



**INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN**

**Especificación de un marco de trabajo para la implementación de
aplicaciones geoespaciales en la WEB**

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

VALDEZ HERNÁNDEZ FRANCISCO JAVIER

DIRECTORES DE TESIS:

Dr. Miguel Jesús Torres Ruiz

Dr. Rolando Quintero Téllez



INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 12:00 horas del día 12 del mes de junio de 2014 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

“Especificación de un marco de trabajo para la implementación de aplicaciones geoespaciales en la WEB”

Presentada por el alumno:

VALDEZ
Apellido paterno

HERNÁNDEZ
Apellido materno

FRANCISCO JAVIER
Nombre(s)

Con registro:

A	1	2	0	4	2	7
---	---	---	---	---	---	---

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA Directores de Tesis

Dr. Miguel Jesús Torres Ruiz

Dr. Rolando Quintero Téllez

Dr. Francisco Hiram Calvo Castro

Dr. Marco Antonio Moreno Ibarra

M. en C. Sandra Dinora Orantes Jiménez

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN
DIRECCIÓN

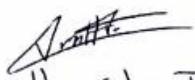


INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día 17 del mes de Junio del año 2014, la que suscribe VALDEZ HERNÁNDEZ FRANCISCO JAVIER alumno(a) del Programa de MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN, con número de registro A12418, adscrita al Centro de Investigación en Computación, manifiesto(a) que es el (la) autor(a) intelectual del presente trabajo de Tesis bajo la dirección del (de la, de los) Dr. Miguel Jesús Torres Ruíz y Dr. Rolando Quintero Téllez y cede los derechos del trabajo titulado Especificación de un marco de trabajo para la implementación de aplicaciones geoespaciales en la WEB, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del (de la) autor(a) y/o director(es) del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones fcovalher@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.


Valdez Hernández Francisco Javier
Nombre y firma del alumno(a)

Resumen

El uso de la *web* como medio de difusión de mapas es considerado un gran avance en la cartografía, las aplicaciones *web* que trabajan con mapas se conocen como Sistemas de Información Geográfica *Web* (*Web SIG*). El desarrollo de soluciones *Web SIG* es complejo y toma tiempo, lo cual ha conducido a la aparición de una serie de API¹ y *frameworks* utilizados para el desarrollo de diferentes tipos de SIG. Estas soluciones permiten una efectiva implementación de aplicaciones SIG para los usuarios que entienden las características de cada una de las API y *frameworks* utilizados, por lo tanto, este enfoque todavía requiere el tiempo de aprendizaje e integración de dichas herramientas, así como un extenso conocimiento en las tecnologías geoespaciales para la creación de funcionalidades específicas en el sistema.

El presente trabajo propone el diseño y desarrollo de un marco de trabajo (*Framework*) con herramientas de código abierto (*open source*), para agilizar el desarrollo de aplicaciones *Web SIG* con la menor codificación por parte del desarrollador. Para llevar a cabo esto se propone generar una interfaz (*front end*) *web* desde donde se pueden crear y administrar las aplicaciones *Web SIG* de una manera dinámica, sin la necesidad de demasiadas configuraciones.

La metodología se basa en la utilización de una arquitectura de tres capas: de datos, de lógica de negocio y de presentación, que contendrá cada aplicación *Web SIG* generada por el *Framework* propuesto. La capa de datos comprende el acceso a la información geoespacial proveniente de distintas fuentes y administrada desde un servidor de mapas; la capa de lógica de negocio se compone de un servidor *web* capaz de atender a las peticiones provenientes del cliente para entregar los datos espaciales recibidos del servidor de mapas. Por último, la capa de presentación es la encargada de entregar los mapas al navegador *web* del cliente.

Cabe mencionar que se implementa como caso de estudio el desarrollo de una aplicación *Web SIG* utilizando el *Framework* propuesto, con información geoespacial de vialidades y transporte del Distrito Federal.

¹ *Application Program Interface*

Abstract

The use of the *web* as a means of dissemination of maps is considered a breakthrough in mapping, *web* applications that work with maps are known as *Web Geographic Information Systems (Web GIS)*. The development of *Web GIS* solutions is very complex and takes a long time so this has led to the emergence of a number of API 's and Frameworks used for the development of different types of GIS systems. These solutions enable effective implementation of GIS applications for users who understand the characteristics of each API and Framework used, therefore, this approach still requires learning time and integration of these tools, as well as an extensive knowledge in geospatial technologies to create specific functionality in the systems.

This paper proposes the design and development of a Framework with OpenSource tools, to expedite the development of *Web GIS* applications with less coding by the developer, in order to accomplish this it is proposed generate a *web* Front End where you can create and manage these *web* GIS applications dynamically without requiring too many configurations.

The methodology is based on the use of a 3-tier architecture: data, business logic and presentation, which will contain each GIS *web* application generated by the proposed Framework. The data layer includes access to geospatial information from various sources and managed from a map server, business logic layer consists of a *web* server capable to respond to requests from the client to deliver spatial data received from map server. Finally the presentation layer is responsible for delivering the maps at the client browser.

Worth mentioning that is implemented as a case study the development of a *Web GIS* application using the proposed Framework with geospatial information of roads and transport of Mexico City.

Agradecimientos

A mi familia. A mi novia. A mis directores. A mis amigos. Al CIC y al CONACYT.

Gracias.

Índice de Contenido

RESUMEN	III
ABSTRACT	IV
AGRADECIMIENTOS	V
ÍNDICE DE CONTENIDO.....	VI
ÍNDICE DE FIGURAS	X
CAPÍTULO 1. INTRODUCCIÓN	1
1.1. Introducción.....	1
1.2. Planteamiento del Problema.....	2
1.3. Justificación	3
1.4. Objetivos	4
1.4.1. Objetivo General	4
1.4.2. Objetivos Particulares	4
1.5. Organización de la Tesis	5
CAPÍTULO 2. ESTADO DEL ARTE	6
2.1. Tecnología <i>Web</i> SIG.....	6
2.1.1. <i>Web mapping</i>	7
2.1.1.1. Ventajas y Desventajas de Mapas Web	7
2.1.1.1.1. Ventajas	7
2.1.1.1.2. Desventajas	9
2.1.2. Servidor de mapas en Internet.....	11
2.1.3. Arquitectura Web SIG.....	12

2.1.3.1. Arquitectura de cliente ligero	13
2.1.3.2. Arquitectura de cliente pesado.....	15
2.1.4. Transferencia de Datos Geoespaciales.....	16
2.2. <i>Frameworks</i> comerciales relacionados.....	18
2.2.1. <i>Frameworks</i> del lado del servidor.....	19
2.2.1.1. Mapserver	20
2.2.1.2. GeoServer.....	22
2.2.1.3. GeoTools	24
2.2.2. <i>Frameworks</i> del lado del Cliente.....	25
2.2.2.1. OpenLayers.....	25
2.2.3. Frameworks del lado del Cliente y del Servidor	26
2.2.3.1. MapBender	27
2.2.3.2. Geomajas.....	29
2.2.3.3. Mapfish	31
CAPÍTULO 3. MARCO TEÓRICO.....	33
3.1. Sistema de Información Geográfica (SIG)	33
3.1.1. Características especiales de la información geográfica	36
3.1.1.1. Representación de la información geográfica	36
3.1.1.2. Procesamiento de la Información Geográfica.....	38
3.1.1.2.1. Funciones de Análisis Espacial.....	38
3.1.1.3. Visualización de la información geográfica.....	40
3.1.1.4. Arquitectura del Sistema y otras cuestiones.....	41
3.1.2. Bases de Datos Espaciales.....	42
3.1.2.1. Datos espaciales	43
3.1.2.1.1. Puntos.....	44
3.1.2.1.2. Líneas.....	44
3.1.2.1.3. Polígonos.....	44
3.2. Estándares internacionales para los SIG.....	45
3.2.1. ISO y OGC	45
3.2.1.1. Especificaciones OGC	47
3.2.1.1.1. WMS	47
3.2.1.1.2. WFS.....	48
3.2.1.1.3. WCS.....	49
3.2.2. Arquitectura genérica para los SIG.....	51

3.3. <i>Framework</i>	52
3.3.1. Problemas de los <i>frameworks</i>	54
CAPÍTULO 4. METODOLOGÍA	55
4.1. Enfoque general de solución	55
4.2. Arquitectura propuesta	57
4.2.1. Capa de Datos	59
4.2.1.1. GeoServer.....	60
4.2.1.2. PostgreSQL/PostGIS.....	61
4.2.1.3. Shapefile.....	63
4.2.2. Capa Lógica de Negocio	64
4.2.2.1. Servidor Web Apache Tomcat	65
4.2.2.2. Módulo de Análisis Espacial.....	65
4.2.2.1.1. Java Server Pages (JSP)	66
4.2.2.1.2. Java Topology Suite (JTS).....	67
4.2.3. Capa de Presentación.....	68
4.2.3.1 Despliegue de Mapas	70
4.2.3.1.1 OpenLayers.....	70
4.2.3.1.2. ExtJS.....	71
4.2.3.1.3. GeoExt.....	72
CAPÍTULO 5. RESULTADOS.....	74
5.1. Diseño del Sistema	74
5.1.1. Diagrama de Caso de Uso	74
5.1.2. Diagramas de Secuencia	75
5.1.2.1. Diagrama de Secuencia: Ejecutar Framework	75
5.1.2.2. Diagrama de Secuencia: Detener Framework	76
5.1.2.3. Diagrama de Secuencia: Crear Aplicación.....	76
5.2. Ejecutando el <i>framework</i>	77
5.3. Administrando las Fuentes de Datos Geoespaciales	78
5.4. Caso de Estudio	81
5.3.1. Dibujado de elementos espaciales.....	86

