostrar u Ocultar las capas que se encuentren en el mapa de visualización. El menú de control de capas, dispondrá de las opciones para mostrar u ocultar capas.
Resumen	El internauta en el menú de control de capas tendrá las opciones para controlar las capas que se generaron o que se añadieron.
Tipo	Primario

Tabla 4.5: Caso de Uso: Mostrar o Ocultar Capa.

<b>Caso de Uso</b>	Mostrar o Ocultar Capa
Actor	Internauta
Propósito	Mostrar información al internauta del objeto geográfico seleccionado.
Resumen	El internauta en el menú de control de capas tendrá las opciones para controlar las capas que se generaron o que se añadieron.
Tipo	Primario

Tabla 4.6: Caso de Uso: Mostrar información de los objetos geográficos.

4.6

### 4.3. Consideraciones de Diseño

En el diseño de una aplicación Web, los objetivos de un arquitecto de software son minimizar la complejidad mediante la separación de las tareas en diferentes áreas de preocupación, mientras que el diseño de una aplicación de segura y de alto desempeño. Seguir las siguientes directrices para asegurarse de que la solicitud se ajuste a nuestras necesidades y llevarse a cabo de manera eficiente en situaciones comunes a las aplicaciones Web.

En el diseño de aplicaciones Web, se deben de tener en cuenta las siguientes consideraciones:

- **Particionar la aplicación lógicamente.** Utilizar capas para particionar la aplicación de forma lógica en capas de presentación, negocios y acceso a datos. Ésto le ayuda a crear código mantenible y le permite controlar y optimizar el rendimiento de cada capa por separado. Una clara separación lógica también ofrece más opciones para ampliar su aplicación.

- **Usar la abstracción para implementar el acoplamiento débil entre las capas.** Ésto puede lograrse mediante la definición de componentes de la interfaz, tales como una fachada con entradas y salidas conocidas que traducen las solicitudes en un formato entendido por los componentes dentro de la capa. Además, también se puede utilizar tipos de interfaces o clases de bases abstractas para definir una abstracción compartida que debe ser implementada por los componentes de la interfaz.
- **Entender cómo los componentes se comuniquen entre sí.** Ésto requiere una comprensión de los escenarios de implementación de la aplicación debe apoyar. Debe determinar si la comunicación a través de límites físicos o los límites del proceso debe ser apoyado, o si todos los componentes se ejecutará dentro del mismo proceso.
- **Reduzca viajes de ida y vuelta.** En el diseño de una aplicación Web, considere el uso de técnicas tales como el almacenamiento en el caché y el búfer de salida para reducir los viajes de ida y vuelta entre el navegador, el servidor Web y servidores en sentido descendente.
- **Considerar el uso de almacenamiento en caché.** Una estrategia de almacenamiento en caché bien diseñada es probablemente la consideración más importante de diseño en función del rendimiento.
- **Considere el uso del registro y la instrumentación.** Se debe auditar y registrar las actividades a través de las capas y niveles de la solicitud de registro. Estos registros se utilizan para detectar actividades sospechosas, lo que con frecuencia ofrece los primeros indicios de un ataque contra el sistema.
- **Evitar el bloqueo durante las tareas de larga duración.** Si se tiene larga duración o operaciones de bloqueo, se considera el uso de un enfoque asíncrono para permitir que el servidor Web procese otras peticiones entrantes.
- **Considere la autenticación de usuarios a través de límites de confianza.** Diseñar la aplicación para autenticar a los usuarios cada vez que cruzan la frontera de confianza; por ejemplo, cuando se accede a una capa de negocios a distancia de su capa de presentación.
- **No dejar pasar los datos sensibles en texto plano a través de la red.** Siempre que necesite para pasar los datos sensibles, como una contraseña o una cookie de autenticación a través de la red, considere la codificación y la firma de los datos o el uso de SSL.
- **Diseñar para ejecutar la aplicación Web utilizando una cuenta con privilegios mínimos.** Si un atacante logra tomar el control de un proceso, la identidad del proceso debe tener acceso restringido al sistema de archivos y otros recursos del sistema con el fin de limitar los posibles daños.



## 4.4. Resumen

El capítulo describió el diseño propuesto para el ambiente de visualización, el cual consiste en tres capas lógicas: presentación, negocios y datos. La capa de presentación esta diseñada con HTML, JavaScript, jQuery, AJAX y OpenLayers 3.0, La capa de negocios con Apache y PHP, la capa de datos con PostgreSQL y PostGIS. El diseño de tres capas es muy adecuado para este SIG Web, por que la comunicación entre OpenLayers 3.0 y PostGIS, no es directa y es necesario de un intermediario para obtener datos geográficos y visualizarlos con OpenLayers 3.0. Fue importante tomar en cuenta las consideraciones de diseño y los casos de uso para implementar el SIG Web.



# Capítulo 5

## Implementación

Este capítulo describe la etapa de implementación de la arquitectura SIG propuesta y como se desarrollaron sus componentes por medio de tecnologías de libre distribución, OpenLayers 3.0 el cliente de mapas, PostgreSQL y PostGIS para almacenar los datos geográficos. Cómo se realizó la comunicación cliente-servidor por medio de archivos GeoJSON que son generados por consultas específicas.

### 5.1. Requerimientos

Los requerimientos del sistema se determinaron con base a la arquitectura propuesta del SIG-Web. La arquitectura de ambiente de visualización consiste de tres capas, en la cuál es necesario equipo extra para ser implementado con éxito. Para el desarrollo de esta tesis las tres capas fueron implementadas en un solo equipo, por lo que las tres capas fueron implementadas de manera lógica y no física.

Los requerimientos para la implementación del sistema de visualización son los siguientes:

- Requerimientos de la capa de presentación:
  - Biblioteca jQuery 1.11.2.
  - Biblioteca OpenLayers 3.7.0.
  - CSS (hojas de estilo).
- Requerimientos de la capa de negocios:
  - Servidor Apache.
  - Modulo PHP (PHP 7) para servidor Apache.
  - CSS (hojas de estilo).
- Requerimientos de la capa de datos:

- PostgreSQL 9.5.1.
- PostGIS 2.2.1.
- BD MapasMexico.

## 5.2. Implementación de la capa de datos

Para la implementación de esta capa fue necesario almacenar las cartas topográficas del INEGI en una BD de nombre MapasMexico, la cuál contiene cartas que describen áreas urbanas, caminos, cuerpos de agua, localidades, entre otros datos. La base de datos MapasMexico contiene datos de todo los Estados Unidos Mexicanos, estas cartas están divididas por zonas, las cuales tienen una clave para identificar a cada carta respecto a otras. Los Estados Unidos Mexicanos están divididos en 2313 zonas geográficas de acuerdo a la INEGI [79] y a datos obtenidos de CONABIO [80] Las cartas topográficas están divididas en 96 tablas que contiene información geográfica de diferentes puntos del país.

Las cartas topográficas, contienen datos geográficos que están en coordenadas UTM. En la Figura 5.1 se aprecia la capa de datos obtenida de CONABIO, la cuál contiene todas las zonas geográficas del país. La Figura 5.1 presenta con detalle la división del país es presentado en diferentes zonas geográficas, estas áreas contienen metadatos geográficos con una escala (ver Subsección 2.6.2 ) de 1:50,000.

Cada zona geográfica tiene coordenadas geográficas dependiendo de la zona UTM (ver Subsección 2.6.4) a la que pertenecen, es decir se tiene que tomar en cuenta la localización de las zonas. Como se aprecia en la Figura 5.2, México está compuesto por 6 zonas UTM las cartas geográfica contienen coordenadas geográficas que dependen a la zona UTM a la cuál pertenecen. Además de las zonas geográficas no respetan la división que posee cada Estado, algunos Estados pertenecen a dos zonas UTM, como por ejemplo Veracruz, esta en dos zonas UTM la zona número 14 y la 15. Las cartas topográficas tienen la característica que al utilizar coordenadas UTM, las cuales solo sirven para ubicar información o objetos geográficos corresponde a una única zona UTM.

### 5.2.1. Implementación de la base de datos

Las cartas topográficas están almacenadas en la base de datos MapasMexico, esta contiene un tamaño aproximado de 30 gigabytes, debido a que contiene información detallada de múltiples zonas geográficas de México. Esta información esta dividida en 97 tablas con propiedades espaciales. Por lo que primero se procede a la creación de la base de datos en PostgreSQL , la cuál consistió en los siguientes pasos:

- Creación de una base de datos:

```
./createdb MapasMexico
```



Figura 5.1: Las 2313 zonas Geográficas de los Estados Unidos Mexicanos.

- Se da soporte para el lenguaje procedural pl/psql en la base de datos.

```
./createdb plpgsql MapasMexico
```

- Ejecutamos el script que carga los objetos que convierten a la base de datos normal en una base de datos postGIS.

```
/psql -U postgres -d MapasMexico -f [dirección del script lwpostgis.sql]
```

- Finalmente cargamos la tabla con los sistemas de referencia para ofrecer la funcionalidad espacial a nuestra base de datos.

```
/psql -U postgres -d MapasMexico -f [dirección del script spatial_ref_sys.sql]
```

Con los pasos anteriormente realizados se ha creado la base de datos y esta preparada para ser cargada con datos geográficos. Lo que falta es llenar la base de datos por medio de un *script* que contiene toda la información de las cartas topográficas. Para ejecutar el *script* seguimos los siguientes pasos:



Figura 5.2: Las 6 Zonas UTM de los Estados Unidos Mexicanos.

- Entrar con el usuario postgres.

```
sudo -u postgres psql postgres
```

- Dar los permisos para que se pueda realizar la copia del *script*.

```
sudo chmod a+x /home/usuario/ubicacionScript/  
sudo chmod a+r /home/usuario/ubicacionScript/MapasMexico
```

- Realizamos la copia del script al usuario postgres.

```
sudo cp /home/usuario/ubicacionScript/MapasMexico postgres/  
sudo chown postgres postgres/MapasMexico
```

- Por último se ejecuta el script.

```
sudo i'/home/usuario/ubicacionScript/MapasMexico'
```

Los pasos para ejecutar el *script* se sugieren si el archivo es muy pequeño, pero fácilmente se puede ahorrar pasos solo entrando al usuario postgres y ejecutar la última instrucción. La ejecución exitosa del script llena la base de datos MapasMexico con todas las cartas topográficas.

Nombre	Código	Tipo CRS	Area de Uso Mundo-Hemisferio	Países que cubre
WGS 84 / Zona UTM 11N	32611	Proyectada	120°O a 114°O	Canadá, EUA y México
WGS 84 / Zona UTM 12N	32612	Proyectada	114°O a 108°O	Canadá, EUA y México
WGS 84 / Zona UTM 13N	32613	Proyectada	108°O a 102°O	Canadá, EUA y México
WGS 84 / Zona UTM 14N	32614	Proyectada	102°O a 96°O	Canadá, EUA y México
WGS 84 / Zona UTM 15N	32615	Proyectada	96°O a 90°O	Canadá, EUA, México, Guatemala
WGS 84 / Zona UTM 16N	32616	Proyectada	90°O a 84°O	Canadá, EUA, México, Belice, Costa Rica, Cuba, Ecuador, El Salvador, Guatemala

Tabla 5.1: Proyecciones WGS 84/UTM para México

### 5.2.2. Zonas UTM y geográficas

Los Estados Unidos Mexicanos están divididos en seis zonas UTM, las cuales abarcan una área de uso de 120°O a 84°O o denominadas también como CRS (*Coordinate Reference Systems*). CRS es parte de un modelo más general llamado SRS (*Spatial Reference Systems*), los cuales incluyen referencias por coordenadas e identificadores geográficos. Los identificadores que posee cada una de las seis proyecciones, tienen como nombre SRID (*Spatial Reference System Identifier*), estos son valores únicos usados para identificar definiciones de sistemas de coordenadas. La Tabla 5.1 muestra los códigos de las zonas UTM que pertenece la República Mexicana, el nombre de cada proyección e incluso que países están en el área de uso.

Las zonas geográficas que dividen a la República Mexicana, contienen diferentes datos, una clave, una geometría para desplegar un cuadrado que representa la zona. La Tabla 5.2 muestra índices de ejemplo que contienen las zonas geográficas. GID es un identificador geográfico para distinguir cada elemento geoespacial en una tabla georeferenciada. Siguiendo analizando la Tabla 5.2 es fácilmente apreciable que esta no contiene que estados se encuentran en dicha zona geográfica. Por lo tanto fue necesario crear una nueva tabla *clave\_entidades\_zona*, que contiene información de cada zona geográfica a que Estados y zona UTM pertenece.

GID	Clave
1	I11D67
2	I11D65
3	I11D66
91	H11B37
605	G14A28
1158	G12D74
1494	F14D35
1712	E16A14
2097	E14D28
2313	D15B62

Tabla 5.2: Ejemplo de claves de las 2313 Zonas Geográficas

Para crear la tabla *clave\_entidades\_zona* fue necesario realizar la siguiente consulta (ver Código 5.1). La línea 1 manda a crear una nueva tabla, la línea 2 va hacer una selección de elementos de tres tablas *indi50kgw*, *zona\_utm* y *dest\_2009gw*. La tabla *indi50kgw* contiene toda información de las zonas geográficas en forma de cuadros, la Tabla *zona\_utm* es una tabla que contiene polígonos que describen cada zona utm que compone México y por último la Tabla *dest\_2009gw* tiene la descripción de cada Estado de la República. La línea de mas interés es la 12 por que esta agrupara cada zona geográfica a su estado y zona UTM correspondiente por medio de la función espacial *ST\_Intersects*, la cuál devuelve verdadero si existe alguna intersección de sus elementos geográficos. La Sección 2.4.3, describe SQL Espacial el cual es utilizado para resolver el problema de las zonas UTM, Estados y zonas geográficas.

```

1 CREATE TABLE clave_entidades_zona AS (
2 SELECT
3   a.gid,
4   a.clave,
5   b.zona_utm
6   b.clave as srid
7   c.entidad
8 FROM
9   indi50kgw AS a INNER JOIN
10  zona_utm
11  AS b, dest_2009gw as c ON
12  ST_Intersects(a.geom,b.geom) and ST_Intersects(a.geom,c.geom)
13 );

```

Código 5.1: Consulta SQL para crear la tabla *clave\_entidades\_zona*.

La Tabla *clave\_entidades\_zona* fue creada con el objetivo de facilitar las consultas y la conversión de elementos geográficos de las tablas de la BD *MapasMexico*, por medio de consultas que seleccionen zonas geográficas o estados por medio de la capa de presentación



que sean enviados a la de negocios y generen nuevos datos vectoriales de forma dinámica.