5.3.2. Panel de Selección de Capas	87
5.3.3. Panel de Leyendas	88
5.3.4. Constructor de Consultas	89
5.5. Comparación de Resultados	90
CAPÍTULO 6. CONCLUSIONES.....	93
6.1. Alcances y Limitaciones	94
6.2. Trabajo a futuro	95
CAPÍTULO 7. REFERENCIAS.....	97
ANEXOS.....	101
A.1. Ejecutando el Framework	101

Índice de Figuras

Figura 1 Arquitectura general de un modelo típico Web SIG.....	12
Figura 2 Arquitectura de cliente ligero (Aplicaciones del lado del Servidor)	14
Figura 3 Arquitectura de cliente pesado (Aplicaciones del lado del Cliente)	16
Figura 4 Interacción de los Frameworks Web SIG.....	19
Figura 5 Arquitectura básica de Mapserver	21
Figura 6 Arquitectura Básica de Geoserver	23
Figura 7 Componentes que integran GeoTools.....	24
Figura 8 Interacción de OpenLayers con los distintos servicios.....	26
Figura 9 Arquitectura de Servicios de Geomajas.....	30
Figura 10 Dependencias de Geomajas.....	31
Figura 11 Arquitectura de MapFish.....	32
Figura 12 Partes de un Sistema de Información Geográfica	35
Figura 13 Representación de las Capas Temáticas de un SIG.....	37
Figura 14 Servicios OGC en la Arquitectura SIG.....	52
Figura 15 Arquitectura Entrada-Salida del Framework	55
Figura 16 Representación de Back End y Front End	56
Figura 17 Arquitectura del Framework Web SIG	58
Figura 18 Arquitectura del funcionamiento de Geoserver.....	60
Figura 19 Funcionamiento de JSP	67
Figura 20 Interacción OpenLayers con Servicios de Mapas.....	71
Figura 21 Dependencias de la Arquitectura Propuesta	73
Figura 22. Diagrama General de Caso de Uso.....	75
Figura 23. Diagrama de Secuencia: Ejecutar Framework	76
Figura 24. Diagrama de Secuencia: Detener Framework.....	76
Figura 25. Diagrama de Secuencia: Crear Aplicación	77
Figura 26. Página de inicio del Framework	78
Figura 27. Acceso a Geoserver	79
Figura 28. Página inicial Geoserver.....	80
Figura 29. Menú principal de Geoserver.....	81
Figura 30. Crear nueva Web SIG	82
Figura 31. Formulario para crear nueva Web SIG	83

Figura 32. Aplicación creada de Caso de Estudio.....	85
Figura 33. Dibujando Polígonos y Líneas	86
Figura 34 Panel de Selección de Capas.....	87
Figura 35 Panel de Leyendas	88
Figura 36. Constructor de Consultas	89
Figura 37. Resultados Consulta	90
Figura 38. Archivo RAR del Framework.....	101
Figura 39. Archivos incluidos dentro del RAR del Framework	102
Figura 40. Ubicación del archivo startup.bat.....	103
Figura 41. Apache Tomcat y Postgres ejecutados	104

Capítulo 1. Introducción

En este capítulo se aborda la introducción a la tesis, así como el planteamiento general del problema y la justificación de este trabajo. Se describen los objetivos, tanto el general como los particulares. Además se presentan los alcances y las limitaciones de la misma.

1.1. Introducción

El uso de la *web* para el análisis e interpretación de cualquier tipo de información susceptible a ser representada espacialmente, ha ido creciendo recientemente. La posibilidad de modificar, compartir y actualizar la información de manera remota, en un contexto geográfico, son sólo algunas de las ventajas.

Las aplicaciones que trabajan con datos geográficos son mejor conocidas como Sistemas de Información Geográfica (SIG), el cual es un tipo de sistema que permite capturar, almacenar, manipular y desplegar la información geográfica. Se caracterizan por requerir de configuraciones para los diversos servicios que utilizan, ya sea acceso a mapas, servidores para la visualización de los mismos, etc. Hoy en día el desarrollo de SIG basados en la *web* es complicado y, a pesar de existir muchas herramientas para la creación de los mismos, no deja de significar un arduo trabajo de configuración, levantamiento de servicios, integración de múltiples API, etc.

En el desarrollo de sistemas de información, y no solo de SIG, es fundamental disponer de un entorno común de trabajo (*framework*) en el cual los aplicativos se realicen de forma fiable y rápida, aprovechando las ventajas de lo que se conoce

como RAD (*Rapid Application Development* - desarrollo rápido de aplicaciones -). Dado que la filosofía de un *framework* es facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Teniendo en mente esto, en este trabajo se diseñarán y elaborarán herramientas que conformarán un *framework* para permitir el desarrollo de SIG para la *web* totalmente funcionales, de una manera ágil y siempre apeándose a los estándares establecidos por la OGC (*Open Geospatial Consortium*) la cual tiene como finalidad la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica y de la *Web*, facilitando el intercambio de la información geográfica.

Por último cabe mencionar que se implementará un caso de estudio utilizando datos geográficos sobre vialidades y transporte en la Ciudad de México para demostrar las ventajas y desventajas del *Framework* desarrollado en cuanto a tiempos y complejidad se refiere.

1.2. Planteamiento del Problema

Los SIG se componen de una integración organizada de software, hardware y datos geográficos, diseñados para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada. Para la integración de todos estos componentes en la *web*, se requiere de distintas herramientas que diversas compañías se han dedicado a desarrollar, tales como API y *Frameworks* para ayudar en la elaboración de SIG, pero esto mismo ha conducido a que hoy en día existan una gran cantidad de estas herramientas, haciendo que al desarrollador se le dificulte la selección, la integración y configuración de las mismas.

Por otra parte al trabajar con SIG se encuentran dificultades, como pueden ser las configuraciones de los servicios, de base de datos, el acceso a mapas y servidores para la visualización de los mismos. Demasiados conceptos, pocas guías orientadas a usuarios sin experiencia, hacen que el desarrollo de estos sistemas sea complejo y requieran de una extensa investigación.

La idea en este trabajo es permitir que los usuarios puedan desarrollar sistemas geográficos de forma sencilla, pudiendo hacerlo sin tener experiencia previa en el tema y sin un estudio profundo del mismo, facilitando de forma considerable el desarrollo de prototipos funcionales completos.

1.3. Justificación

Actualmente cuando se desarrolla un SIG basado en la *web*, el desarrollador normalmente gasta una gran parte de tiempo en la instalación, configuración e interconexión de las diferentes API, plataformas, BDE (Bases de Datos Espaciales) necesarias para lograr el completo funcionamiento de un SIG. Por ese motivo, se necesita de un entorno completo de trabajo que no requiera de muchas configuraciones ni instalaciones y que el tiempo usado se centre sólo en el desarrollo del propio SIG.

Por otra parte, al elaborar un SIG es esencial el trabajo con datos geográficos, o comúnmente llamados datos geoespaciales, con lo cual el desarrollador se encuentra con distintas áreas de estudio que a menudo son extensas, pero son imprescindibles para el análisis, manipulación, captura y despliegue de dichos datos geoespaciales. Dicho lo anterior, el problema aparece cuando se necesita destinar bastante tiempo al estudio y comprensión de esas áreas para el correcto desarrollo de un SIG. Es por ello que se plantea el desarrollo de la tesis aquí presentada con la finalidad de obtener desarrollos de SIG sobre la *web* de una manera más ágil y fácil.

1.4. Objetivos

1.4.1. Objetivo General

Generar una herramienta que agilice y facilite el desarrollo de sistemas de información geográfica basados en la *web*, evitando que el desarrollador tenga que profundizar en los conceptos entorno a la información geoespacial, así como en las configuraciones de las distintas plataformas necesarias; que además sea de alta interoperabilidad con otros *frameworks* para la *web*.

1.4.2. Objetivos Particulares

- Realizar el desarrollo de este *framework* integrando múltiples plataformas de desarrollo *web* existentes en el mercado de software libre o de código abierto.
- Proponer una Arquitectura Genérica para las aplicaciones web SIG creadas con la herramienta propuesta cumpliendo con los estándares de la Open Geospatial Consortium (OGC) para lograr interoperabilidad y comunicación entre las distintas tecnologías que trabajan con información geográfica.
- Desarrollo de una interfaz web (*front end*²) para la creación de las aplicaciones Web SIG y así minimizar el esfuerzo en cuanto a escritura de código de programación.
- Implementar una aplicación, como caso de estudio, con datos geográficos de vialidades y transporte de la Ciudad de México utilizando el *framework* desarrollado en esta tesis con la finalidad de verificar el desarrollo de una aplicación Web SIG.

² En diseño de software el front-end es la parte del software que interactúa con el o los usuarios

1.5. Organización de la Tesis

El resto del documento se encuentra organizado de la siguiente manera:

El Capítulo 2 describe el estado del arte y el enfoque de la investigación, se detallan algunos de los trabajos similares o relacionados con el de esta Tesis.

En el Capítulo 3 se presentan los fundamentos teóricos relacionados a los temas del conocimiento que se utilizan en este trabajo de Tesis.

En el Capítulo 4 se describe la metodología propuesta con la que se pretende llevar a cabo el desarrollo del presente trabajo de Tesis.

El Capítulo 5 presenta los resultados obtenidos a partir de la metodología propuesta y así como un caso de estudio para comparar el trabajo realizado con los existentes en el mercado.

En el último capítulo, el Capítulo 6, se muestran las conclusiones del trabajo haciendo énfasis en las aportaciones del trabajo, los alcances y limitaciones del mismo; así como los trabajos a futuro, tomando como base el trabajo de esta tesis.

Por último se enlistan las referencias bibliográficas utilizadas y consultadas para la elaboración de esta tesis.

Capítulo 2. Estado del Arte

En este capítulo se muestran antecedentes, trabajos existentes y estado actual de los conocimientos relacionados con los temas abordados en este trabajo de tesis.

Cabe mencionar que existen diversos *frameworks* para el desarrollo de aplicaciones *Web SIG* similares al de esta tesis por lo cual se analizarán y discutirán las ventajas y desventajas de cada uno.

2.1. Tecnología *Web SIG*

El desarrollo de la *web* y la expansión del Internet proveen 2 principales capacidades que pueden ayudar en gran medida a las personas dedicadas al estudio de la cartografía³. Primero, la *web* permite interacción visual con los datos, mediante un servidor *web*, clientes pueden producir mapas, mientras estos y sus tablas son publicados en Internet, otros clientes pueden ver estas actualizaciones, ayudando en aumentar la velocidad de los procesos de evaluación. Segundo, debido a la naturaleza de Internet, los datos geoespaciales pueden ser ampliamente accesibles, clientes pueden trabajar estos datos desde casi cualquier sitio (Gillavry, 2000).

Sin embargo, los *Web SIG* no dejan de tener ciertos defectos. El problema primario es la velocidad, los *SIG* dependen ampliamente del uso de gráficos, las velocidades de conexión sobre Internet pueden hacerse pesadas con el uso de gráficos, siendo intolerablemente lento para los usuarios.

³ Cartografía. es la ciencia que se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales y demás

2.1.1. *Web mapping*

Web mapping es el subproceso que lleva a cabo cualquier aplicación *Web* SIG que consta del diseño, implementación, generación y entrega de mapas hacia el navegador *web* del cliente como lo menciona (Fu P. & J. Sun 2001).

El uso de la *web* como medio de difusión de mapas puede ser considerado como un gran avance en la cartografía y abre muchas nuevas oportunidades, tales como mapas en tiempo real, difusión más barata, actualizaciones de datos y software más frecuentes y más baratas, contenido de mapas personalizado, orígenes de datos distribuidos e intercambio de información geográfica. Asimismo, (Andreas Neuman, 2007) nos menciona que implica muchos desafíos debido a las restricciones técnicas (baja resolución de pantalla y ancho de banda limitado, sobre todo en dispositivos informáticos móviles, muchos de los cuales son físicamente pequeños, lentos y utilizan conexiones inalámbricas de Internet lentas) y los problemas de seguridad, problemas de fiabilidad y complejidad técnica. Aunque los mapas *web* primero fueron principalmente estáticos, hoy en día pueden ser totalmente interactivos e integrar múltiples medios.

2.1.1.1. Ventajas y Desventajas de Mapas Web

2.1.1.1.1. Ventajas

- Los mapas sobre la *web* pueden ofrecer información actualizada. Si los mapas se generan automáticamente a partir de bases de datos, que pueden mostrar información en casi tiempo real. Estos no necesitan ser impresos, materializados y distribuidos. Ejemplos:

- Un mapa que mostrando resultados de la elección, tan pronto como los resultados de las elecciones estén disponibles.
 - Un mapa que muestra la situación del tráfico en tiempo real mediante el uso de cerca los datos de tráfico recogidos por las redes de sensores.
 - Un mapa que muestra las ubicaciones actuales de los vehículos de transporte público, como autobuses o trenes, lo que permite a los clientes reducir su tiempo de espera en las paradas o estaciones, o estar al tanto de las demoras en el servicio.
- Software y hardware para la infraestructura de mapas *web* son de bajo costo.
 - Las actualizaciones de productos se pueden distribuir fácilmente. Debido a que los mapas *web* distribuyen tanto la lógica y los datos con cada solicitud o carga, actualizaciones de producto puede ocurrir cada vez que el usuario vuelve a cargar la aplicación *web*. En la cartografía tradicional, cuando se trata de mapas impresos o mapas interactivos distribuidos en los medios de comunicación fuera de línea (CD, DVD, etc), una actualización de mapas causa serios esfuerzos, lo que provoca una reimpresión o remasterización, así como una redistribución de los medios de comunicación. Con mapas *web*, datos y actualizaciones de productos es más fácil, más barato y más rápido, y puede ocurrir con más frecuencia.
 - Trabajan en todos los navegadores y sistemas operativos. Si los mapas *web* están implementados con base en estándares abiertos, el sistema operativo subyacente y el navegador no importan.
 - Mapas *web* pueden combinar fuentes de datos distribuidas. El uso de estándares abiertos y API documentadas se puede integrar (mashups)

diferentes fuentes de datos, si el sistema de proyección, escala del mapa y la calidad de los datos lo permiten. El uso de fuentes de datos centralizadas elimina la carga para las organizaciones individuales para mantener copias de los mismos conjuntos de datos. Lo malo es que uno tiene que confiar y confiar en las fuentes de datos externas.

- Admiten hipervínculos a otras informaciones en la *web*. Al igual que cualquier otra página *web*, mapas *web* puede actuar como un índice de otra información en la *web*. Cualquier área sensible en un mapa, un texto de la etiqueta, etc puede proporcionar enlaces a información adicional. A modo de ejemplo, un mapa que muestra las opciones de transporte público se puede conectar directamente a la sección correspondiente en la tabla horarios de trenes en línea.

2.1.1.1.2. Desventajas

- Problemas de fiabilidad - la fiabilidad de la Internet y la infraestructura de servidores *web* no es todavía lo suficientemente bueno. Especialmente si un mapa *web* se basa en las fuentes de datos externas, distribuidos, el autor original a menudo no puede garantizar la disponibilidad de la información.
- Datos geográficos son caros - A diferencia de los Estados Unidos, donde recolectar datos geográficos por las instituciones gubernamentales suele estar disponible de forma gratuita o a bajo costo, Datos Geográficos suelen ser muy caros en otras partes del mundo.
- Problemas de ancho de banda - mapas *web* por lo general necesitan un ancho de banda relativamente alto.

- Espacio limitado de la pantalla - Los mapas *web* tienen el problema del espacio limitado de la pantalla. Esto es especialmente un problema para los mapas *web* y servicios móviles basados en la localización donde los mapas tienen que aparecer en las pantallas muy pequeñas con resoluciones tan bajas como 240x320 píxeles. Con suerte, los avances tecnológicos ayudan a superar estas limitaciones.
- Calidad y precisión de los temas - Muchos mapas *web* son de mala calidad, tanto en la precisión de simbolización, contenido y datos.
- Complejo para el desarrollo - A pesar de la creciente disponibilidad de herramientas libres y comerciales para crear mapas *web* y aplicaciones web SIG, sigue siendo una tarea compleja para crear mapas *web* interactivos. Muchas tecnologías, servicios, módulos y fuentes de datos deben ser dominadas e integradas.
- Herramientas inmaduras de desarrollo - en comparación con el desarrollo de aplicaciones independientes con herramientas de desarrollo integradas, los entornos de desarrollo y depuración de un conglomerado de diferentes tecnologías *web* sigue siendo difícil e incómodo.
- Los derechos de autor - Muchas personas todavía se resisten a publicar datos geográficos, especialmente en vista del hecho de que Los datos geográficos son caros en algunas partes del mundo. Temen a las infracciones de derechos de autor de otras personas que utilizan sus datos sin solicitudes apropiadas de permiso.

- Los problemas de privacidad - Con información detallada disponible y la combinación de fuentes de datos distribuidas, es posible descubrir y combinar una gran cantidad de información privada y personal de las personas individuales. Inmuebles y propiedades de los individuos son ahora accesibles a través de alta resolución e imágenes de satélite de todo el mundo.

Con lo anterior mencionado podemos determinar que actualmente es imprescindible contar con aplicaciones que permitan el manejo de mapas sobre la *web* y a pesar de las restricciones existentes la tendencia de la cartografía fluye hacia el contexto de Internet, lo difícil viene al momento del desarrollo de estas aplicaciones SIG en la *web*, que como ya se mencionó en algunos puntos anteriores aun es bastante complejo el manejo de tantas tecnologías que aún siguen en desarrollo para la elaboración de estas aplicaciones.

2.1.2. Servidor de mapas en Internet

Un servidor de mapas en Internet o IMS (Internet Map Server, *por sus siglas en inglés*) permite a las aplicaciones SIG basadas en la *web* hacer fácilmente sus datos espaciales accesibles a través de navegador *web* para los usuarios finales. Hacer accesible la información a todo el mundo es ciertamente factible y cualquier organización que tenga un sitio *web* público, sin duda puede añadir un IMS sin abrir demasiados agujeros de seguridad adicionales.

(Peterson, M. P., 2003) nos dice que para trabajar con un IMS, el software requiere de dos componentes para su funcionamiento. Un motor de procesamiento de datos geoespaciales que se ejecuta en el lado del servidor como un servicio, servlet o aplicaciones Common Gateway Interface (CGI) y procesar los datos espaciales

crudos en un mapa y un servidor *web* estándar que gestione las peticiones de entrada y responda con los datos apropiados al navegador del cliente o de la ventana de la aplicación. El producto final es ya sea una imagen JPEG o GIF o un vector, que se transmite de vuelta al navegador o un flujo de datos que es interpretado por un plug-in para el navegador del cliente.