La Figura 5.3 muestra tablas que conforman la BD, que de acuerdo a la subsección 4.1.4, el diseño de nuestro ambiente de visualización solo contempla un numero de 11 tablas que sean accesibles por medio de la capa de negocios. Estas Tablas contienen información de carreteras, lugares, cuerpos de agua, áreas verdes, entre otras. Esta medida se tomo para implementar con rapidez la capa de negocios y de presentación.

### 5.3. Implementación de la capa de negocios

Como se mostró en el capítulo cuatro la arquitectura contiene tres capas, la capa de negocios es la capa encargada de la comunicación entre la capa de presentación y de datos. La capa de negocios fue implementada gracias a la instalación de un servidor Web Apache y de su modulo de PHP que permite el uso de este lenguaje en el servidor. Las reglas de negocios están desarrolladas en PHP, Esta capa contiene los métodos que permiten obtener información geográfica solicitada de la capa de presentación a través de la comunicación que permite PHP con la BD.

#### 5.3.1. Elección de zonas geográficas

El SIG Web tiene como objetivo mostrar información geográfica de zonas geográficas, se hizo principal énfasis en dos formas de selección, la más fácil por medio de la selección de zonas en las que esta dividido México o la selección de estados.

Las cartas topográficas representa diferentes zonas UTM, si deseamos obtener información geográfica de una carta deseada que este contenida en la capa de datos, es necesario seleccionarla por medio de una clave, cada carta tiene una clave única para poderse seleccionar. En el sistema se permite la selección interactiva de la carta topográfica. Por medio de consultas a la base de datos por medio del lenguaje SQL, como se aprecia en el Código 5.2 se aprecia cómo se elige contenido de la tabla y clave seleccionada.

```
1 SELECT the_geom,enti FROM (SELECT ST_Transform(ST_SetSrid(a.geom,b.
    clave_srid), 4326) as the_geom, b.entidad as enti from ".$tabla ." as
    a, clave_entidades_zona as b where lower('".$clave."')=a.idcarta and
    lower(b.clave)=lower('".$clave."') ) g
```

Código 5.2: Selección por medio de claves.

#### 5.3.2. Elección de Estados de la República Mexicana

La selección de zonas geográficas, es muy conveniente para la visualización de pequeñas áreas de estudio, pero para visualizar los datos geográficos es necesario utilizar la siguiente función geográfica que proporciona PostGIS, *ST\_Transform(ST\_SetSrid(a.geom,b.clave\_srid),*

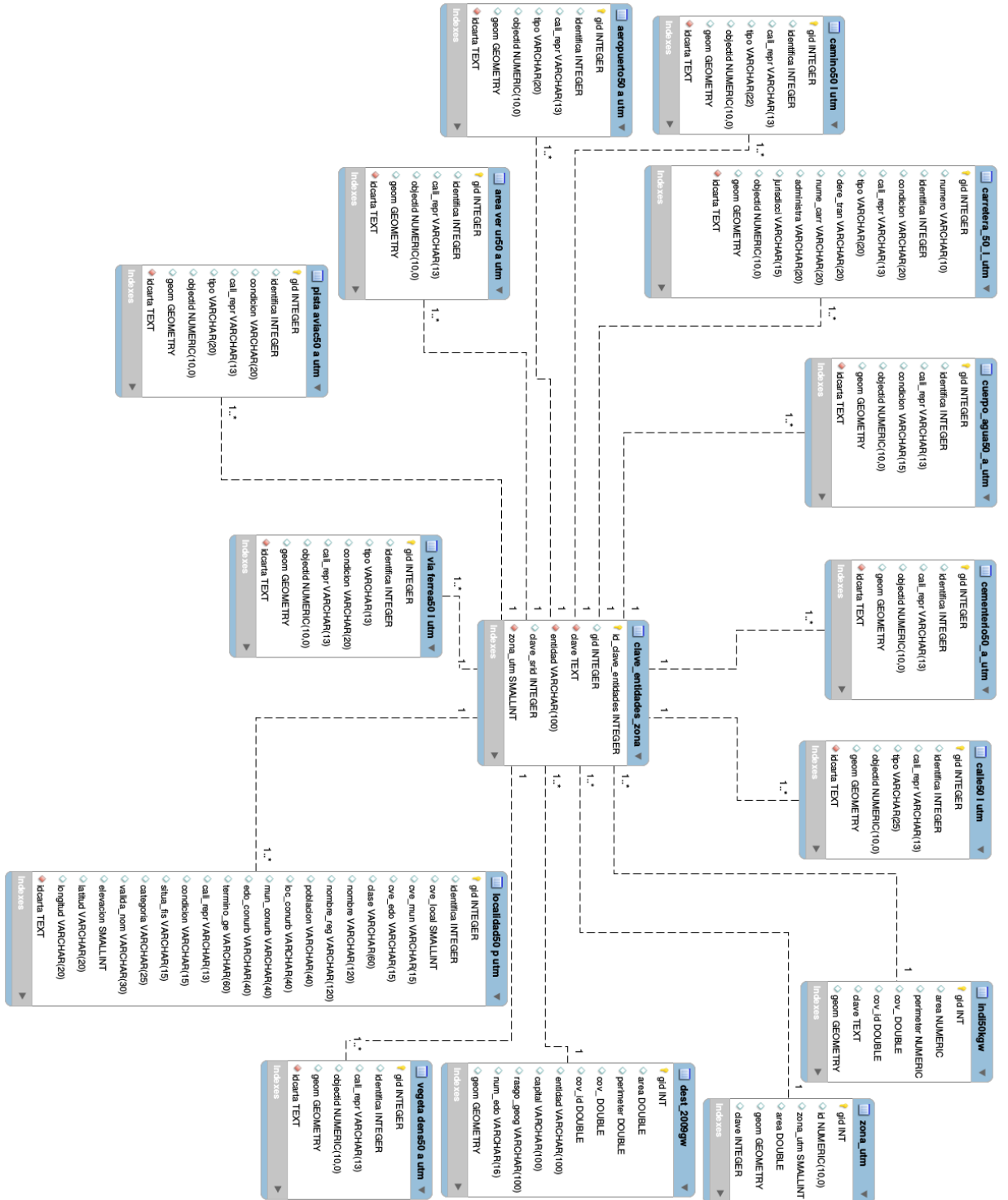


Figura 5.3: BD MapasMexico.

4326) as the\_geom, se esta realizando es la obtención de datos vectoriales, la función *ST\_SetSrid* le da a la geometría (geom) seleccionada un valor conocido como *Spatial Reference System Identifier* (SRID), éste asigna un identificador espacial, éste es de utilidad para el manejo de la información que proviene de las cartas topográficas. Después de asignar el identificador espacial a la geometría, la función *ST\_Transform* realiza la conversión de la geometría al sistema de coordenadas 4326, la cuál es conocida como WGS 84 (ver Subsección 2.6.3, el estándar para América del Norte, al hacer la conversión se toma en cuenta a que zona UTM pertenece de acuerdo a la tabla *clave\_entidades\_zona*.

Para la selección de estados el procedimiento es similar, pero con la gran diferencia que un estado cuenta con varias cartas topográficas. Además, no es necesario elegir elementos que no tengan relación con la el estado seleccionado, por esto fue necesario solo incluir aquellos elementos que sean intersectados en el polígono que represente el Estado. La Figura 5.3, muestra como se realiza la consulta de elementos geográficos.

```
1 SELECT the_geom,enti,gid FROM(SELECT ST_Transform(ST_SetSrid(a.geom,b.
  clave_srid), 4326) as the_geom, a.gid as gid ,b.entidad as enti from
  ".$tabla ." as a, clave_entidades_zona as b, dest_2009gw as c where
  lower(b.clave)=a.idcarta and b.entidad='".$estado.'" and c.entidad=b.
  entidad and ST_Intersects(c.geom,ST_Transform(ST_SetSrid(a.geom,b.
  clave_srid), 4326) )) g
```

Código 5.3: Selección por medio de Estado.

La consulta de elección de estados es muy parecida a la consulta de cartas topográficas, con la gran diferencia que ahora es necesario utilizar uno o varios polígonos que tengan un intersección con los elementos de las cartas, la tabla *dest\_2009gw* representa la división política de los Estados Unidos Mexicanos del año 2009, esta contiene polígonos que representan cada área territorial de México, estados e islas. Los polígonos de la tabla *dest\_2009gw* son utilizados para realizar una función de intersección *ST\_Intersects(c.geom,ST\_Transform(ST\_SetSrid(a.geom,b.clave\_srid), 4326))*, con los datos geográficos de las cartas topográficas para verificar si pertenecen al estado que se desea visualizar.

La consulta SQL para la seleccion de Estados puede ser apreciada de forma grafica en la Figura 5.4, es fácilmente apreciable tres elementos a considerar la capa de estados, zonas UTM y las zonas geográficas. La intersección de las tres capas da como resultado la posibilidad de seleccionar elementos del país a solamente un estado. En este ejemplo se muestra como se genera la capa de carreteras del Estado de Puebla a partir de zonas que intersectan al Estado.

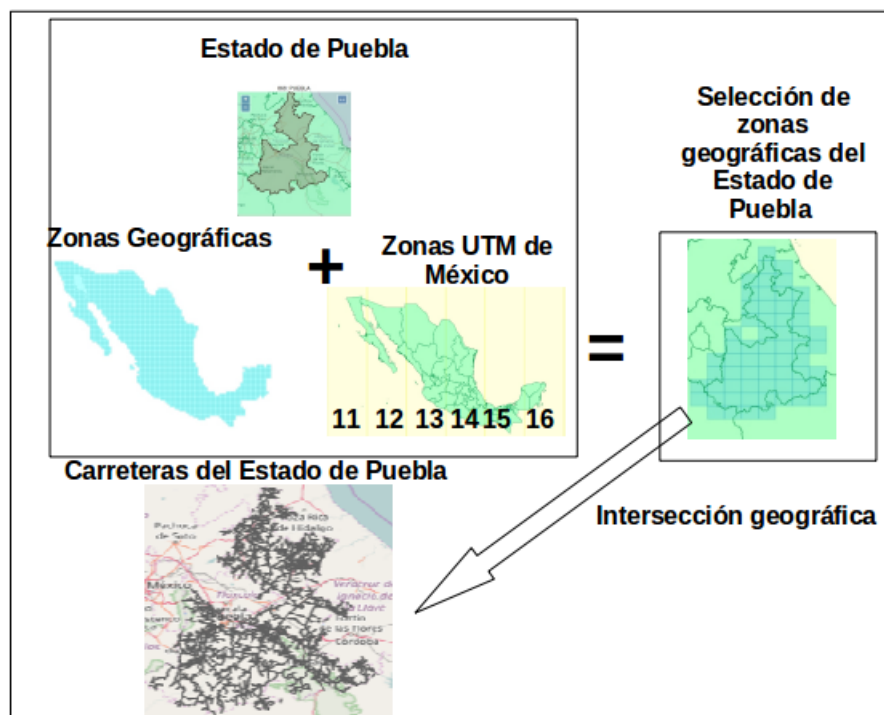


Figura 5.4: Pasos para generar una capa de forma dinámica.

### 5.3.3. Generación de archivos GeoJSON

#### GeoJSON

GeoJSON es un formato para codificación de una variedad de estructuras de datos geográficos para intercambiar estos datos entre diferentes lenguajes. Un objeto de GeoJSON puede representar geometrías, una característica o una colección de características. GeoJSON es un formato de intercambio de datos geoespaciales basado en JavaScript Object Notation (JSON). Un objeto GeoJSON puede representar una geometría, una característica, o una colección de características.

GeoJSON soporta los siguientes tipos geométricos: Point, LineString, Polygon, MultiPolygon, MultiLineString, y MultiPolygon. Los objetos geométricos cuentan con propiedades adicionales como los objetos Feature. Los conjuntos de características están contenidas por los objetos FeatureCollection.

GeoJSON es un formato estándar muy popular para la codificación de información perteneciente a los SIGs. GeoJSON usa puntos de latitud y longitud como sus coordenadas, sin embargo, se requiere cualquier software para encontrar la proyección, la escala, y el método de translación correcta con el objetivo de visualizar los datos.

Una estructura de datos GeoJSON siempre es un objeto (en términos JSON). En GeoJ-

JSON, un objeto consiste de una colección de un par nombre/valor – también es llamado miembros. Para cada miembro, el nombre siempre es un string. Los valores miembro son o bien una cadena, número, objeto, matriz o uno de los literales: true, false, y null. Una matriz se compone de elementos, donde cada elemento es un valor como se describe anteriormente. El apéndice B describe el formato GeoJSON y cada elemento que compone este estándar.

De acuerdo a la arquitectura propuesta, se requieren que los datos geográficos sean transformados al formato GeoJSON, para que estos puedan ser visualizados por OpenLayers. Para la generación de archivos GeoJSON, se utilizan consultas SQL y código en PHP. El Código 5.4 muestra, la consulta SQL completa para la generación de datos geográficos. La línea número uno cuenta con la función `row_to_json`, la cuál interpreta la información consultada en formato JSON, con la gran diferencia que se generara información JSON espacial, por esto es necesario que la información que usualmente obtenemos como filas se agrupe en una colección de características (*FeatureCollection*) por medio de la función `array_to_json(array_agg(f))` .

```

1 SELECT row_to_json(fc)
2           FROM (
3             SELECT 'FeatureCollection'
4             As type, array_to_json(array_agg(f)) As
              features
5             FROM (
6               SELECT 'Feature' As type
7               , ST_AsGeoJSON(
8                 col.the_geom
9               )::json As geometry ,
              row_to_json(lp) as properties
10            FROM (
11              SELECT the_geom,enti,gid
                FROM(SELECT
                  ST_Transform(
                    ST_SetSrid(a.geom,b.
                    clave_srid), 4326) as
                    the_geom, a.gid as
                    gid ,b.entidad as
                    enti from ".$tabla ."
                    as a,
                    clave_entidades_zona
                    as b, dest_2009gw as
                    c where lower(b.clave
                    )=a.idcarta and b.
                    entidad='".$estado.'"
                    and c.entidad=b.
                    entidad and
                    ST_Intersects(c.geom,
                    ST_Transform(

```

```

12                                     ST_SetSrid(a.geom,b.
13                                     clave_srid), 4326) ))
14                                     g
15                                     ) as col INNER JOIN (SELECT ".
16                                     $completa_query.", idcarta
                                     from ".$tabla.") as lp
                                     ON col.gid=lp.gid
                                     ) As f
                                     ) As fc ;

```

Código 5.4: Consulta para generar GeoJSON.

La línea 7, muestra la función `ST_AsGeoJSON`, la cuál permite la conversión de los datos vectoriales obtenidos de las cartas topográficas ser transformadas en formato GeoJSON y permitir su visualización en forma de polígonos, líneas, polilíneas, puntos, entre otros.