2.1.3. Arquitectura Web SIG

El desempeño de las tareas de análisis SIG, Web SIG es similar a la típica arquitectura de tres niveles de cliente/servidor. El geoprocesamiento⁴ se está rompiendo en tareas de servidor y de cliente. Un cliente normalmente es un navegador *web*. El lado del servidor se compone de un servidor *web*, una base de datos espacial y software Web SIG como se muestra (Helali 2001) en la *Figura 1*

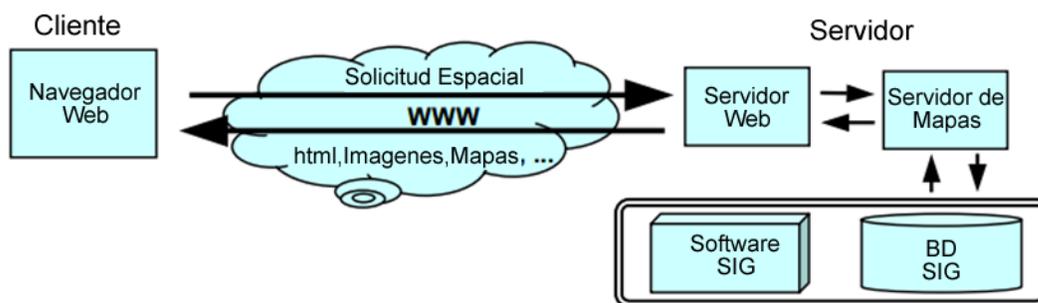


Figura 1 Arquitectura general de un modelo típico Web SIG

Este modelo de red ampliamente existe dentro de las empresas, en los que algunos ordenadores actúan como servidores y otros actúan como clientes. El servidor simplemente tiene ejecutando el SIG, y añade una interfaz en el lado del cliente y

⁴ El Geoprocesamiento es la ejecución metódica de una secuencia de operaciones en los datos geográficos para crear nueva información.

un *middleware*⁵ en el lado del servidor para la comunicación entre el cliente y el software de SIG.

El desarrollo reciente en la programación orientada a objetos hace posible la producción de componentes de software, y enviarlos al cliente antes de ejecutarlo en la máquina del cliente. Esto último viene siendo lo que se conoce como una arquitectura de cliente pesado. Una arquitectura de cliente pesado permite que la máquina del cliente haga la mayor parte de los trabajos de procesamiento localmente. Por otro lado también se tiene lo que es una arquitectura de cliente ligero, que al contrario de una arquitectura de cliente pesado, la primera le deja todo el trabajo de procesamiento al lado del Servidor.

2.1.3.1. Arquitectura de cliente ligero

La arquitectura de cliente ligero se utiliza en la arquitectura típica. En un sistema de cliente ligero, los clientes sólo tienen interfaces de usuario para comunicarse con el servidor y mostrar los resultados. Todo el procesamiento se realiza en el servidor en realidad, como se muestra en la [Figura 2](#). Los equipos del lado del servidor por lo general tienen más poder que el cliente y administran los recursos centralizados. Además, la funcionalidad principal está en el lado del servidor, en la arquitectura de cliente ligero también existe la posibilidad de que los programas útiles en el lado del servidor puedan vincularse con el software de servidor. La [Figura 2](#) muestra la comunicación entre el navegador *web*, servidor *web* y el servidor SIG. En el lado del servidor *web*, hay algunas posibilidades para realizar la conexión de los SIG para la World Wide Web (WWW); CGI, interfaz de programación de aplicaciones de servidor *web* (API), Active Server Pages (ASP),

⁵ Middleware es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos.

Java Server Pages (JSP) y servlets Java. Las posibilidades antes escritas son mencionadas por (Helali, 2001) pero en la actualidad, existen aún muchas más posibilidades para lograr un SIG del lado del servidor.

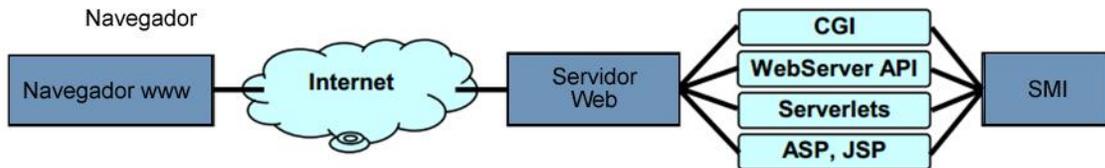


Figura 2 Arquitectura de cliente ligero (Aplicaciones del lado del Servidor)

El usuario del lado del cliente no necesita ningún conocimiento acerca de la vinculación con el servidor de mapas en el lado del servidor, pero los desarrolladores que administren el sistema o la aplicación deben estar familiarizados con estas técnicas.

Las principales ventajas de este modelo son las siguientes:

- Control central
- Fácil carga o actualización de datos
- Se mantiene la última versión
- Comúnmente más barato
- Posibilidades de Integración

Y desventajas tales como:

- No responde a las necesidades locales
- Grandes tamaños de datos al momento de transferir información
- Tiempos de respuesta lentos
- Menos interacción, en el lado del cliente hay limitaciones de aplicación y navegador

- Datos vectoriales no pueden ser mostrados del lado del cliente sin necesidad de algún plug-in⁶ adicional.

2.1.3.2. Arquitectura de cliente pesado

En general, un navegador *web* puede manejar documentos HTML y las imágenes de mapa de bits incrustados en los formatos estándar. Para hacer frente a otros formatos de datos como datos vectoriales, clips de vídeo o archivos de música, la funcionalidad del navegador tiene que ampliarse. Usando exactamente la misma comunicación cliente/servidor como en la arquitectura cliente ligero, el formato de archivos de vectores no se podía utilizar. Para superar este problema la mayoría de las aplicaciones de navegador ofrecen un mecanismo que permite a los programas de terceros trabajar junto con el navegador como un plug-in.

La funcionalidad de la interfaz de usuario ha pasado de una simple visualización documentos a mostrar aplicaciones más interactivas. De acuerdo a (Byong-Lyol, 1998) dicho progreso es el siguiente: HTML, CGI, utilizando formularios HTML y CGI, Java script para aumentar las capacidades de la interfaz de usuario, los applets⁷ de Java para proporcionar la funcionalidad del lado del cliente. Actualmente las capacidades de la interfaz de usuario se combinan con invocaciones remotas, véase la *Figura 3*.

⁶ Un plug-in es un complemento de una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

⁷ Un applet es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo en un navegador web.

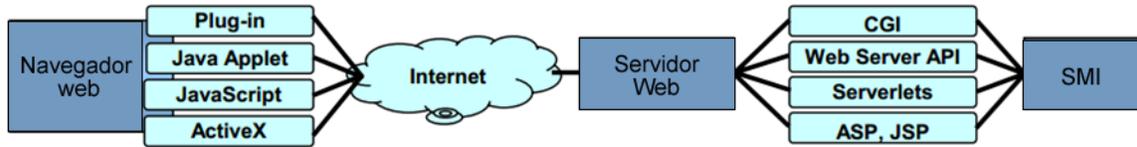


Figura 3 Arquitectura de cliente pesado (Aplicaciones del lado del Cliente)

Las principales ventajas de este modelo son:

- No se requieren estándares de documentos / gráficos
- Datos vectoriales pueden ser usados
- Calidad de la imagen no restringida a solo GIF o JPEG
- Posibilidad de Interfaces más intuitivas
- Mayor interacción con el usuario

Y por otra parte algunas de las desventajas son:

- Usuario requiere obtener software adicional
- Algunas plataformas o navegadores a veces no son compatibles

2.1.4. Transferencia de Datos Geoespaciales

Una cuestión decisiva para el uso de SIG en la *web* es el formato de los datos (vector o raster) los cuales son usados para transferir dichos datos hacia el cliente. Para la transmisión de los datos hacia el cliente, el mapa es convertido en raster⁸ o

⁸ Imagen raster es una estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color, denominada matriz, que se puede visualizar en un monitor, papel u otro dispositivo de representación.

en un formato adecuado de vectores⁹. Cuando un tipo de dato raster es transferido al cliente, cualquier navegador *web* sin extensión alguno puede ser usado, mientras el navegador muestre imágenes GIF o JPEG. Lo anterior significa que los datos en el servidor tienen que ser convertidos a un formato raster. La desventaja de usar datos raster es la falta de comodidad de manejo sobre aspectos cartográficos, si pasamos el puntero del mouse no podemos seleccionar o resaltar elementos individuales como lo expresa (Gaffuri, J., 2012). Además de lo anterior, la comunicación con el servidor es necesaria por cada petición del cliente.

Por otra parte mientras el tamaño de los datos vectoriales es menor, estos se transmiten más rápido que los raster. Los datos vectoriales son manejados por un navegador *web* estándar con alguna extensión (por ejemplo, el uso de *plug-ins*). El usuario obtiene una mayor funcionalidad con datos vectoriales. Por ejemplo, los objetos individuales se pueden seleccionar. (Nayak, 2000) menciona que una ventaja más de la utilización de los datos vectoriales es la posibilidad de procesamiento local; no es necesario ponerse en contacto con el servidor por cada acción ejecutada del navegador. La cantidad de datos vectoriales enviados a través de la *web* podría ser tres o cuatro veces menor que la cantidad de datos raster necesarios para la resolución equivalente que resulta en un tiempo más rápido de respuesta y una mayor productividad. Las desventajas de los datos vectoriales dependen de los fabricantes, así como, el cambio de volumen de datos; la cantidad de datos que varía con el área seleccionada.

La elección de la transferencia de datos (vector o raster) varía con las aplicaciones y las infraestructuras existentes. Los productos de software, que ofrecen transferencia de vector o raster opcional, pueden proporcionar ventajas. Pueden

⁹ Una imagen vectorial es una imagen digital formada por objetos geométricos independientes (segmentos, polígonos, arcos, etc.), cada uno de ellos definido por distintos atributos matemáticos de forma, de posición, de color, etc.

permitir una preselección con datos raster, y después, la carga de los datos vectoriales reales con la posibilidad de proceso posteriormente locales (Leukert y Reinhardt, 2000).

Diferentes consorcios han desarrollado formatos estándares para la transferencia de datos a través de Internet. El consorcio OGC, por ejemplo, desarrolló Geography Markup Language (GML). GML es un sublenguaje de XML descrito como una gramática en un esquema XML para el modelaje, transporte y almacenamiento de información geográfica. Su importancia radica en que a nivel informático se constituye como una lengua franca para el manejo y trasvase de información entre los diferentes software que hacen uso de este tipo de datos, como los SIG (OGC, 2003).

2.2. *Frameworks* comerciales relacionados

En el presente apartado se muestran los trabajos comerciales existentes relacionados o similares al del desarrollado en esta tesis, se analiza y comenta cada uno de ellos haciendo énfasis en las ventajas y desventajas de estos. Cabe mencionar que se abordarán 3 categorías generales para los *Frameworks* Web SIG, del lado del cliente, del lado del servidor y basados en cliente y servidor.

En la [Figura 4](#) se muestra cómo interactúan los Web SIG *Frameworks* tanto de lado del cliente como del servidor.

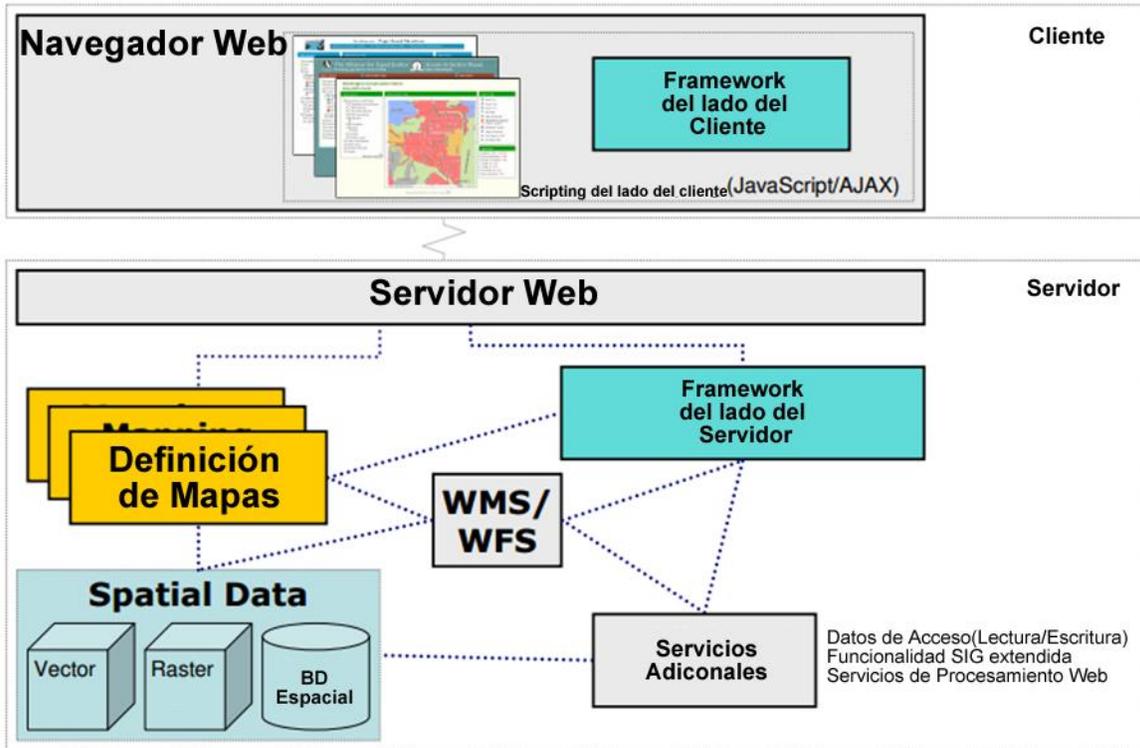


Figura 4 Interacción de los Frameworks Web SIG

2.2.1. Frameworks del lado del servidor

Por el lado del servidor más que un *Framework* como tal, nos encontramos principalmente con servidores de cartografía digital o Servidores de Mapas de Internet (IMS) la especificación estándar para estos servidores la provee la OGC ([WMS 1.3 OGC® 06-042, 2006](#)).

En la actualidad existen distintos desarrollos de este tipo de entornos de trabajo que son ya comúnmente usados y están bien definidos, a continuación se describen algunos de ellos.

2.2.1.1. Mapserver

MapServer es un entorno de desarrollo en código abierto (Open Source¹⁰ Initiative) para la creación de aplicaciones SIG en Internet/Intranet con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología IMS. MapServer no es un SIG completo, pero tampoco aspira a serlo.

MapServer fue desarrollado originalmente por la Universidad de Minnesota, Proyecto Fernet, en cooperación con la NASA y el Departamento de Recursos Naturales de Minnesota. Más tarde fue organizada por el alojado por TerraSIP, un proyecto patrocinado por la NASA entre la UMN y un consorcio de intereses por la gestión de la tierra.

MapServer es ahora un proyecto de OSGeo, y es mantenido por un número cada vez mayor de desarrolladores de todo el mundo. Es apoyado por un grupo diverso de organizaciones que brinda fondo de mejoras y de mantenimiento, y administrado dentro de OSGeo por el Comité Directivo del Proyecto MapServer compuesto por desarrolladores y otros colaboradores ([MapServer, 2013](#)). Sus principales características son:

- Que se ejecuta bajo plataformas Linux/Apache y Windows (MS4W)
- Formatos vectoriales soportados: ESRI shapefiles, PostGIS, ESRI ArcSDE, GML y otros muchos vía OGR.
- Formatos raster soportados: JPG, PNG, GIF, TIFF/GeoTIFF, EPPL7 y otros vía GDAL.
- Fuentes TrueType
- Configuración "al vuelo" vía parámetros GET pasados por URL

¹⁰ Open Source o Código abierto es la expresión con la que se conoce al software distribuido y desarrollado libremente.

- MapScript proporciona una API para poder acceder a las funcionalidades de MapServer mediante lenguajes de programación como PHP, Java, Perl, Python, Ruby o C#.
- Soporte de estándares interoperables y conformes con Open Geospatial Consortium, como WMS, SLD, WFS, WCS y SOS.

En la *Figura 5* muestra la arquitectura básica de Mapserver.

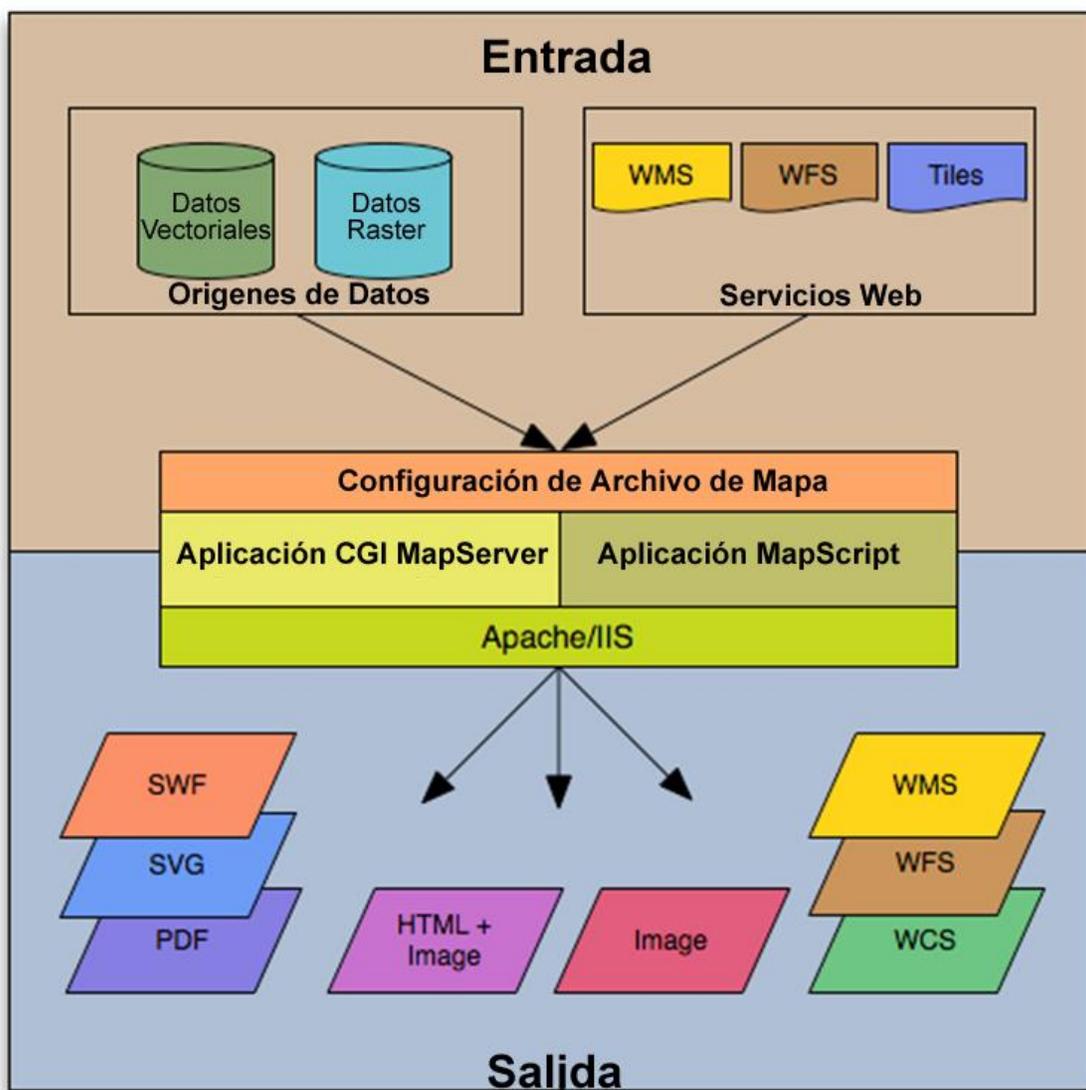


Figura 5 Arquitectura básica de Mapserver

Si bien Mapserver nos brinda una serie de herramientas para la visualización, consulta y análisis de información geoespacial y apegada a los estándares de la OGC, aun para la creación de una aplicación Web SIG completa hace falta de un entorno de trabajo en el lado del cliente para poder manejar esa información geoespacial entregada por MapServer.