PHP es utilizado para la creación de la capa lógica o de negocios, donde se implementa la comunicación entre la capa de presentación y la capa de datos. Los módulos de PHP, proporcionan la conexión con la base de datos MapasMexico, que contiene las diferentes tipos de capas a visualizar. Esta capa incluye la generación de consultas SQL, dependiendo de las peticiones de la capa de presentación realizadas por el usuario. La generación de archivos de tipo GeoJSON son realizados por los módulos PHP, los cuales son solicitados por la capa de presentación por medio de AJAX y jQuery.

La capa de negocios esta conformada por los archivos “`generador_consultas.php`”, `aeropuerto50_a_utm.php`, `area_urbana.php`, `calle50_l_utm.php`, `camino50_l_utm.php`, entre otros. La combinación de estos archivos realiza la comunicación entre la capa de presentación y la capa de datos. El archivo “`generador_consultas.php`”, contiene Código 5.4, y también las instrucciones para la conexión con la BD MapasMexico (ver Código 5.5), el cual utiliza un constructor del `generador_consultas`.

```

1 public function __construct()
2 {
3     $this->connection1 = pg_connect("host=localhost port=5432 dbname
4                             =MapasMexico user=postgres password=postgres")
5                             or print("cant connect");
6 }

```

Código 5.5: Conexión con BD MapasMexico.

```

1 <?php
2 include "../generador_consultas.php";
3 $aeropuertos = new generador_consultas();
4 $estado = $_POST['estado'];
5 $result = $aeropuertos->getMultiPolygon(19,$estado);

```



```
4 </script>
```

Código 5.7: Creación de un mapa con OpenLayers 3.0.

### 5.4.1. Implementación de la interfaz

El diseño de la interfaz del SIG Web, se tomo a consideración la selección por estados y zonas geográficas. La selección por estados, es fácilmente implementable por medio de elementos HTML a partir del elemento *select* y agregándole como opciones los nombres de los estados de la República Mexicana. Pero para el diseño de este sistema, se eligió una forma más interactiva. La creación de dos mapas OpenLayers 3.0, uno para visualización de las cartas geográficas y otro para la selección del estado o de la zona geográfica.

Para desplegar dos mapas en una pagina Web fue de una manera simple. Primero se declararon dos elementos div de clases map. Segundo. se le asigno a cada mapa un identificador, con el fin de controlar cada uno por medio de JavaScript y jQuery. El código 5.8 muestra la declaración de los dos mapas, para que OpenLayers pueda interpretarlos como mapas.

Las coordenadas de un objeto Polygon son una matriz de coordenadas.

```
1 <div id="map" class="map"></div>
2 <div id="map1" class="map"></div>
```

Código 5.8: Dos Mapas HTML.

La interfaz resultante es muy parecida al diseño propuesto anteriormente en el capítulo 4, el cuál utiliza dos mapas el primero para la visualización y el segundo para la selección de zonas estados. Se implementaron dos tipos de interfaz. La primera se enfoca en estados de la República Mexicana (5.7), y la segunda interfaz se enfoca en la selección de zonas geográficas (5.8). Ambas interfaces son parecidas, y proporcionan una experiencia de navegación similar.

En la Sección 4.1.2, se propuso un diseño de dos mapas, donde “map” es el elemento donde se desplegara un mapa que sirve para la visualización de capas vectoriales, contiene herramientas del SIG. El elemento “map1” es declarado para ser el mapa de navegación, el cual permite la selección de zonas geográficas o Estados, en el Código 5.9 se aprecia como esta conformado los dos mapas creados, los dos son casi iguales por eso no ponemos la declaración de “map1”. Las leyendas de esta capa son mostradas con el color que le corresponde a cada una. La capa base de los dos mapas es obtenida de OpenStreetMap, para ver si la posición de las capas vectoriales corresponde la declaración de una capa, se aprecia en la línea 16 como se inicializa una capa que tiene como fuente a OSM.

La línea 2 inicializa los controles que dispone nuestro mapa, la línea 7 define un nuevo control que fue creado para almacenar las herramientas de medición y anotación. La instrucción target de la línea 10, permite al navegador entender en cual elemento HTML se debe crear el mapa correspondiente. Es visible la instrucción EPSG(European Petroleum



Survey Group), pero solo es otra forma de llamar al SRID. Por último la línea 28 muestra la creación de una vista, donde se declara que tipo de coordenadas maneja el mapa, el limite de ampliación del mapa.

```
1  mapa = new ol.Map({
2      controls: ol.control.defaults({
3          attributionOptions: ({
4              collapsible: false
5          })
6      }).extend([new ol.control.FullScreen(),
7          new app.CustomToolbarControl()
8      ])
9
10     ,
11     target: 'map',
12
13     layers: [ new ol.layer.Group({
14                 'title': 'Base maps',
15                 'name': 'Mapas Base',
16                 layers: [
17                     new ol.layer.Tile({
18                         title: "OSM",
19                         name: "osm",
20                         type: 'base',
21                         visible: true,
22                         source: new ol.source.OSM()})
23                 ]
24             }),
25             layerGroupMediciones,
26             layerGroupAnotaciones
27         ],
28     view: new ol.View({
29         center: ol.proj.transform([ -100.622635, 23.523569], 'EPSG:
30             :4326', 'EPSG:3857'),
31         zoom: 5,
32         minZoom: 4
33     })
34 });
```

Código 5.9: Creacion de “map” en OpenLayers.

## OpenStreetMap

OpenStreetMap (OSM) es un proyecto internacional fundado en 2004 con el objetivo de crear un mapa libre del mundo. Los datos son recopilados por ciudadanos entusiastas interesados en compartir datos geográficos de forma libre y abierta ya sea para ser utilizados por individuos, organizaciones, empresas o gobiernos. OpenStreetMap es un mapa hecho por todos y para todos [81]. El mapa libre de OSM permite recolectar datos que

pueden ser desde carreteras, vías férreas, ríos, bosques, selvas, edificios y todo lo que te puedas imaginar. OpenStreetMap es un proyecto de colaboración que tiene como objetivo crear un mapa libre y abierto del mundo gracias a la labor de los voluntarios con GPS asistido, conocedores de sus barrios, ciudades, imágenes aéreas, la recolección así como la publicación de los datos geográficos públicos.

OSM es utilizado en este SIG-Web por medio de OpenLayers como una capa de tipo *tiled*(mosaico). Esta capa es utilizada como fondo del mapa, el cual sirve para mostrar al usuario el lugar donde esta navegando. OSM sirvió para mostrar que las capas vectoriales generadas por la selección de estados o zonas, están ubicadas en el lugar correspondiente. La Figura 5.6 muestra a OpenStreetMap en funcionamiento, de esta imagen es fácilmente visible parte de la República Mexicana.

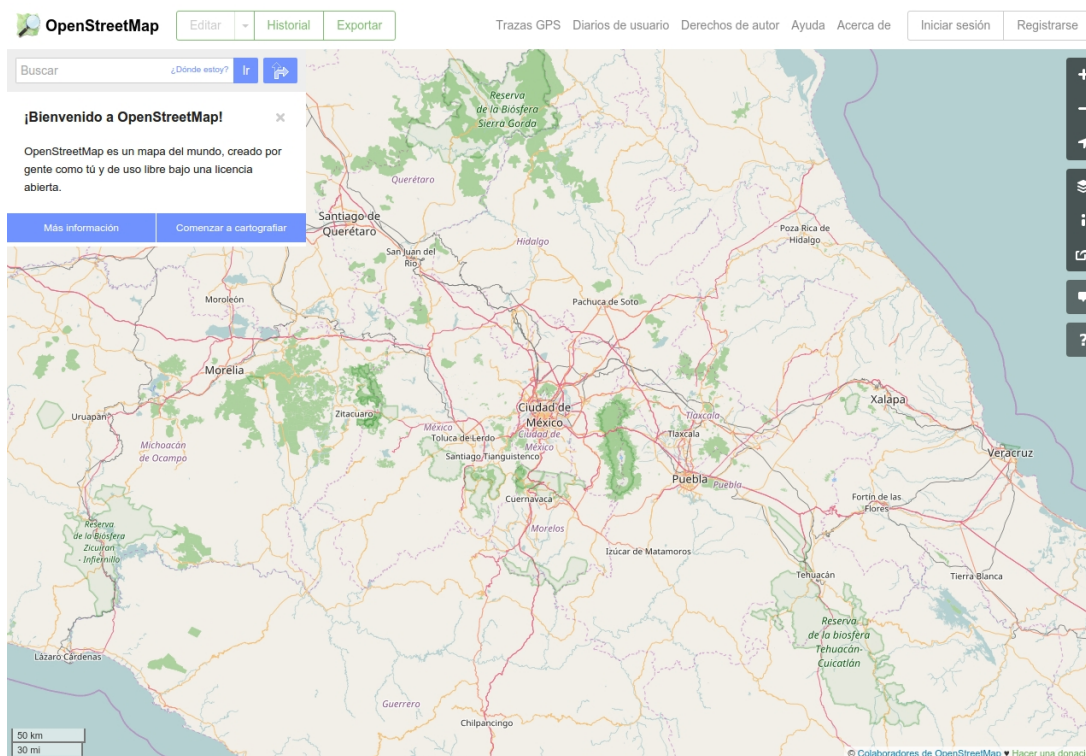


Figura 5.6: OpenStreetMap en funcionamiento.

Siguiendo del diseño de la interfaz propuesto en la subsección 4.1.2, se creó por medio de HTML los elementos de cada parte propuesta y se agregaron otras partes que son de gran utilidad como los formularios de agregar capa vectorial, categorizar capas, graduar capas. El menú de control de capas es creado por el Código 5.10, el cual solo muestra un elemento `div` que pertenece a la clase “tree” o árbol, como indica su nombre “layertree”, éste contendrá los elementos de control para mostrar u ocultar las capas que se tengan en la interfaz.

```
1 <div id="layertree" class="tree"></div>
```

Código 5.10: Creación del menú de control de capas.

El menú de selección de capas, tiene la función de permitir al usuario la selección de capas que desea añadir al mapa de visualización. Este menú es creado por medio de elementos HTML tipo “checkbox”, el sistema detecta cuando se da clic en el botón Agregar cuales capas se quiere desplegar y cuales no. El sistema puede añadir nuevas capas después de añadir anteriormente. El funcionamiento de este menú utiliza jQuery y JavaScript para su adecuado funcionamiento.

```
1 <div id="right">
2     <h3>Seleccion de Capas</h3>
3     <input type="checkbox" class="single-checkbox" name="option"
4         id="option1" value="Capa carretera50_l_utm">Capa
5         carretera50_l_utm<br>
6     <input type="checkbox" class="single-checkbox" name="option"
7         id="option2" value="Capa area_urb50_a_utm" > Capa
8         area_urb50_a_utm<br>
9     <input type="checkbox" class="single-checkbox" name="option"
10        id="option3" value="Capa cuerpo_agua50_a_utm">Capa
11        cuerpo_agua50_a_utm<br>
12    <input type="checkbox" class="single-checkbox" name="option"
13        id="option4" value="Capa cementerio50_a_utm">Capa
14        cementerio50_a_utm<br>
15    <input type="checkbox" class="single-checkbox" name="option"
16        id="option5" value="Capa aeropuerto50_a_utm">Capa
17        aeropuerto50_a_utm<br>
18    <input type="checkbox" class="single-checkbox" name="option"
19        id="option6" value="Capa area_ver_ur50_a_utm">Capa
20        area_ver_ur50_a_utm<br>
21    <input type="checkbox" class="single-checkbox" name="option"
22        id="option7" value="Capa calle50_l_utm">Capa
23        calle50_l_utm<br>
24    <input type="checkbox" class="single-checkbox" name="option"
25        id="option8" value="Capa camino50_l_utm">Capa
26        camino50_l_utm<br>
27    <input type="checkbox" class="single-checkbox" name="option"
28        id="option9" value="Capa pista_aviac50_a_utm">Capa
29        pista_aviac50_a_utm<br>
30    <input type="checkbox" class="single-checkbox" name="option"
31        id="option10" value="Capa via_ferrea50_l_utm">Capa
32        via_ferrea50_l_utm<br>
33    <input type="checkbox" class="single-checkbox" name="option"
34        id="option11" value="Capa vegeta_dens50_a_utm">Capa
35        vegeta_dens50_a_utm<br>
```

```

17     <input type="checkbox" class="single-checkbox" name="option"
18         id="option12" value="Capa localidad50_p_utm">Capa
19         localidad50_p_utm<br>
20     <button id="boton" type="button" onclick="">Agregar</button>
21     <br>
22     <button id="botonRem" type="button" onclick="">Remover</
23         button>
24 </div>

```

Código 5.11: Creación del menú de selección de capas.

Los formularios de agregar capa vectorial, categorizar capas, graduar capas se crearon utilizando los formularios de HTML para permitir al usuario llenar los diferentes campos que esto conlleva. En el Código 5.12 se muestra el formulario de categorización de capas el cual permite al usuario elegir la capa y que elemento de la capas se va a categorizar. La categorización se realiza de una forma sencilla por medio de JavaScript, el cual consiste que a cada elemento de una capa que sea diferente se le asigna un color de forma que no se repita a los elementos anteriores.

```

1 <div id="categorizar" style="display: none;">
2     <br>
3     <form id="categorizar__form" class="categorizar_class">
4         <p>Seleccione la capa vectorial a categorizar </p>
5         <table>
6             <tr>
7                 <td>Capa Vectorial:</td>
8                 <td><select id="txtCapaVectorial" name="
9                     CapaVectorial" required="required">
10
11                 </select></td>
12             </tr>
13             <tr>
14                 <td>Campo a categorizar:</td>
15                 <td><select id="txtCampoCategorizar" name="
16                     CampoCategorizar" required="required"></
17                 select></td>
18             </tr>
19             <tr>
20                 <td><input id="btnCategorizar" type="button"
21                     value="Categorizar"></td>
22                 <td><input id="btnCategorizarCancel" type="
23                     button" value="Cancelar"></td>
24             </tr>
25         </table>
26     </form>
27 </div>

```

Código 5.12: Formulario de Categorización de capas.

Ya explicado el diseño y casi toda la implementación de las tres capas lógicas, especialmente la implementación de la capa de presentación, ésta esta conformada por diferentes partes como los mapas de navegación y visualización, el menú de control decapas y el menú de selección de capas. La Figura 5.7 muestra el ambiente de visualización Web en funcionamiento, se aprecia que las capa visualizadas son del Estado de Tlaxcala. En el mapa de visualización se incluye un menú de herramientas para anotar y medir. En el mapa de navegacion que se distingue por ser mas pequeño que el mapa de visualización contiene una tabla con las leyendas para distinguir que capa es por medio de colores. La parte que tiene el titulo “Capas” es el menú de control de capas y muestra las capas vectoriales que están visualizándose o fueron añadidos al mapa de visualización. El menú de selección de capas tiene 12 capas a elegir. Estas partes también son observables en la Figura 5.7, con la gran diferencia en el mapa de navegación y visualización, donde se selecciona zona geográficas. La Figura 5.8, muestra una zona que pertenece también al Estado de Tlaxcala.