2.2.1.2. GeoServer

GeoServer, al igual que Mapserver, es un servidor de mapas, pero con la diferencia de ser de código abierto, escrito en Java, permite a los usuarios compartir y editar datos geoespaciales. Diseñado para la interoperabilidad, publica datos de las principales fuentes de datos espaciales usando estándares abiertos. GeoServer ha evolucionado hasta llegar a ser un método sencillo de conectar información existente a globos virtuales¹¹ tales como Google Earth y NASA World Wind (véase así como mapas basados en *web* como OpenLayers, Google Maps y Bing Maps). GeoServer sirve de implementación de referencia del estándar Open Geospatial Consortium Web Feature Service, y también implementa las especificaciones de Web Map Service y Web Coverage Service ([Geoserver, 2013](#))

Entre las principales características de GeoServer se pueden citar algunas como:

- Compatible con las especificaciones WMS, WCS y WFS.
- Fácil utilización a través de la herramienta de administración vía *web*, no es necesario entrar en archivos de configuración grandes y complicados.

¹¹ Un globo virtual es un software que representa la Tierra o cualquier planeta. EL primer software que logró interesar al público en general fue Google Earth.

- Soporte amplio de formatos de entrada PostGIS, Shapefile, ArcSDE y Oracle. VFP, MySQL, MapInfo y WFS en cascada también están entre los formatos de entrada soportados.
- Soporte de formatos de salida tales como JPEG, GIF, PNG, SVG y GML.
- Soporte para edición de datos de banco de datos individuales a través del protocolo WFS transactional profile (WFS-T), disponible para todos los formatos de datos.
- Basado en servlets Java (JEE), puede funcionar en cualquier servlet contenedor.
- Diseñado para ser compatible con extensiones.
- Facilidad de escritura de nuevos formatos de datos con la interfaz de almacenamiento de datos GeoTools y clases de ayuda.

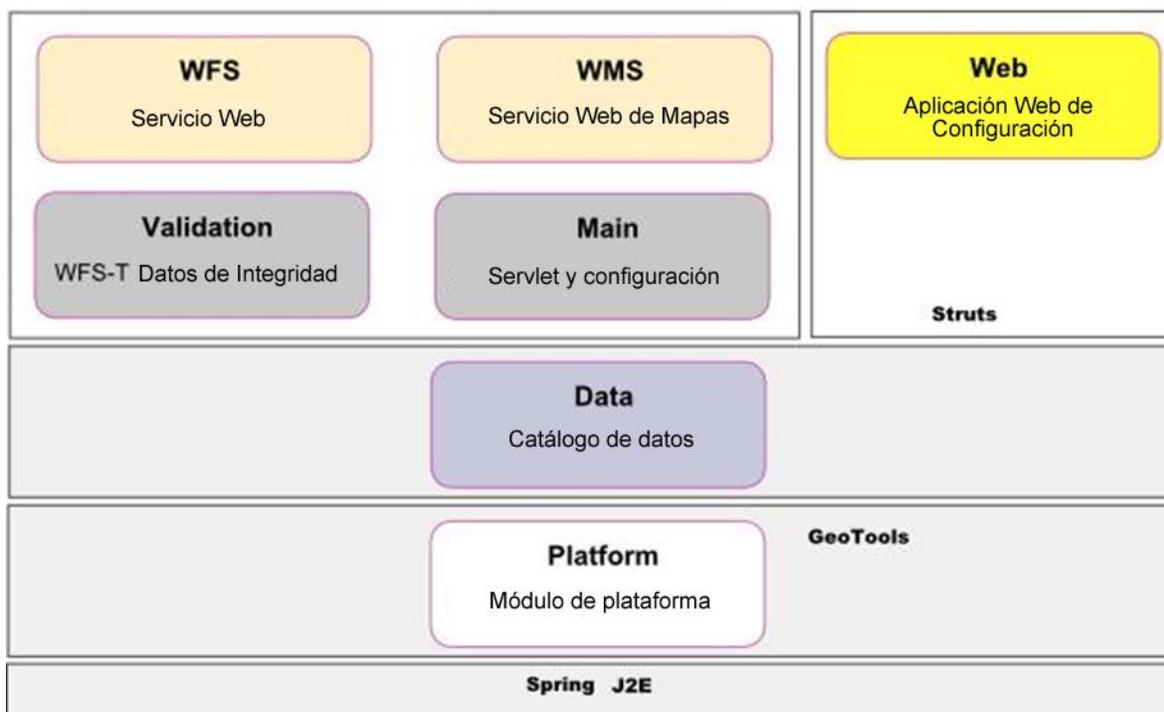


Figura 6 Arquitectura Básica de Geoserver

2.2.1.3. GeoTools

GeoTools es una biblioteca SIG de código libre que permite desarrollar soluciones adaptadas a los estándares. Proporciona una implementación de las especificaciones del Open Geospatial Consortium según van apareciendo.

Está escrito en el lenguaje de programación Java y se encuentra actualmente en un desarrollo activo al nutrirse de una comunidad de usuarios muy dinámica. Su diseño y concepción modular hace que numerosas implementaciones de software libre en el ámbito de los SIG hagan uso de los desarrollos de GeoTools (Geotools, 2013).

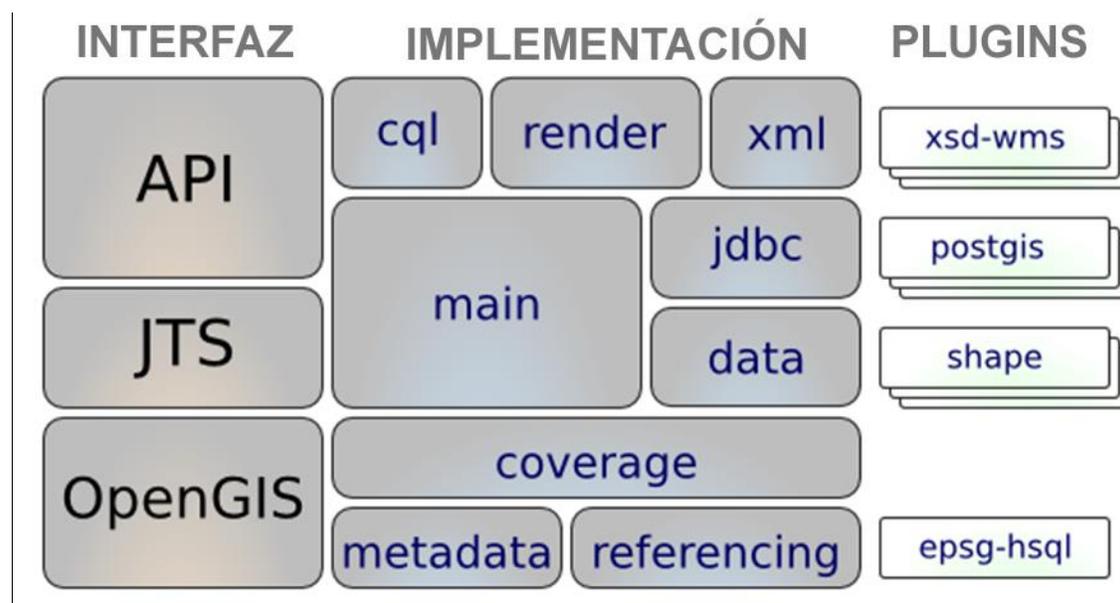


Figura 7 Componentes que integran GeoTools

GeoTools, al igual que MapServer, permite solo trabajar con la información geoespacial del lado del servidor, por lo cual para lograr una aplicación SIG completa se requiere aun desarrollar la aplicación del lado del cliente para que los

datos entregados por estas herramientas puedan ser desplegados, manipulados y actualizados del lado del cliente.

2.2.2. Frameworks del lado del Cliente

Por el lado del cliente nos encontramos con herramientas principalmente de *scripting* para la consulta, el análisis y la presentación de los datos geoespaciales en el navegador *web*, pero que por sí solos podrían brindarnos la capacidad de crear aplicaciones Web SIG , ya que, como se ha descrito anteriormente, se requiere de un servidor de mapas trabajando del lado del servidor y que no han sido dotados con la funcionalidad de poder realizar operaciones de análisis geoespacial sobre los datos, tal como lo hacen los *Frameworks* del lado del servidor.

2.2.2.1. OpenLayers

OpenLayers es una biblioteca de JavaScript de código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores *web*. OpenLayers ofrece un API para acceder a diferentes fuentes de información cartográfica en la red: Web Map Services, mapas comerciales (tipo Google Maps, Bing, Yahoo), Web Features Services, distintos formatos vectoriales, mapas de OpenStreetMap, etc, además no cuenta con dependencias de Servidor (OpenLayers, 2013).

Como se menciona es una herramienta que únicamente nos permite desplegar mapas en la *web* obtenidos de diferentes repositorios de datos geoespaciales.

En la *Figura 8* podemos apreciar cómo OpenLayers puede interactuar con los

distintos servicios estandarizados por la OGC, GeoServer y repositorios de bases de datos espaciales.

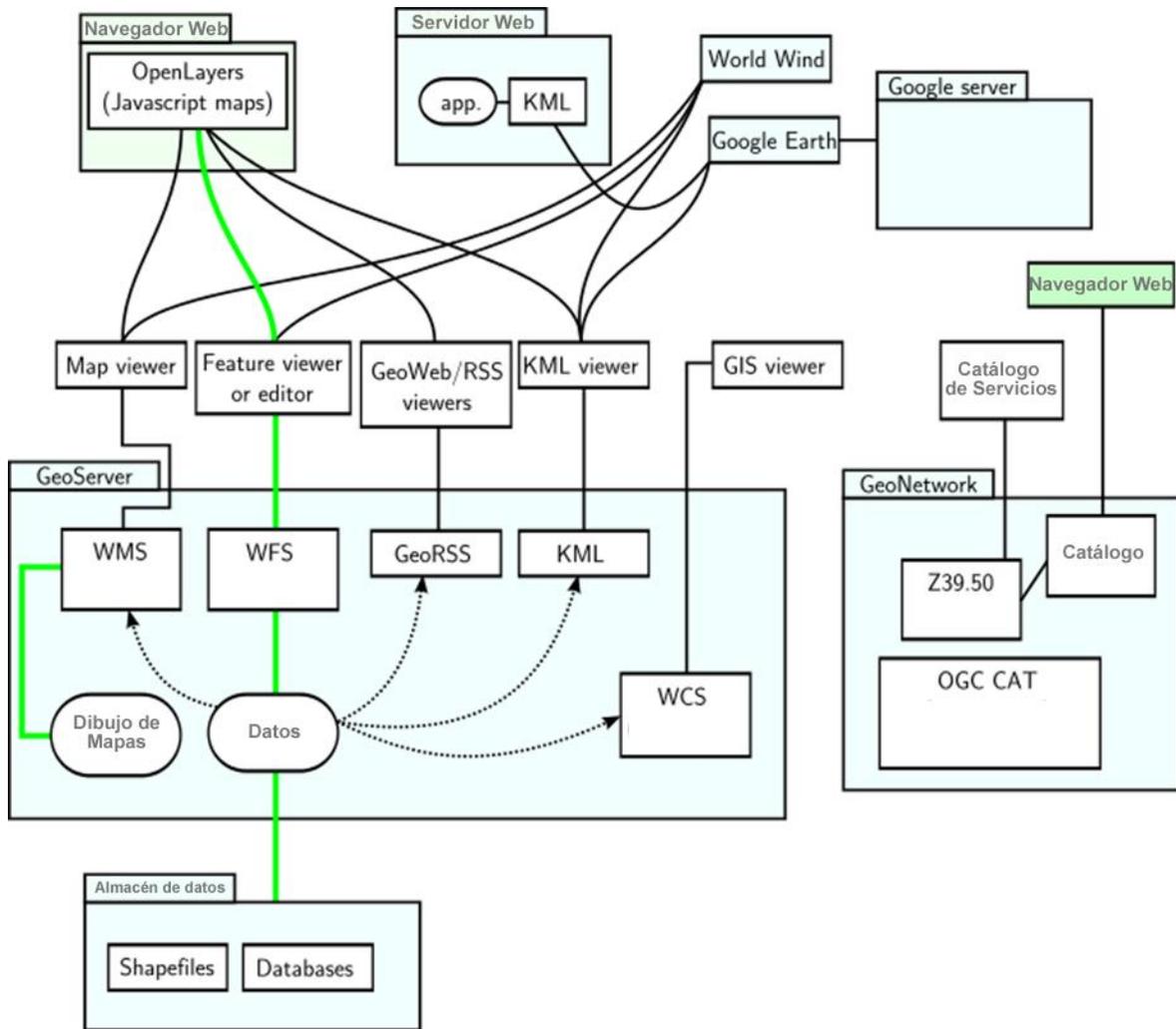


Figura 8 Interacción de OpenLayers con los distintos servicios

2.2.3. Frameworks del lado del Cliente y del Servidor

En esta parte se describen y comentan algunos de los desarrollos existentes de *Frameworks* que abarcan tanto el lado del cliente como del servidor para la creación de aplicaciones Web SIG, el trabajo de esta tesis pertenece a este grupo de

Frameworks donde sin necesidad de requerir alguna herramienta adicional puedan realizarse aplicaciones SIG completas y totalmente funcionales para la *web*.

2.2.3.1. MapBender

MapBender es un proyecto graduado en OSGeo. Consiste de un entorno de geoportales para publicar, registrar, visualizar, navegar, seguir y conceder accesos seguros a servicios de Infraestructura de Datos Espaciales (IDE).

MapBender es la punta del iceberg, una metacapa provista por un software que proporciona acceso a los componentes de una IDE al adherirse a los estándares internacionales de interoperabilidad para servicios de información geográfica vía *web*. MapBender permite a los usuarios crear navegadores *web* de mapas o geoportales personalizados fácilmente, a partir de una amplia gama de aplicaciones o widgets con mínima o ninguna necesidad de programación. Los *widgets*, integrados con funcionalidades de tipo servidor, proveen funcionalidades avanzadas como seguridad de tipo *proxy*, digitalización en línea a través del navegador, auto *snapping* y más.

La gestión de aplicaciones en MapBender es directa y fácil de usar ofreciendo una destacada cantidad de funcionalidades enfocadas para la operación a gran escala de los componentes distribuidos de la IDE. Es la herramienta de elección para los administradores de IDE's que necesitan mantener, categorizar y controlar servicios *web* de mapas y entidades (*features*). MapBender permite gestionar el acceso a personas, grupos y administradores. La adhesión a los estándares de los servicios, tales como WMS y WFS Transaccional de OGC (Open Geospatial Consortium) le permite sacar provecho de los servicios interoperables de una multitud de plataformas disponibles a nivel de servidor ([Mapbender, 2013](#)).

El lado del cliente de MapBender se basa en HTML y JavaScript usando las bibliotecas jQuery y jQuery UI, que proporcionan gran cantidad de aplicaciones tipo *widgets* al navegador *web*. El lado del servidor Mapbender está implementado en PHP y PostgreSQL/PostGIS.

MapBender da una nueva perspectiva a la gestión de las IDE al utilizar nada más que un navegador *web* estándar. Sus propiedades principales son:

- Es sencillo crear una interfaz *web* de mapas utilizando plantillas de aplicaciones
- Agregar/eliminar/personalizar aplicaciones jQuery tipo *widgets*
- Personalizar estilos usando jQuery UI Themeroiler
- Cargar y compartir servicios remotos OGC de tipo WMS (mediante cacheo de capacidades o Capabilities caching)
- Organizar capas para superponer los servicios WMS cargados
- Editar la visibilidad de las capas
- Desplegar servicio de metadatos (según perfil ISO) basado en el documento de capacidades para catálogo según protocolo OGC con interfaz CSW (Catalog Service Web).
- Cargar y configurar Servicios WFS para buscar, encontrar y resaltar objetos o elementos espaciales (es equivalente al servicio Gazetteer o Nomenclátor).
- Enlazar de forma transaccional servicios OGC WFS y WMS para permitir la digitalización “en-línea” (con funcionalidades como el servicio cruzado de snapping, o la manipulación de geometrías como recorte (split), combinación o clonación).

- Crear usuarios y grupos, para compartir y conceder accesos a los servicios y módulos de forma personalizada.
- Manipular Web Map Context (WMC)
- Conectar a servicios remotos de catálogos, y de forma opcional a Geonetwork Opensource.
- Restringir acceso a usuarios con el módulo de seguridad proxy OWS vía servidores validados.
- Control de protocolos encriptados y notificación de estado y auto-actualizaciones de servicios desde un servicio OGC remoto.
- Desplegar el cliente OpenLayers desde el repositorio de servicios de Mapbender.
- Imprimir mapas de alta calidad usando tus propias plantillas PDF.

2.2.3.2. Geomajas

Geomajas es un *Framework* Web SIG libre y de código abierto para la creación de aplicaciones ricas en Internet. Cuenta con sofisticadas capacidades de visualización y gestión de la información geoespacial. El diseño modular hace que sea fácilmente ampliable. La arquitectura cliente-servidor garantiza una escalabilidad grande. El enfoque de Geomajas es proporcionar una plataforma para la integración del servidor de datos geoespaciales, lo que permite que varios usuarios puedan controlar y gestionar los datos desde sus propios navegadores. En esencia, Geomajas proporciona un conjunto de bloques de construcción de gran alcance, de la que las aplicaciones de SIG más complejos se pueden construir fácilmente (Geomajas, 2013). Las características clave incluyen:

- Arquitectura modular
- Programas claramente definidos

- Integrado arquitectura cliente-servidor
- La seguridad integrada
- Geometría avanzada y de edición de atributos de validación
- Definiciones de atributos personalizados, incluyendo las relaciones de objeto
- Las capacidades avanzadas de consulta (búsqueda, filtros, estilo, etc.)