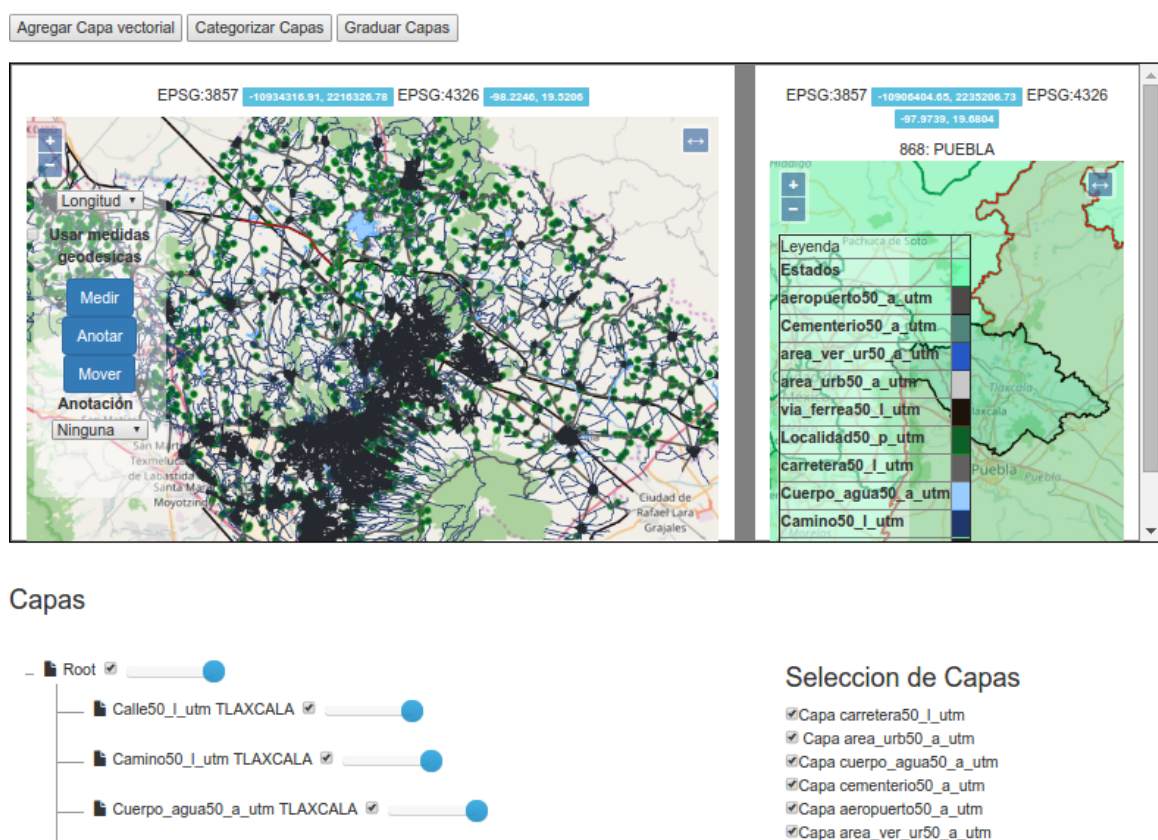


Figura 5.7: Interfaz de selección de estados.

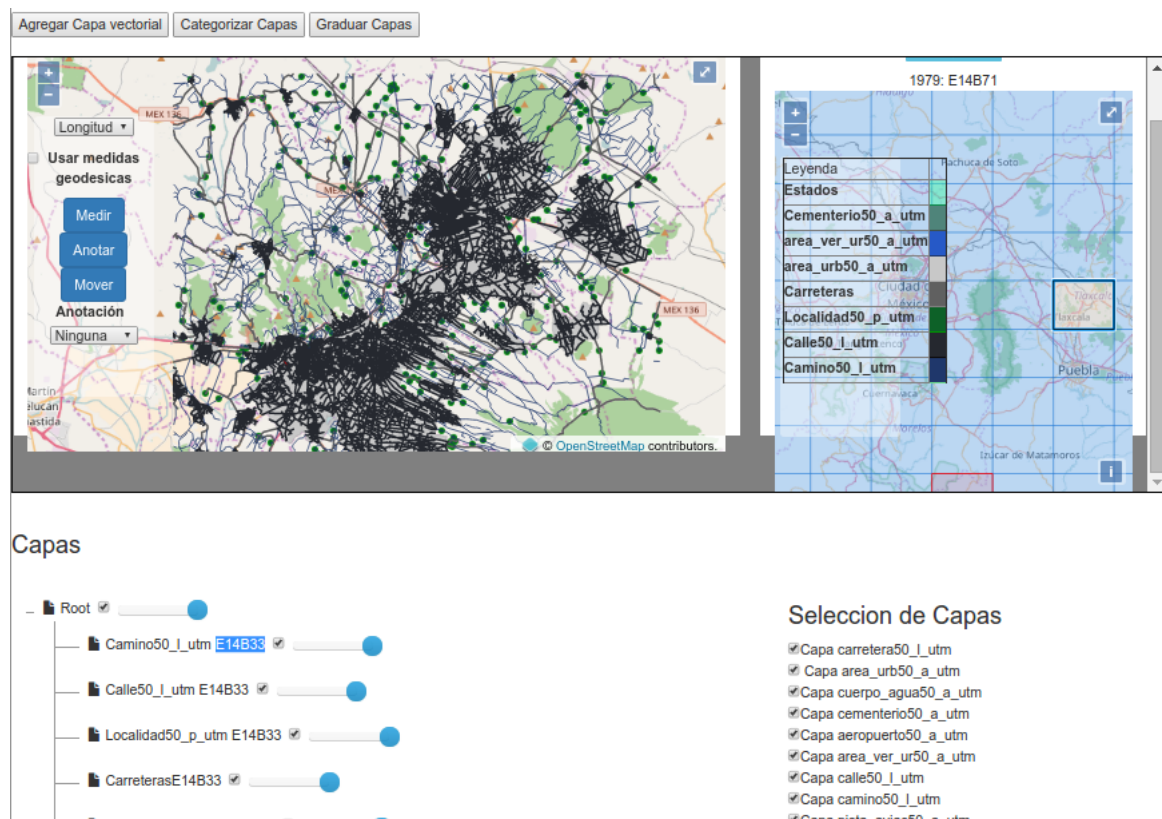


Figura 5.8: Interfaz de selección de zonas geográficas.

El Código 5.13 permite la categorización (código basado en [75]) en una capa por medio de sus datos nominales, la función recibe los parametros de “layer” y “attribute”. Las líneas 2 y 3 son arreglos para guardar los atributos de cada capa y el color que le corresponderá a cada atributo. De la línea 5 a la 16 son las instrucciones donde se realiza la obtención de los atributos y se le asigna un color de forma aleatoria. Por ultimo, de la líneas 18 a la 37 se declara el estilo que llevara la capa vectorial de acuerdo a sus atributos.

```

1 function categorizarCapa(layer, attribute) {
2   var attributeArray = [];
3   var colorArray = [];
4   var randomColor;
5   layer.getSource().forEachFeature(function (feat) {
6     var property = feat.get(attribute).toString();
7     if (attributeArray.indexOf(property) === -1) {
8       attributeArray.push(property);
9       console.log(property);
10      do {
11        randomColor = randomHexColor();
12      } while (colorArray.indexOf(randomColor) !== -1);

```

```
13         colorArray.push(randomColor);
14     }
15     }, this);
16
17     layer.setStyle(function (feature, res) {
18         var index = attributeArray.indexOf(feature.get(attribute).
19             toString());
20         var style = new ol.style.Style({
21
22             image: new ol.style.Circle({
23                 radius: 3,
24                 fill: new ol.style.Fill({
25                     color: colorArray[index]
26                 })
27             }),
28             stroke: new ol.style.Stroke({
29                 color: colorArray[index],
30                 width: 2
31             }),
32             fill: new ol.style.Fill({
33                 color: colorArray[index]
34             })
35         });
36         return [style];
37     });
38 }
```

Código 5.13: Categorización de capas.

## Generación de consultas en la capa de presentación

La generación de consultas necesita información de la capa de presentación, con la selección de Estados o zonas geográficas y las capas que se deseen visualizar. Las funciones para generar las capas de forma dinámica son casi exactas, por lo tanto no hace falta mencionar las demás. La función “funciongetDataCamino” (ver Código 5.14, recibe los parámetros mapas y Estado, la opción de la selección por zonas solo cambia Estado por zona, pero la mayor diferencia de es en la capa de negocios, donde las consultas son diferentes. Siguiendo con la descripción de la función getDataCamino50\_l\_utm, la línea 4 corresponde a la dirección URL al archivo php, encargado de llamar al archivo generador\_consultas.php para que realice la comunicación entre las tres capas presentación, negocios y datos.

```
1 function getDataCamino50_l_utm(map, estad)
2 {
3     var url = './php/camino50_l_utm.php';
4     var est=estad;
```

```

5     xhr[7]=$.ajax({
6
7         url:    url,
8         data: {"estado": est},
9         type:  'post',
10        success: function (response)
11        {
12            layers[7] = cargarGeoSonMultipolygon(map,response,"
13                Camino50_1_utm "+est,strokeCamino50_1_utm);
14            addLayers(map,layers[7], function()
15            {
16                setTimeout(inicializaLayerTree, 50);
17                console.log("creado");
18                $('#mtable tbody').append('<tr id="idfila"+row+> <td><
19                    label>Camino50_1_utm</label></td><td id="idcol'+row+'
20                    "><label></label></td><tr>');
21
22                document.getElementById('idcol'+row).style.
23                    backgroundColor = strokeCamino50_1_utm;
24                document.getElementById('idcol'+row).style.
25                    borderColor = fillCamino50_1_utm;
26                row++;
27            });
28        });
29    });
30 }

```

Código 5.14: Función getDataCamino para generar consultas en la interfaz.

La línea 6 es donde se pide el uso de el motor AJAX proporciona la capacidad de realizar peticiones y respuestas de forma asíncrona, sin esta opción no se puede realizar el enlace entre la capa de presentación y de negocios. De las líneas 10 a 25 cargo datos de el servidor por medio de una petición “POST HTTP”. La línea 11 indica si la petición fue exitosa, entonces realiza las instrucciones de la líneas 12 a la 25. La línea 13 realiza una función que crea la capa vectorial a partir de los datos obtenidos de la BD MapasMexico en formato GeoJSON. La línea 14 a la 24 se agrega una nueva capa al mapa de visualización. El Código anteriormente explicado es el mas importante por que con el se hace la petición a la capa de negocios (ver subsección 5.3.3).

## 5.5. Resumen

Este capítulo mostró la implementación del diseño de tres capas propuesto para el ambiente de visualización, se enuncio los requerimientos, como se creo la capa de datos a partir de una BD MapasMexico contenida en un *script*, esta BD necesito de nuevas tablas para poder convertir las coordenadas geográficas de los elementos geográficos a uno en



común, el WGS84. La implementación de la capa de negocios se realizó para la selección de Estados y zona geográficas de México por medio del lenguaje SQL. Por último la capa de presentación realizada con OpenLayers 3.0 para visualizar las capas vectoriales a partir de datos vectoriales provenientes de la capa de negocios y el uso de AJAX para poder realizar la consulta al servidor. La Figura 5.9 muestra el funcionamiento de los componentes implementados del ambiente de visualización.

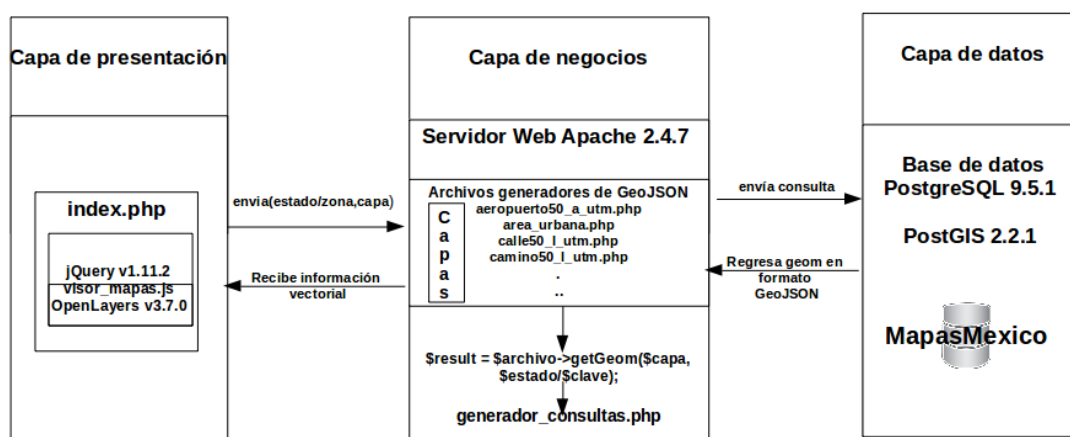


Figura 5.9: Arquitectura Final del ambiente de visualización.



# Capítulo 6

## Validación y pruebas al SIG Web

En este capítulo, se describe las diferentes pruebas que se realizaron SIG Web con diferentes capas geográficas. La primera parte muestra validaciones de la generación de mapas por medio archivos GeoJSON creados a partir de consultas SQL para diferentes Estados de la República Mexicana, la primera validación se utiliza la capa camino50\_lutm de diferentes Estados. La segunda validación es realizada al Estado de México y a estas capas de este Estado se le realizó la función de categorización. Por último se analizan estos resultados y se termina con una comparativa entre los diferentes SIG Web estudiados en el capítulo 3.

### 6.1. Primera Validación

En esta sección se presenta una prueba a diferentes Estados para generar una sola capa, la capa seleccionada es camino50\_lutm, la cual contiene una gran cantidad de elementos geográficos de tipo MultiLine. Los Estados que fueron seleccionados para esta validación cuentan con diferentes tamaños por ejemplo el Estado de Colima es muy pequeño y el Estado de Veracruz es muy extenso. Cada Estado de la República Mexicana cuenta con una amplia cantidad de elementos geográficos de tipo camino50\_lutm. En esta misma sección se abarca la validación del ambiente de visualización, la selección por zonas geográficas, las cuales cuentan con menos área que cualquier otro Estado.

La Figura 6.1, muestra la capa vectorial camino50\_lutm en cuatro Estados diferentes Veracruz(a), Zacatecas (b) Puebla (c), Colima (d). En el Estado de Veracruz es fácilmente apreciable, un cuadrado que no contiene ningún elemento de tipo camino50\_lutm, esto se debe a que esa carta actualmente no está contenida en la BD MapasMexico. Haciendo énfasis en los Estados de Veracruz, Zacatecas y Puebla es fácilmente apreciable la formación de la capa vectorial de acuerdo a la forma del Estado. En contraste, el Estado de Colima no es tan apreciable la cantidad de elementos geográficos, debido al tamaño del Estado y de los pocos elementos geográficos con los que cuenta.

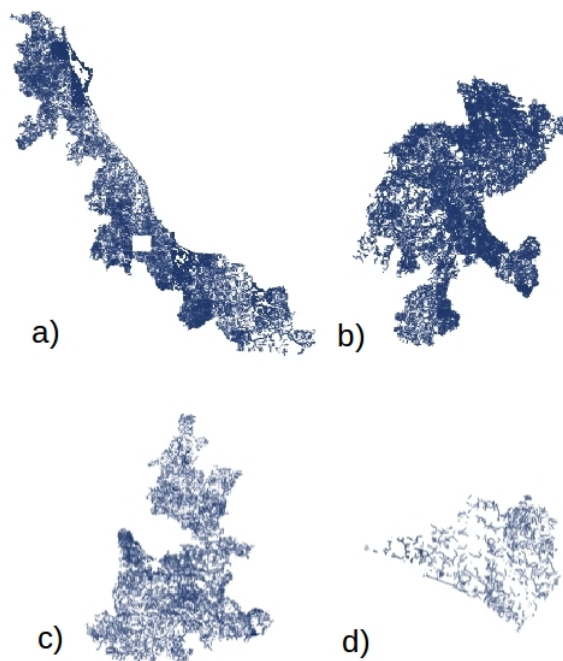


Figura 6.1: Despliegue de la capa camino50\_Lutm en cuatro Estados. (a) Veracruz , (b) Zacatecas, (c) Puebla y (d) Colima

La validación del prototipo por selección de zonas, consistió en probar la visualización de tres zonas diferentes de la República Mexicana. Las capas seleccionadas para la validación por zonas fueron la capas de carretera50\_Lutm y la capa localidad50\_p\_utm, éstas contienen elementos en casi toda la República, por lo que son capas ideales para probar el ambiente de visualización por selección de zonas. Las zonas seleccionadas para la validación fueron : (a)E14B49 zona ubicada en el Estado de Veracruz específicamente en los municipios de Veracruz y Boca del Río. (b)E14B43 la zona visualizada contiene a la ciudad de Puebla, Puebla y (c)G14C26 contiene a la ciudad de Monterrey , Nuevo León (ver Figura 6.2). Se usaron estas zonas reconocidas para que se aprecie junto con la capa de OSM para verificar que se crearon en la zona seleccionada.

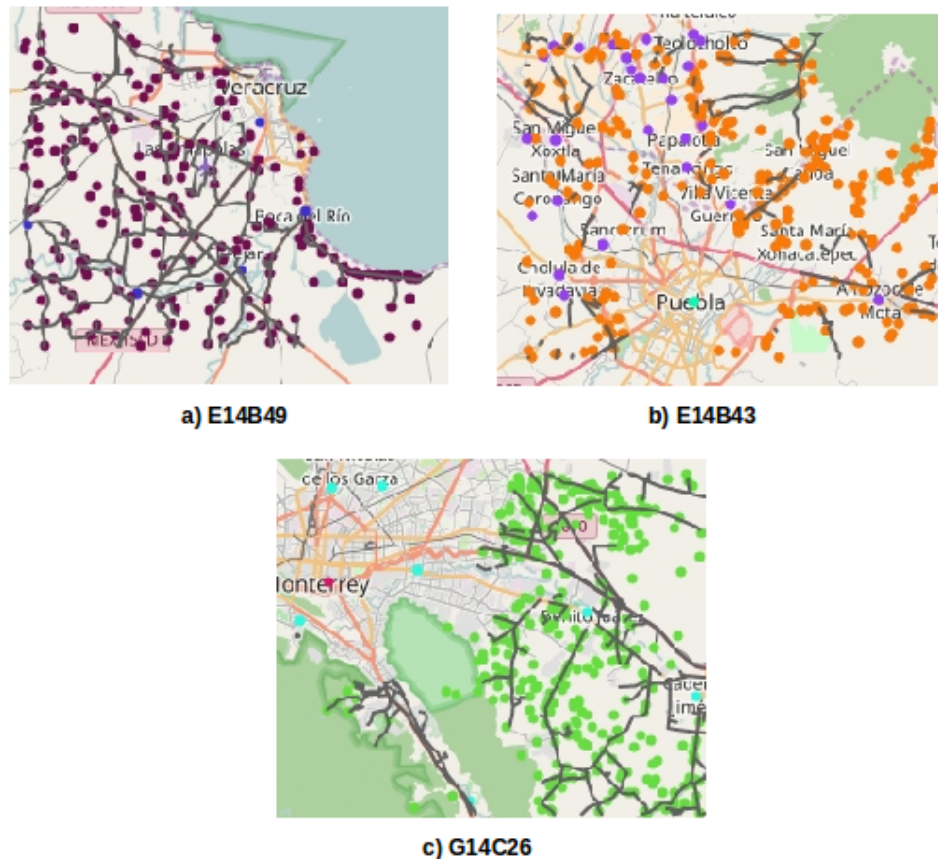


Figura 6.2: Pruebas de visualización a tres diferentes zona geográficas. (a)E14B49 Veracruz, (a)E14B43 Puebla, (c)G14C26 Monterrey.

## 6.2. Segunda Validación, Caso de Estudio: Estado de México

Para la segunda validación del SIG Web, se realizó un caso de estudio a un Estado para mostrar el despliegue de 12 capas que cuenta este prototipo SIG Web. El Estado del caso de estudio es el Estado de México, Estado de interés por tener gran cantidad de elementos geográficos.

La Figura 6.3 muestra las diferentes capas del Estado México, algunos polígonos de la capa `area_urb50_a_utm` muestran zonas correspondientes de la Ciudad de México esto se debe a la utilización de la función de intersección y no la función de diferencia que se mencionaron en el capítulo 5.

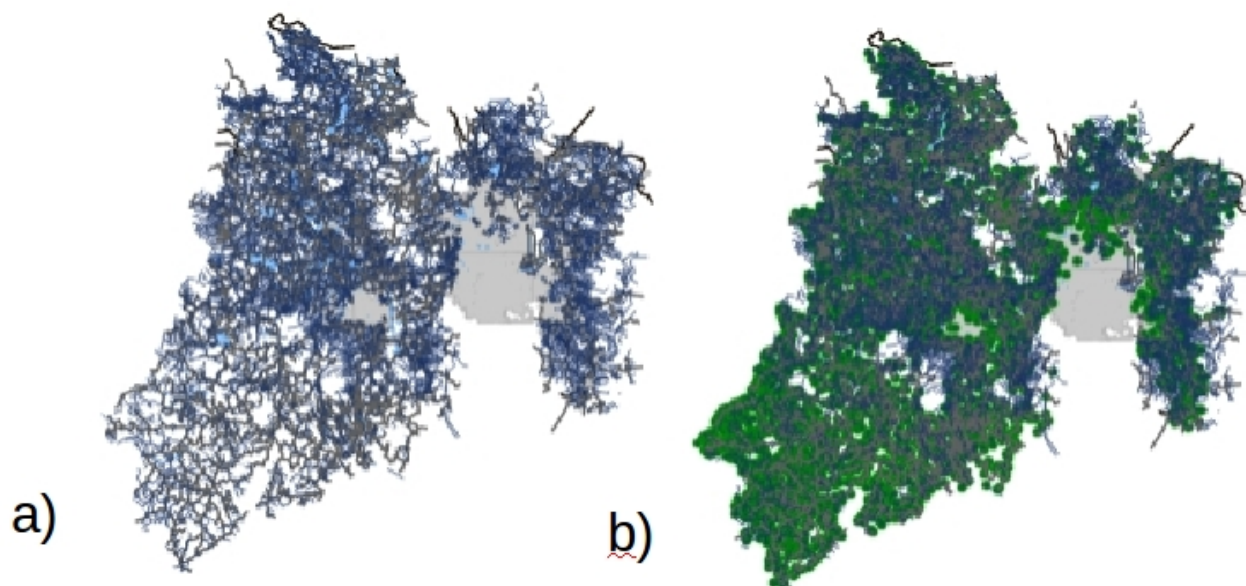


Figura 6.3: Visualización del Estado de México con 11 capas vectoriales. a) Sin la capa localidad50\_p\_utm, b) Con la capa localidad50\_p\_utm.

Al observar la Figura 6.3, solo se visualizo 11 capas, la capa calle50\_l\_utm no se pudo visualizar por que a diferencia de las demás capas cuenta con una gran cantidad de objetos geográficos. En (a) se visualiza solo 10 capas por que se desactivo la capa localidad50\_p\_utm, que no permitía apreciar las otras capas con claridad, en (b) se puede apreciar localidad50\_p\_utm por medio de puntos de color verde. Para este caso especial se desarrollo un método de diferente, la selección por zonas geográficas, este tipo de selección permite la visualización por zonas y esto resulta en un mejor desempeño a la hora de visualizar. Para demostrar lo anterior en la Figura 6.4 se muestra la selección de una zona del Estado de México.

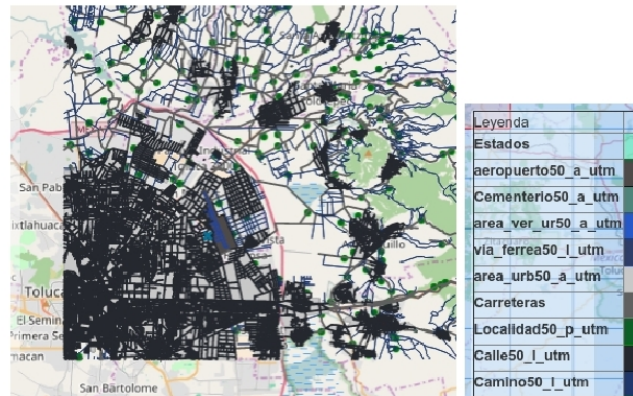


Figura 6.4: Visualización de una Zona del Estado de México con 10 capas vectoriales.

La Zona del Estado de México tiene por clave E14A38, en esta zona se encuentran elementos pertenecientes a la Capital del Estado, la ciudad de Toluca, donde ya es apreciable la capa de calle50\_l\_utm, solo son observables 10 capas a causa de que no existen elementos de vegetación en esta zona (vegeta\_dens50\_a\_utm, area\_ver\_ur50\_a\_utm). Muchas de de las capas no se pueden generar por que no todas las zonas geográficas o Estados cuentan con elementos geográficos ahí.

### 6.2.1. Validación de la función de categorización

Para validar las funciones de categorizar capas vectoriales, siguiendo el mismo caso de estudio, se selecciono la capa de camino50\_l\_utm y carretera50\_l\_utm, para categorizarlas y a la capa localidad50\_p\_utm. La función de categorizar se realiza con base en los datos nominales que contienen las capas vectoriales. Estas funciones son importantes para interpretar los datos de forma visual.

Primero se le realizara la categorización a la capa carretera50\_l\_utm a su campo nominal "jurisdicci", este campo señala si la jurisdicción de la carretera es de tipo Estatal(Edo.Mex), Federal, Ninguna o Indeterminada. Al categorizar con base a esto se realizara la coloración de los objetos geográficos de acuerdo a la jurisdicción que contenga cada objeto. La Figura 6.5, muestra la categorización obtenida.

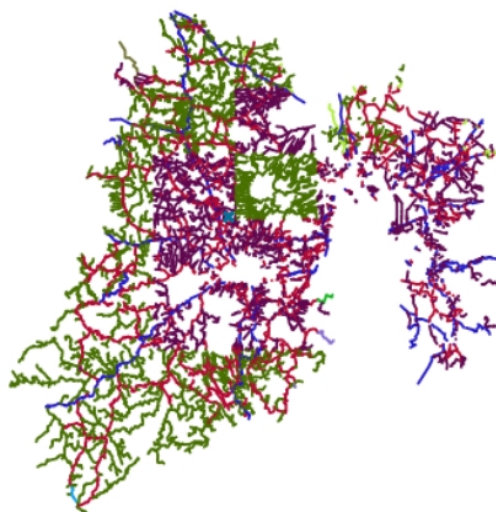


Figura 6.5: Categorización de la capa `carretera50_lutm` a su campo “jurisdicci” (jurisdicción).

Para la siguiente prueba a la función de categorización se eligió la capa `camino50_lutm` y su campo tipo el cual identifica el tipo de camino si es una brecha o una vereda. La Figura 6.6, se aprecia la coloración de la capa, y se nota que existen miles de elementos de la `camino50_lutm` y con la categorización podemos notar que el Estado de México solo tiene dos tipos de camino: brecha y vereda.

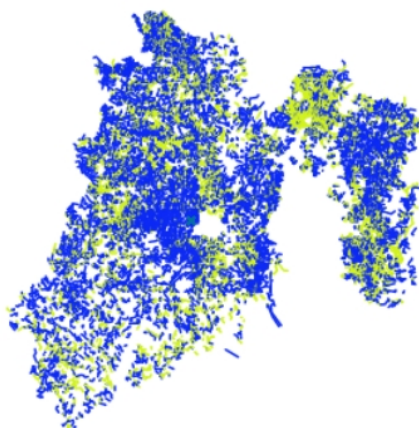


Figura 6.6: Categorización de la capa `camino50_lutm` a su campo tipo.

La última prueba de esta sub sección, se utilizara la capa `localidad50_p_utm` al campo de, este campo categoría se eligió debido a su características de indicar que tipo de localidad, si es Cabecera Municipal, Capital del Estado u Otro. La figura 6.7 muestra el resultado de la categorización y como cada punto en el mapa se coloreo dependiendo



de su categoría como localidad. Con esta prueba y las dos anteriores, realizadas a tres diferentes capas vectoriales, se probó satisfactoriamente la función de categorización.

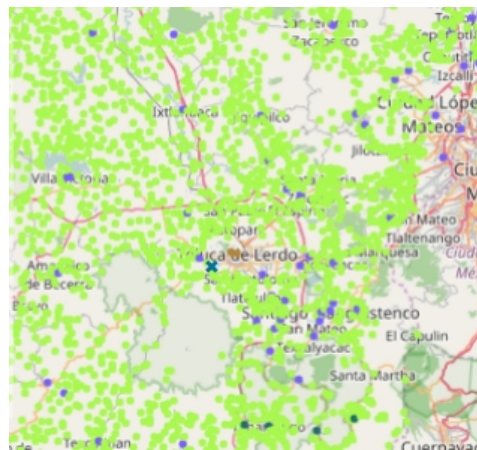


Figura 6.7: Categorización de la capa localidad50\_p\_utm a su campo categoría.

### 6.3. Tercera Validación

Para la última validación del sistema, se utilizaron dos capas vectoriales no incluidas en la base de datos MapasMexico. Las dos capas vectoriales son obtenidas a partir de diferentes formatos de archivos, GeoJSON y KML. EL primer formato de archivo es reconocido por la OGC y el segundo es muy utilizado por aplicaciones como GoogleMaps y Google Earth. Esta validación tiene como finalidad mostrar que el SIG Web puede visualizar diferentes tipos de formatos: KML y GeoJSON.

Para la primera capa vectorial es de tipo KML, con el nombre de muni\_2015gw.kml, esta contiene todos los municipios que integran a la República Mexicana, a esta capa no se le puede realizar la función de categorización debido a que es un formato diferente a GeoJSON. Los municipios visualizados son mostrados en la Figura 6.8. Los municipios son mostrados con color rojo e incluso se muestra las islas que conforman al país. Esta capa no permitió la función de categorización a sus municipios.



Figura 6.8: Capa de municipios obtenida de un archivo KML.

La segunda capa vectorial es de formato GeoJSON, el nombre de esta capa es clima1mgw, esta contiene los diferentes climas que predominan en zonas de la República Mexicana. A esta capa se le aplicó la categorización para observar los diferentes tipos de clima de forma visual (Fig. 6.9). Los climas no son señalados por un nombre sino por una simbología específica para los climas.

Las capas utilizadas para esta validación fueron obtenidas del sitio de CONABIO [80]. Este sitio cuenta con cientos de capas que se pueden incorporar al sistema si se convierten al formato GeoJSON o KML.

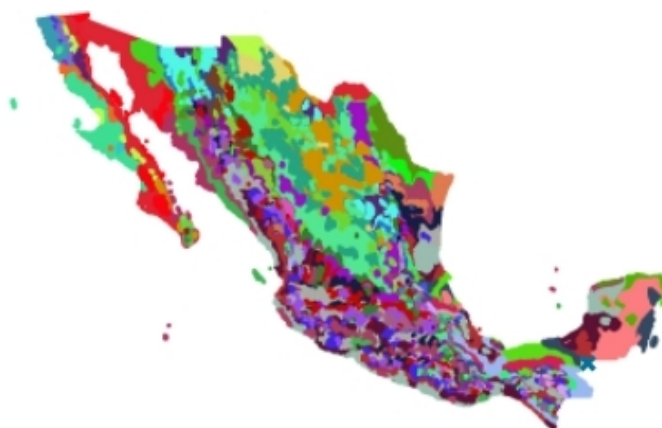


Figura 6.9: Capa de climas obtenida de un archivo GeoJSON.

## 6.4. Análisis de resultados

A partir de las validaciones realizadas se obtuvieron los siguientes resultados:

- La primera validación demostró que el ambiente de visualización, permite seleccionar diferentes Estados (Puebla, Colima, Zacatecas, Veracruz) y obtener un capa de forma dinámica en este caso `camino50_1_utm`.
- La segunda validación demostró que es posible generar no solo un tipo de capa a partir de un Estado, sino que permitió la creación de varios tipos de capas al Estado de México. Fue en esta validación donde se aprecia, que no es posible generar todas la capas, por ejemplo `calle50_1_utm` no se pudo generar debido a los miles de polígonos que contiene. Por lo que en estos casos es preferible la opción de selección por zonas geográficas para poder visualizar capas de gran tamaño.
- La selección por zonas geográficas (segunda validación) demostró que es posible generar capas por zonas, pero esta opción tiene la limitación que en esa zona no exista por ejemplo un aeropuerto, zonas verdes, cuerpos de agua, entre otras. Por lo que las dos opciones selección de zonas y Estados son necesarias para realizar diverso estudios (e.g. actividad forestal, caminos, red de transporte, salud, etc.).
- La segunda validación se demuestra que es posible aplicarle la función de categorización a zonas geográficas y Estado es posible. La categorización tiene el problema que al existir una gran cantidad de datos y estos generan muchas categorías generan confusión por su color, el cual es muy parecido a otros.
- La última validación sirvió para mostrar que el sistema puede desplegar capas de diferentes fuentes siempre que sean en en formato GeoJSON y KML. Con el formato KML se tuvieron problemas para realizar la categorización lo cual se debe posiblemente a que OpenLayers 3.0 tiene problemas con el formato KML.

Por último se realizo una comparativa del SIG-Web con los sistemas estudiados en el capítulo 3, el nombre “AV en SIG-Web” se refiere al ambiente creado. La Tabla 6.1, muestra las arquitecturas del estado del arte, la mayoría de éstas fueron realizadas por medio de herramientas *opensource*. La herramienta OpenLayers es muy útil para la creación de mapas Web. Las arquitecturas de estos trabajos están basados en tres capas lógicas para su funcionamiento. Las tecnologías WFS y WMS son muy utilizadas para obtener información geográfica por medio de servidores de mapas, siendo GeoServer el mas utilizado. En conclusión el ambiente de visualización utiliza software *OpenSource* esencial en la creación de SIG Web. la gran diferencia con los demás sistemas es el formato geográfico utilizado GeoJSON, el cual no es mencionado, pero la nueva version de OpenLayers 3.0 ha logrado que este formato retome su importancia.

Nombre	Formato de geográfico	Lenguaje de desarrollo	Año	BD
AV en SIG-Web	GeoJSON	PHP y JavaScript (OpenLayers 3.0),	2016	PostgreSQL y PostGIS
SIG Web (sección 3.5) [67]	ASP.NET, TF.Net, FDO y JavaScript(OpenLayers 2.0)	.Net	2016	SQL Express
MapBeing [60]	WFS, WMS y WCS	JavaScript	2014	PostgreSQL y PostGIS
SIG inventario emisiones (sección 3.4) [66]	WMS y WFS (WFS-T)	JavaScript (GoogleMaps y OpenLayers 2.0)	2014	PostgreSQL y PostGIS
PA-CA [63]	WFS	Flash y SVG	2013	PostgreSQL y PostGIS
SIG Web (sección 3.3) [64]	WFS y WMS	JavaScript (GoogleMaps)	2006	Datos de Google-Maps

Tabla 6.1: Comparativa entre las diferentes arquitecturas estudiadas.

## 6.5. Resumen

El capítulo describió las tres validaciones que se realizaron al ambiente de visualización, la primera a diferentes Estados (Puebla, Colima, Zacatecas y Veracruz) y zonas geográficas de puntos reconocidos como son las ciudades de Puebla, Veracruz y Monterrey. La segunda validación tomo como caso de estudio al Estado de México, un sitio que cuenta con información geográfica de todos tipo (e.g. caminos, puentes, aeropuertos, cuerpos de agua). y se puede realizar la categorización a cada uno de los atributos de cada capa. La tercera validación mostró que el sistema puede crecer y complementarse a partir de información externa y que puede ser estudiada. Se describe por último una comparativa entre los diferentes SIG Web estudiados.

# Capítulo 7

## Conclusiones y trabajo a futuro

En este capítulo se describe las contribuciones obtenidas a lo largo del desarrollo de la presente tesis, mencionando puntos importantes a resaltar como objetivos logrados y problemas que se lograron resolver. Se realiza las conclusiones de este proyecto, problemas resolver y algunas ideas para continuar el trabajo desarrollado hasta ahora.

### 7.1. Conclusiones

La Web a cambiado la forma de vida de los humanos y ha tenido una gran influencia en la creación de nuevos Sistemas y los SIG no son la excepción. Los SIG han tenido un crecimiento desde la década de los 60, éstos han pasado de ser sistemas de uso exclusivo de gobiernos y de universidades a uso público, gracias a la Internet que ha permitido la creación un nuevo tipo de sistema.

La unión de los SIG y la Web dio nacimiento a un nuevo tipo de sistemas denominados SIG Web, los cuales utilizan las tecnologías Web como HTML, Servicios Web (SOAP, XML), JavaScript, AJAX, entre otras para la solución de problemas geográficos. Como una conclusión general, los SIG Web seguirán creciendo debido a la necesidad ofrecer servicios geográficos de manera remota, ofrecer los SIG sin la necesidad de realizar su instalación en un equipo de computo.

De acuerdo a los resultados obtenidos de las validaciones del SIG-Web y de los objetivos de esta tesis podemos realizarlas siguientes observaciones:

- Se cumple con el objetivo de visualizar objetos geográficos de tipo vectorial, no solo de las cartas topográficas sino también de una fuente diferente como es la lectura de archivos de tipo GeoJSON y KML para poder probar diferentes capas sin la necesidad de conectarse a la BD.
- La aplicación puede seleccionar datos geográficos (e.g. carreteras, caminos, calles, etc) de diferentes Estados de la República Mexicana y de sus zona geográficas a

partir de las cartas topográficas existentes. Además es necesario agregar elementos geográficos (aeropuertos, áreas verdes, lugares ) en ciertas zonas de México. Cabe mencionar que la función que se utilizó fue la intersección del Estado con objetos de la capa seleccionada para no perder elementos, lo anterior provoca que en varios casos los elementos geográficos se extiendan más allá del Estado seleccionado. La función de diferencia también se implementó pero esta tarda demasiado en generar el archivo GeoJSON y solo fue conveniente en capas con menores elementos (e.g. el Estado de Veracruz es muy grande y procesa miles de elementos de sus caminos).

- Se puede consultar datos geográficos (e.g. información de una carretera, un aeropuerto ), por medio de la selección de estos en el mapa, y presentar la información consultada en el mismo mapa.
- Se crearon dos interfaces para la visualización de las cartas topográficas, ambas utilizan dos mapas para ofrecer la selección de Estados o zonas geográficas de una manera visual y interactiva. La primera está enfocada en la selección de Estados y la segunda en la selección de zonas geográficas.

La plataforma Web creada tiene una arquitectura de tres capas (presentación, negocios y datos), cada una de las capas está conformada por herramientas y APIs de libre distribución (*open source*). La combinación de este tipo de herramientas PostGIS, AJAX, JavaScript, jQuery, OpenLayers, Apache, GeoJSON, PHP, permitieron crear un ambiente de visualización sin gastar un solo peso o tener que gastar por la licencia por ser herramientas de libre distribución. Lo cual supuso un gran reto por que muchas de estas incorporan diferentes herramientas que no cuentan con una comunicación directa como las aplicaciones comerciales las cuales ofrecen múltiples herramientas (e.g servidores Web geoespaciales, conexión directa con una BDE, herramientas más precisas, compatibilidad con un *kit* de herramientas comerciales) y facilidades (e.g ESRI es la compañía líder en productos geoespaciales y ofrece varios programas para trabajar con datos de formato raster y vectorial) para la comunicación entre sus módulos.

El SIG Web necesita ser modificado para ofrecer una mejor interfaz, lo cual es importante recalcar que este tipo de sistemas son muy flexibles a cambios y no es necesario la instalación o la actualización de este tipo de sistemas en un equipo en particular. Retomando la flexibilidad de estos sistemas es posible incorporar diferentes módulos, diferentes tecnologías o lo más importante enfocar la aplicación a una área de estudio en particular tomando como base el presente proyecto.

## 7.2. Trabajo a futuro

El SIG Web desarrollado en este proyecto de tesis está enfocado en zonas geográficas y Estados, para el trabajo a futuro se tiene contemplado adaptar el sistema para la

selección de municipios y localidades, lo anterior con el fin de crear capas más específicas. Incorporar las capas restantes de la BD MapasMexico , modificar las coordenadas de las tablas para no realizar la conversión de forma dinámica sino de forma permanente y realizarles diferentes pruebas. A continuación se mencionan diferentes puntos que deben realizarse para que el SIG Web sea mejorado.

### **Creación de módulos**

La creación de nuevos módulos, estarán enfocados dar nuevas capacidades al SIG Web, siendo el tema de insertar información geográfica y nominal a la BD MapasMexico de gran importancia. Para hacer ésto de manera más sencilla es incorporar módulo que permitan manejar Servicios Web Geoespaciales, en este caso el servicio WFS-T, el cual permite consultar, insertar, eliminar información por medio de un servidor de mapas. Para realizar ésto es necesario utilizar un servidor de mapas compatible con servicios estándar geoespaciales, como son GeoServer [82] y MapServer [83]. El más útil sería utilizar un servidor GeoServer por que es el que actualmente tiene un gran soporte con diferentes herramientas de libre distribución como son PostGIS, QGIS y OpenLayers. Otro modulo de gran importancia sería el poder integras por medio del servicio WMS (ver Sección 2.7), el consumo de capas vectoriales externas al sistema.

### **Métricas para el SIG Web**

Actualmente no existen métricas estandarizadas para evaluar las aplicaciones de mapas Web, existe una propuesta [84]. Esta propone métricas de mapas Web para evaluar aplicaciones en bibliotecas académicas. También contempla evaluar diferentes puntos de las aplicaciones SIG Web como la funcionalidad, la usabilidad y incluye otros puntos a evaluar como son los datos, los metadatos, los números de variables, resolución de los elementos espaciales y la interacción con la información. La aplicación SIG creada necesita mejorar su desempeño, es necesario mejorar las consultas SQL y tal vez convertir las coordenadas de las tablas de BD MapasMexico de forma definitiva a WGS84, es de utilidad medir los tiempos de respuesta del sistema en un solo equipo e implementar laa arquitectura de tres capas de manera física.

### **Usabilidad**

Es necesario realizar un estudio de usabilidad, ésto con el fin de encontrar características que se necesiten mejorar en la interfaz del SIG Web desarrollado, probar diferentes diseños, la desarrollada utiliza un diseño de dos mapas, por lo que es necesario comparar interfaces de un mapa y dos para obtener cual es la mejor opción. La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. Esta propiedad es de suma importancia para

cualquier sistema.

La interfaz necesita ser mejorada para ofrecer una mejor experiencia al usuario, además es necesario agregar nuevas herramientas al SIG Web para extender su funcionalidad.

## Toma de decisiones

SIG Web desarrollado actualmente incorpora la visualización de capas vectoriales y el módulo de categorización, que ayudan a tomar decisiones pero esto a nivel infraestructura, por lo que la toma de decisiones se realiza por medio de la observación. Porque es necesario crear nuevos módulos para complementar este SIG Web, un módulo de estadística sería de mucha ayuda para analizar los datos nominales (e.g. población, área, altitud, nombre, capital, ciudad, pueblo, etc). Con la estadística también sería conveniente la creación de gráficas para interpretar de manera visual la información contenida en la BD.

## Uso de imágenes raster y imágenes SVG

El SIG Web solo está enfocado en la visualización de datos vectoriales, para la visualización de datos raster se necesita mostrar una gran cantidad de información, lo cual no es nada sencillo si hablamos de tecnologías Web. Para visualizar imágenes raster es necesario la creación de un proveedor de estas imágenes alojadas en un servidor para su consumo, lo anterior completaría el SIG desarrollado.

El formato SVG ofrece imágenes de tipo vectorial, OpenLayers 3.0 actualmente no puede utilizar imágenes SVG por lo que es necesario el uso de bibliotecas externas. La biblioteca que ha comprobado ser de utilidad al combinarse con OpenLayers es D3 (*Data-Driven Documents*) [85] la cual ofrece la visualización de diferentes formatos de datos como SVG. La combinación de estas dos bibliotecas es de mucha utilidad para mejorar la capacidad de visualización del SIG Web.

## Paredes de Vídeo

Las paredes de vídeo son herramientas que han llegado a ser bastante populares en la comunidad científica, especialmente en el área de visualización científica. Estos dispositivos permiten el despliegue de visualizaciones de alta resolución y de grandes dimensiones (ver Sección 3.6). Se tiene contemplado adaptar el SIG Web a paredes de vídeo, el desafío de acoplar a este tipo de dispositivos es la arquitectura de la pared de vídeo si es un sistema distribuido se tiene que controlar la comunicación de nodos a un nodo central, si es un servidor de visualización con varias tarjetas de vídeo. El SIG Web tiene que adaptarse por medio de WebGL, y modificar módulos de la biblioteca de OpenLayers 3.0 que actualmente cuenta con la capacidad de renderizar objetos por medio de WebGL, ésto con el fin utilizar las tarjetas de vídeo del servidor de visualización y poder incorporar



incluso elementos 3D a la aplicación.

Actualmente se trabaja mucho con los SIG Web y SIG Móviles, para expandir el área de los SIG. Se tiene como trabajo a futuro adaptar el SIG a dispositivos móviles, lo cual permitirá añadir nuevas funciones y aprovechar el objetivo de los SIG Web, crear sistemas distribuidos y con nuevas funciones, lo anterior permite un campo más amplio de usuarios tiene acceso la aplicación y enfocarla a casos particulares como: emergencias, desastres naturales, reportar incidentes, colaboración. Las nuevas tendencias mencionan que los SIG Web serán ubicuos y estarán en casi todo dispositivo móvil, computadora, entre otros. Para finalizar, los SIG Web tienen un gran futuro por delante y la aplicación desarrollada tiene diferentes opciones para mejorar y crecer en el futuro.



# Apéndice A

## OpenLayers 3.0

OpenLayers 3.0 es una biblioteca bien estructurada, modular, y compleja, donde la flexibilidad y consistencia toman más prioridad que desempeño. Sin embargo esto no significa que OpenLayers 3 sea lento. Por el contrario, la biblioteca supera a su predecesora OpenLayers 2. Por lo tanto su diseño cómodo y lógico realmente no afecta adversamente su funcionamiento. La relación de las partes más esenciales pueden ser descrita con un diagrama UML (Universal Modeling Language) el cual se muestra en la Figura A.1.

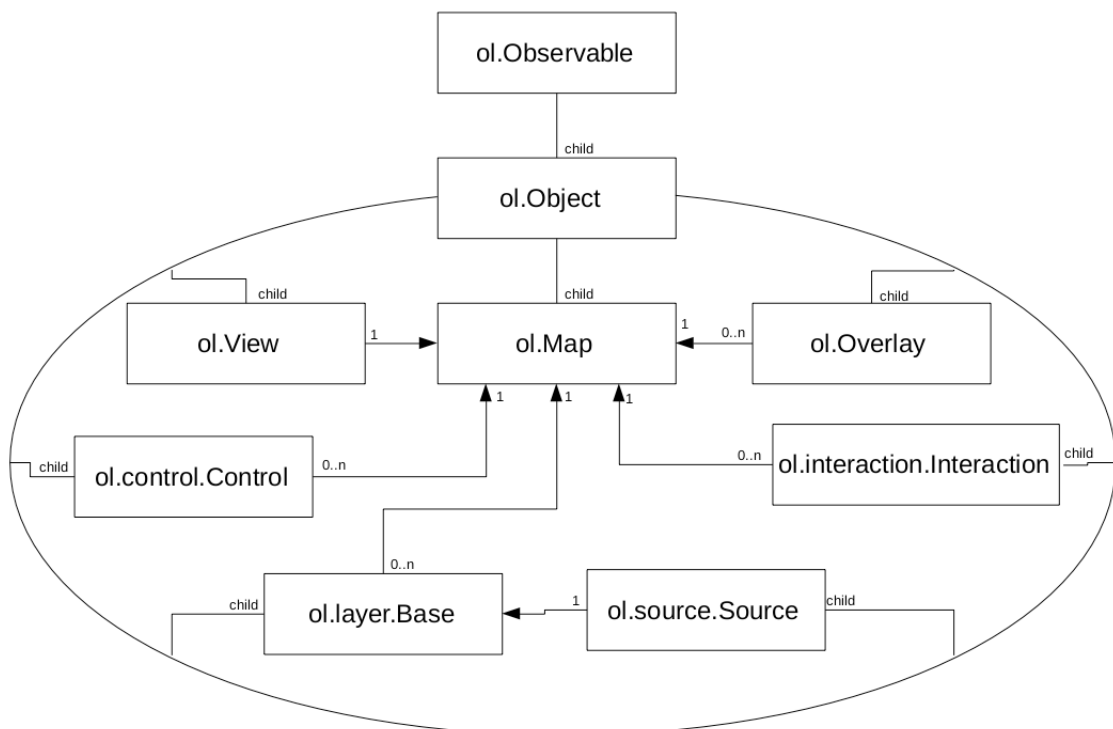


Figura A.1: Diagrama UML de OpenLayers 3.

En la Figura anterior se puede observar dos superclase `ol.Observable` y `ol.Object`, las cuales se sugiere no tener contacto directo. La primer superclase `ol.Observable` garantiza que todas sus clases hijas tenga métodos consistentes. La segunda superclase `ol.Object`, extiende a su padre con métodos que permitan una administración más fácil. `ol.Object` también permite el manejo los métodos `get` y `set`.

## Propiedades y métodos de los mapas

La biblioteca de OpenLayers 3 incluye tres tipos de renderización: Canvas, DOM y WebGL. Los tres soportan datos raster de servidor y datos vectoriales. Sin embargo WebGL solo soporta vectores de tipo punto y no soporta etiquetas. Claramente solos los navegadores que soporta la renderización Canvas pueden usar este tipo de renderización. Igualmente solo los navegadores y dispositivos que soporten WebGL pueden utilizar este tipo.

## Objetos y nomenclatura

OL3 usa una jerarquía similar a las bibliotecas cerradas. Existe un espacio de nombre de nivel superior `ol` (básicamente `var ol=;`). Las subdivisiones que forman a la clase `ol` son:

- Otros espacios de nombres, como `ol.layer`. estos tienen una inicial minúscula.
- Objetos simples que contienen propiedades y métodos estáticos, tales como `ol.animation`; Estos también tienen una inicial de minúscula.
- Tipos, que tienen una mayúscula inicial. Estas son principalmente 'clases', que aquí significa una función constructora con herencia, como `ol.Map` o `ol.layer.Vector` (la clase `Vector` dentro del espacio de nombres de la capa). Sin embargo, existen otros tipos más simples, como `ol.Extent`, que es una matriz.

Los espacios de nombres de clase, como `ol.layer`, tienen un tipo de clase base con el mismo nombre, como `ol.layer.Layer`. Estas son principalmente clases abstractas, de las cuales heredan las otras subclases.

## Elementos principales de OpenLayers 3.0

### Map

La clase `ol.Map` usa internamente solo cuatro propiedades principales, las cuales son: `target`, `layergroup`, `view`, `size`. La clase `ol.Map` es el componente principal de Openlayers. El cual es renderizado a un contenedor `target`(por ejemplo un elemento `div` sobre una pagina

web puede contener el mapa). Todas las propiedades de map pueden ser configurados en el momento de la construcción, o mediante el uso de métodos setter como por ejemplo `setTarget()`.

`ol.Map` es la clase que permite manejar el concepto de mapa en nuestra aplicación. Por medio de `ol.Map` se puede añadir y remover capas, controles, incrustaciones y interacciones. En su forma más simple, crear una nueva instancia de `ol.Map` requiere que el usuario especifique un objeto con las siguientes propiedades:

- *target*, el elemento HTML destino(*target*) , donde se mostrará el mapa.
- *layers*, una o más referencias de capa con los datos a mostrar.
- *view*, una instancia `ol.View` es responsable de manejar la manera en que el mapa es visualizado.

Para poder crear un mapa es necesario crear un elemento DOM que actuará como el objetivo, y permitirá el despliegue de un mapa. Por lo tanto, dado el siguiente elemento DOM, que actuará como el objetivo (código A.1):

```
1 <div id="map" style="width: 100%, height: 400px"></div>
2 <script>
3   var map = new ol.Map({target: 'map'});
4 </script>
```

Código A.1: Creación de elemento DOM para asignar un mapa

## View

La clase `ol.Map` no esta encargada de funciones como centrar, el nivel de ampliación y proyección. El encargado de realizar estas funciones es la clase `ol.View`. Esta clase también tiene una proyección (*projection*). La proyección determina el sistema de coordenadas del centro (*center*) y de las unidades que usa el mapa para calcular resoluciones. La proyección por defecto en OpenLayers es la proyección Esférica de Mercator (EPSG: 3857), con metro como unidades del mapa.

La clase `ol.View` (código A.2) se encarga de la opción *zoom*, la cual es conveniente para especificar la resolución del mapa, y permite delimitar el nivel máximo y minino de aplicación del mapa.

```
1   map.setView(new ol.View({
2     center: [0, 0],
3     zoom: 2
4   }));
```

Código A.2: Declaración de un objeto del tipo `ol.View`

## Source

OpenLayers 3 usa la subclase `ol.source.Source` (código A.3) para obtener datos remotos para una capa. Existen servicios de mapas gratis y comerciales como OpenStreetMap o Bing, para fuentes OGC como WMS o WMTS, y para datos vectoriales en formatos como GeoJSON o KML.

```
1 var osmSource = new ol.source.OSM();
```

Código A.3: Creación de un objeto `ol.source`

## Layer

Una capa(layer) es una representación visual de datos de una fuente(source). OpenLayers 3 cuenta con tres tipos de capas: `ol.layer.Tile` (código A.4), `ol.layer.Image` y `ol.layer.Vector`.

- `ol.layer.Tile`, es para fuentes de capa que proporcionan imágenes en forma de mosaico prerenderizadas en cuadrículas que están organizadas por niveles de zoom para resoluciones específicas.
- `ol.layer.Image`, es para las imágenes renderizadas por el servidor que están disponibles para extensiones y resoluciones arbitrarias.
- `ol.layer.Vector`, es para datos vectoriales que se procesa en el lado del cliente.

```
1 var osmLayer = new ol.layer.Tile({source: osmSource});  
2 map.addLayer(osmLayer);
```

Código A.4: Declaración de una nueva capa

Al poner los elementos anteriormente mencionados, `ol.Map`, `ol.View`, `ol.source`, `ol.layer`. Se obtiene un mapa funcional con contenido (código A.5).

```
1 <div id="map" style="width: 100%, height: 400px"></div>  
2 <script>  
3   new ol.Map({  
4     layers: [  
5       new ol.layer.Tile({source: new ol.source.OSM()})  
6     ],  
7     view: new ol.View({  
8       center: [0, 0],  
9       zoom: 2  
10    }),  
11    target: 'map'  
12  });  
13 </script>
```

Código A.5: Mapa de OpenLayers 3

# Apéndice B

## GeoJSON

GeoJSON es un formato para codificación de una variedad de estructuras de datos geográficos. GeoJSON es un formato de intercambio de datos geoespaciales basado en JavaScript Object Notation (JSON) [86]. Un objeto GeoJSON puede representar una geometría, una característica, o una colección de características.

GeoJSON soporta los siguientes tipos geométricos: Point, LineString, Polygon, MultiPolygon, MultiLineString, y MultiPolygon. Los objetos geométricos con propiedades adicionales son objetos Feature. Los conjuntos de características están contenidas por los objetos FeatureCollection.

Una estructura de datos GeoJSON siempre es un objeto (en términos JSON). En GeoJSON, un objeto consiste de una colección de un par nombre/valor también es llamado miembros. Para cada miembro, el nombre siempre es un string. Los valores miembro son o bien una cadena, número, objeto, matriz o uno de los literales: true, false, y null. Una matriz se compone de elementos, donde cada elemento es un valor como se describe anteriormente.

## Coordenadas

Una coordenada es el constructor geométrico fundamental. El miembro “coordinates” de un objeto geométrico es compuesto de una posición (en el caso de un Point Geométrico), un arreglo de posiciones (Geometrías LineString o MultiPoint), un arreglo de arreglos de posiciones (Polígonos, MultiLineStrings), o un arreglo multidimensional de posiciones (MultiPolygon). Una posición es representado por una arreglo de números. Deben ser 2 elementos o más, los cuales deben seguir el orden x, y ,z (longitud, latitud y altitud).

## Punto

Las coordenadas de los puntos (código B.1) están en orden x, y (longitud y latitud para coordenadas geográficas):

```
1 { "type": "Point", "coordinates": [100.0, 0.0] }
```

Código B.1: Ejemplo de un Point en GeoJSON

## MultiPoint

Este objeto esta compuesto por múltiples Point y los contiene en un arreglo de posiciones (código B.2).

```
1 { "type": "Point", "coordinates": [100.0, 0.0] }
```

Código B.2: Ejemplo de un MultiPoint

## LineString

Las coordenadas de los objetos LineString son un arreglo de posiciones (código B.3).

```
1 { { "type": "LineString",  
2   "coordinates": [ [100.0, 0.0], [101.0, 1.0] ]  
3   } }
```

Código B.3: Ejemplo de un LineString

## MultiLineString

Las coordenadas de los objetos MultiLineString son un arreglo de LineString con sus arreglos de coordenadas (código B.4).

```
1 { "type": "MultiLineString",  
2   "coordinates": [  
3     [ [100.0, 0.0], [101.0, 1.0] ],  
4     [ [102.0, 2.0], [103.0, 3.0] ]  
5   ]  
6 }
```

Código B.4: Ejemplo de un MultiLineString



## Polygon

Las coordenadas de un objeto Polygon son una matriz de coordenadas (código B.5).

```

1 { "type": "Polygon",
2   "coordinates": [
3     [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0,
4       0.0] ]
5   ]
6 }

```

Código B.5: Ejemplo de un Polygon

## MultiPolygon

Las coordenadas de un objeto MultiPolygon contienen un arreglo de varios objetos Polygon en forma de arreglos de coordenadas (código B.6).

```

1 { "type": "MultiPolygon",
2   "coordinates": [
3     [[ [102.0, 2.0], [103.0, 2.0], [103.0, 3.0], [102.0, 3.0], [102.0,
4       2.0]]],
5     [[ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0,
6       0.0]],
7     [[ [100.2, 0.2], [100.8, 0.2], [100.8, 0.8], [100.2, 0.8], [100.2,
8       0.2]]]
9   ]
10  }

```

Código B.6: Ejemplo de un MultiPolygon

## GeometryCollection

El objeto GeometryCollection es un objeto geométrico o geográfico que contienen los elementos anteriormente descritos (código B.7).

```

1 { "type": "GeometryCollection",
2   "geometries": [
3     { "type": "Point",
4       "coordinates": [100.0, 0.0]
5     },
6     { "type": "LineString",
7       "coordinates": [ [101.0, 0.0], [102.0, 1.0] ]
8     }
9   ]
10 }

```

---

Código B.7: Ejemplo de un GeometryCollection

# Referencias

- [1] E. M. Knapp and D. Rider, “Automated geographic information systems and landsat data: A survey,” *Harvard Library of Computer Graphics*, vol. 4, pp. 57–68, 1979.
- [2] T. Ott and F. Swiaczny, “The analysis of cultural landscape change: a gis approach for handling spatio-temporal data,” *History and Computing*, vol. 10, no. 1-3, pp. 37–49, 1998.
- [3] P. Fu and J. Sun, “*Web GIS: principles and applications*”. Esri Press, 2010.
- [4] P. Hendriks and H. Ottens, “Geografische informatie systemen in ruimtelijk onderzoek,” *Assen, Van Gorcum*, 1997.
- [5] T. Ott and F. Swiaczny, “*Time-integrative geographic information systems: management and analysis of spatio-temporal data*”. Springer Science & Business Media, 2001, vol. 1.
- [6] J. Raper, “2.5-and 3-d gis for coastal geomorphology,” *Marine and Coastal Geographical Information Systems*, pp. 129–136, 1999.
- [7] F.-J. Behr, “*Strategisches GIS-Management: Grundlagen und Schritte zur System-einführung*”. Wichmann, 1998.
- [8] R. Laurini and D. Thompson, “*Fundamentals of spatial information systems*”. Academic press, 1992, vol. 37.
- [9] O. G. Consortium *et al.*, “Opengis® implementation specification for geographic information-simple feature access-part 1: Common architecture,” *OGC document*, 2011.
- [10] P. Longley, “*Geographic information systems and science*”. John Wiley & Sons, 2005.
- [11] A. Sakuraba, S. Nodas, T. Ishida, Y. Ebara, and Y. Shibata, “Tiled display environment to realize gis based disaster information system,” in *Network-Based Information Systems (NBiS), 2013 16th International Conference on*. IEEE, 2013, pp. 311–316.

- [12] J. Meier, A. Homer, D. Hill, J. Taylor, and P. Bansode, “*Web Application Architecture Guide: patterns and practices*”. Microsoft, 2008.
- [13] S. V. C. Vergara, J. M. Cruz, N. S. Romero, and J. C. E. Mora, “*Memoria de la Base de Datos de la Colección Nacional Microbiana CDBB-500*”, 1st ed. Porrúa, 2013.
- [14] P. L. Luna, “Sistema de información geográfica para la toma de decisiones en políticas sociales,” Tesis de maestría, CINVESTAV-IPN, Mexico-DF, 2 2010.
- [15] C. H. Nava, “Diseño y desarrollo de una api para la visualización de datos espaciales en sistemas web de información geográfica para dispositivos móviles.” Tesis de maestría, CINVESTAV-IPN, Mexico-DF, 2 2009.
- [16] P. L. Luna, “Minería de datos visual sobre una pared de vídeo,” Tesis de maestría, CINVESTAV-IPN, Mexico-DF, 10 2008.
- [17] L. P. R. Rivera, “Objetos distribuidos visuales,” Tesis de doctorado, CINVESTAV-IPN, Mexico-DF, 11 2013.
- [18] L. P. Rivera, S. V. C. Vergara, and A. M. Viveros, “Visual data mining over a video wall,” in *Electrical Communications and Computers (CONIELECOMP), 2012 22nd International Conference on*. IEEE, 2012, pp. 239–244.
- [19] C. G. H. Pérez, “Programación automática de sistemas web a partir de descripciones xml. caso de estudio: Sistemas de información geográfica,” Tesis de maestría, CINVESTAV-IPN, Mexico-DF, 6 2008.
- [20] F. R. López, “Modelado de datos para base de datos espaciales. caso de estudio: sistemas de información geográfica.” Tesis de maestría, CINVESTAV-IPN, Mexico-DF, 5 2008.
- [21] P. A. Burrough and R. A. McDonnell, “Principles of gis,” *Oxford University Press: London, UK*, 1998.
- [22] T. W. Foresman, “*The history of geographic information systems: perspectives from the pioneers*”. Prentice Hall, 1998.
- [23] T. K. Peucker and N. Chrisman, “Cartographic data structures,” *The American Cartographer*, vol. 2, no. 1, pp. 55–69, 1975.
- [24] C. Harder, “*Serving Maps on the Internet: geographic information on the world wide web*”. ESRI press, 1998.
- [25] S. Emmott and S. Rison, “Towards 2020 science,” 2006.

- [26] T. R. Smith, S. Menon, J. L. Star, and J. E. Estes, "Requirements and principles for the implementation and construction of large-scale geographic information systems," *International Journal of Geographical Information System*, vol. 1, no. 1, pp. 13–31, 1987.
- [27] S. Aronoff, *Geographic information systems: a management perspective*. Taylor & Francis, 1989.
- [28] D. J. Cowen, "Gis versus cad versus dbms: What are the differences?." *Photogramm. Eng. Remote Sens.*, vol. 54, no. 11, pp. 1551–1555, 1988.
- [29] P. Bolstad, *GIS Fundamentals: A First Text on Geographic Information Systems*. Eider Press, 2005.
- [30] K. E. Foote and M. Lynch, "Geographic information systems as an integrating technology: Context, concepts, and definitions," <http://www.colorado.edu/geography/gcraft/notes/intro/intro.html>, accedido: 2015-11-10.
- [31] L. Lang, "Gis for health organizations," 2000.
- [32] L. Lang and Calif, *Transportation GIS*. ESRI Press Redlands, 1999, vol. 118.
- [33] R. W. Greene, *GIS in public policy: Using geographic information for more effective government*. ESRI, Inc., 2000.
- [34] D. Martin, *Geographic information systems: socioeconomic applications*. Psychology Press, 1996.
- [35] A. S. Belward, C. R. Valenzuela *et al.*, *Remote sensing and geographical information systems for resource management in developing countries*. Kluwer Academic, 1991.
- [36] V. Singh and M. Fiorentino, *Geographical information systems in hydrology*. Springer Science & Business Media, 2013, vol. 26.
- [37] P. Hohl, *GIS data conversion: strategies, techniques, and management*. OnWord Press, 1998.
- [38] M. B. Hesse, "Models and analogies in science," 1966.
- [39] G. Hake, D. Grünreich, and L. Meng, *Kartographie: Visualisierung raum-zeitlicher Informationen*. Walter de Gruyter, 1994.
- [40] M. F. Goodchild, "Geographical information science," *International journal of geographical information systems*, vol. 6, no. 1, pp. 31–45, 1992.

- [41] D. J. Peuquet and D. F. Marble, "Arc/info: An example of a contemporary geographic information system," *Introductory Readings in Geographic Information Systems*, pp. 90–99, 1990.
- [42] A. Lange, "Gps/gis integration-accuracy specifications affect application success," *GIS WORLD*, vol. 10, pp. 58–59, 1997.
- [43] R. O. Obe and L. S. Hsu, "*PostGIS in action*". Manning Publications Co., 2015.
- [44] O. Huisman and R. de By, "Principles of geographic information systems," *ITC Educational Textbook Series*, vol. 1, 2009.
- [45] K. Stolze, "Sql/mmm spatial-the standard to manage spatial data in a relational database system." in *BTW*, vol. 2003, 2003, pp. 247–264.
- [46] S. Gopi *et al.*, "*Advanced Surveying: Total Station, GIS and Remote Sensing*". Pearson Education India, 2007.
- [47] "Esri," <http://www.esri.com/>, accedido: 2016-10-10.
- [48] "Googlemaps," <https://www.google.com.mx/maps>, accedido: 2016-11-6.
- [49] "Yahoo maps," <https://maps.yahoo.com/>, accedido: 2016-10-10.
- [50] "Bing maps," <https://www.bing.com/maps/>, accedido: 2016-10-4.
- [51] "Grass gis," <https://grass.osgeo.org/>, accedido: 2016-10-4.
- [52] "Qgis," <https://www.qgis.org/>, accedido: 2016-10-4.
- [53] J. Campbell and M. E. Shin, "*Essentials of Geographic Information Systems*". Flat world knowledge, 2011.
- [54] Y. Fang, V. Shandas, and E. Arriaga Cordero, "*Spatial Thinking in Planning Practice: An Introduction to GIS*". Portland State University Library, 2014.
- [55] R. F. Tomlinson, "*Thinking about GIS: geographic information system planning for managers*". ESRI, Inc., 2007.
- [56] P. Richardus and R. K. Adler, "*Map projections for geodesists, cartographers and geographers*". North-Holland, 1972.
- [57] T. Berners-Lee, T. Bray, D. Connolly, and P. Cotton, "Architecture of the world wide web, volume one," <https://www.w3.org>, accedido: 2016-07-12.
- [58] M. Rouse, "Definition web 2.0," <http://whatis.techtarget.com/definition/Web-20-or-Web-2>, accedido: 2016-08-12.

- [59] A. Alesheikh, H. Helali, and H. Behroz, "Web gis: technologies and its applications," in *Symposium on geospatial theory, processing and applications*, vol. 15, 2002.
- [60] M. Moshi, N. Nahar, R. Rahman, and K. Sakib, "Mapbeing: An architecture for manipulating and publishing vector data in web based geographic information system," in *Software, Knowledge, Information Management and Applications (SKIMA), 2014 8th International Conference on*. IEEE, 2014, pp. 1–7.
- [61] N. Rishe, S.-C. Chen, N. Prabakar, M. A. Weiss, W. Sun, A. Selivonenko, and D. Davis-Chu, "Terraflly: A high-performance web-based digital library system for spatial data access." in *ICDE Demo Sessions*, 2001, pp. 17–19.
- [62] K. Charvat, P. Horak, M. Vlk, J. HIELKEMA, S. Kafka, J. CEPICKU, J. JEZEK, and O. CERBA, "Geohosting—publish your spatial data yourself," *IST Africa, Kampala May*, 2009.
- [63] A. M. MacEachren, S. Crawford, M. Akella, and G. Lengerich, "Design and implementation of a model, web-based, gis-enabled cancer atlas," *The Cartographic Journal*, 2013.
- [64] A. Sayar, M. Pierce, and G. Fox, "Integrating ajax approach into gis visualization web services," in *Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services (AICT-ICIW'06)*. IEEE, 2006.
- [65] L. Lei and Z. Duan, "Integrating ajax and web services for cooperative image editing," *IT Professional*, vol. 9, no. 3, pp. 25–29, 2007.
- [66] B. Delipetrev, A. Jonoski, and D. P. Solomatine, "Development of a web application for water resources based on open source software," *Computers & Geosciences*, vol. 62, pp. 35–42, 2014.
- [67] D. Lv, X. Ying, X. Gao, W. Tao, Y. Cui, and T. Hua, "A webgis platform design and implementation based on open source gis middleware," in *Geoinformatics, 2016 24th International Conference on*. IEEE, 2016, pp. 1–5.
- [68] M. Czerwinski, G. Robertson, B. Meyers, G. Smith, D. Robbins, and D. Tan, "Large display research overview," in *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 2006, pp. 69–74.
- [69] P. A. Navrátil, B. Westing, G. P. Johnson, A. Athalye, J. Carreno, and F. Rojas, "A practical guide to large tiled displays," in *Advances in Visual Computing*. Springer, 2009, pp. 970–981.

- [70] R. Ball, M. Varghese, B. Carstensen, E. D. Cox, C. Fierer, M. Peterson, and C. North, "Evaluating the benefits of tiled displays for navigating maps," in *IASTED International Conference on Human-Computer Interaction*, 2005, pp. 66–71.
- [71] A. Ebert, S. Thelen, P.-S. Olech, J. Meyer, and H. Hagen, "Tiled++: An enhanced tiled hi-res display wall," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 1, pp. 120–132, 2010.
- [72] A. Sorokine, "Implementation of a parallel high-performance visualization technique in grass gis," *Computers & Geosciences*, vol. 33, no. 5, pp. 685–695, 2007.
- [73] B. Fjukstad, T.-M. S. Hagen, D. Stødle, P. H. Ha, J. M. Bjørndalen, and O. Anshus, "Interactive weather simulation and visualization on a display wall with many-core compute nodes," in *Applied Parallel and Scientific Computing*. Springer, 2012, pp. 142–151.
- [74] P. Yin, X. Jiang, J. Shi, and R. Zhou, "Multi-screen tiled displayed, parallel rendering system for a large terrain dataset." *IJVR*, vol. 5, no. 4, pp. 47–54, 2006.
- [75] G. Farkas, *"Mastering OpenLayers 3"*. Packt Publ., 2016.
- [76] J. D. Gauchat, *"El gran libro de HTML5, CSS3 y Javascript"*. Marcombo, 2012.
- [77] J. J. Garrett *et al.*, "Ajax: A new approach to web applications," *Adaptive Path*, vol. 18, 2005.
- [78] B. Momjian, *"PostgreSQL: introduction and concepts"*. Addison-Wesley New York, 2001, vol. 192.
- [79] "Inegi," <http://www.inegi.org.mx/>, accedido: 2016-8-10.
- [80] "Conabio," <https://www.gob.mx/conabio>, accedido: 2016-8-18.
- [81] "Openstreetmap," <http://www.openstreetmap.org/>, accedido: 2016-6-18.
- [82] "Geoserver," <http://geoserver.org/>, accedido: 2016-10-10.
- [83] "Mapserver," <http://mapserver.org/>, accedido: 2016-10-10.
- [84] N. Kong, T. Zhang, and I. Stonebraker, "Common metrics for web-based mapping applications in academic libraries," *Online Information Review*, vol. 38, no. 7, pp. 918–935, 2014.
- [85] "D3," <https://d3js.org/>, accedido: 2016-10-10.
- [86] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub, and C. . Schmidt, "The geojson format specification," <http://geojson.org/geojson-spec.html>, accedido: 2016-10-10.