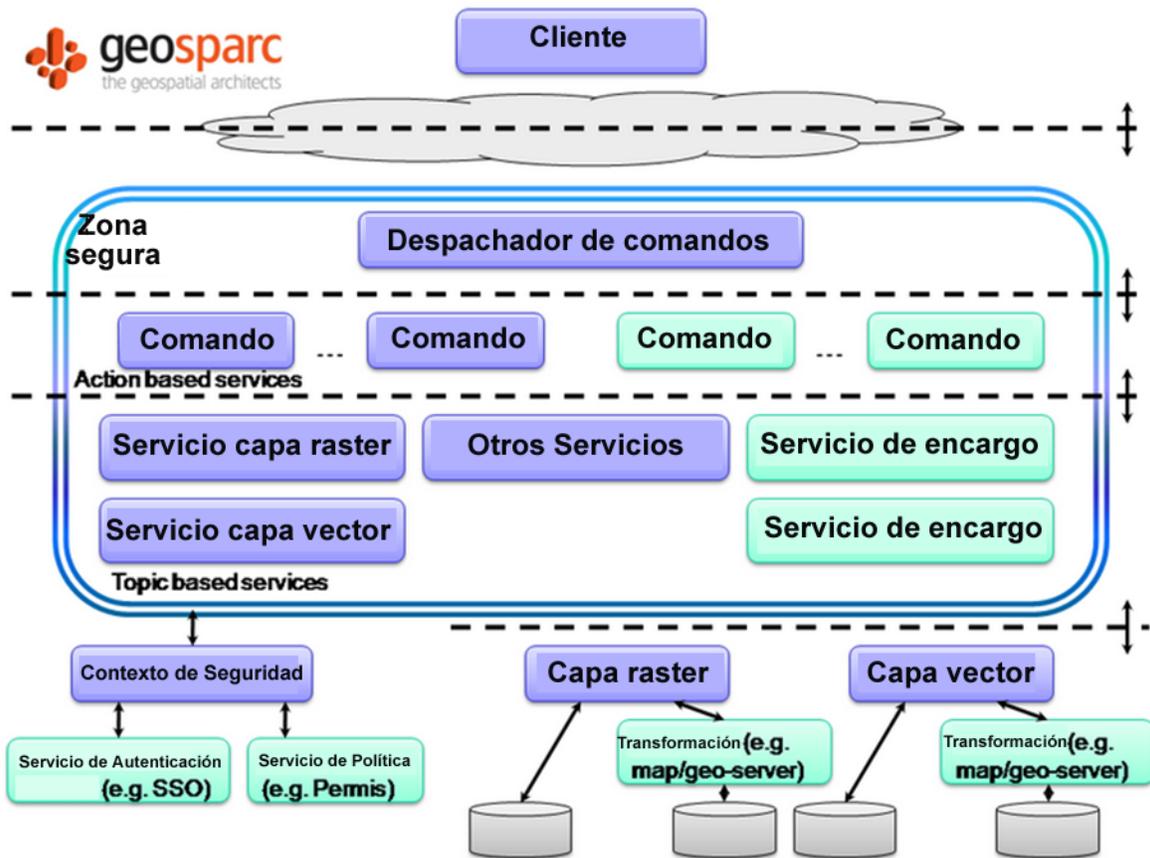


Figura 9 Arquitectura de Servicios de Geomajas

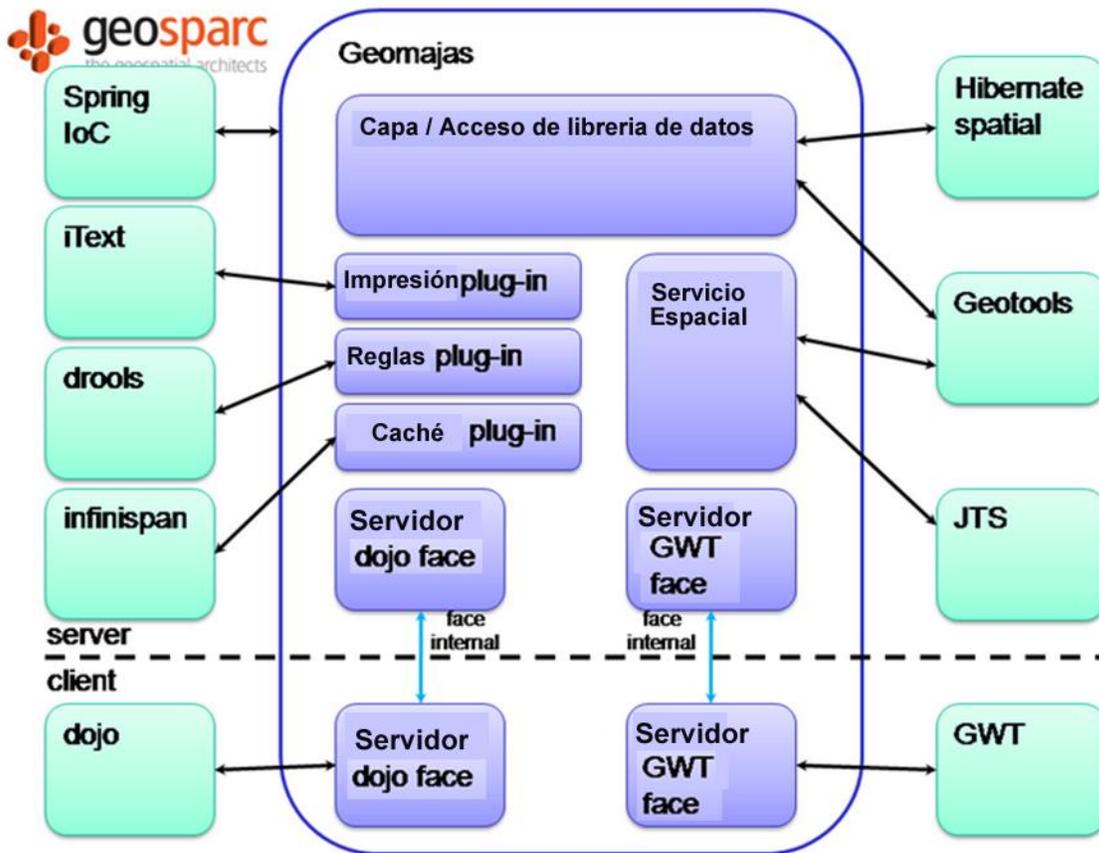


Figura 10 Dependencias de Geomajas

2.2.3.3. Mapfish

MapFish es un *Framework* flexible y completo para crear aplicaciones de mapas en la *web*. Se hace hincapié en una alta productividad y el desarrollo de alta calidad.

MapFish proporciona la tecnología para crear aplicaciones *web* de mapeo. Con MapFish, JavaScript y tecnologías *toolbox*, el desarrollador puede crear interfaces de usuario y con tecnologías de servidor MapFish, es posible el desarrollo de los servicios *web* necesarios.

En el lado cliente, bibliotecas de JavaScript se utilizan principalmente, en el lado del servidor, varios idiomas pueden ser utilizados. Originalmente, se utilizaron los componentes de Python, pero con el tiempo, Java, Ruby y componentes PHP también han sido desarrollados. Esto demuestra la flexibilidad y la apertura del Framework (MapFish, 2011).

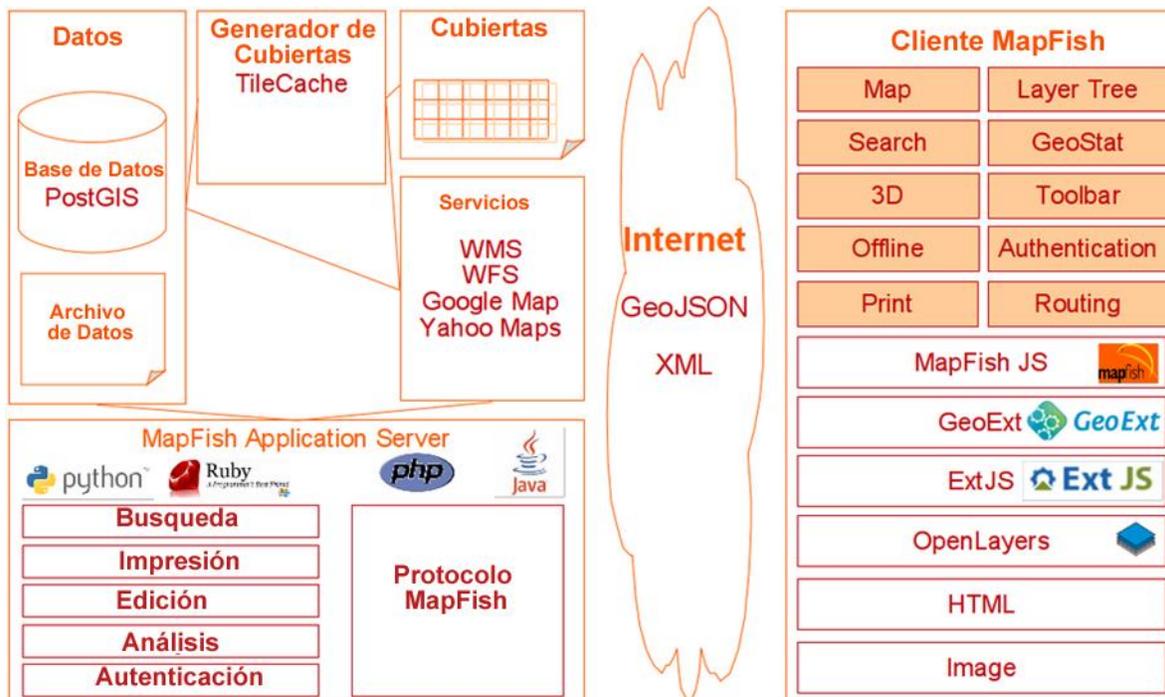


Figura 11 Arquitectura de MapFish

Capítulo 3. Marco Teórico

En este capítulo se estudia el marco teórico del presente trabajo de tesis, que como tópicos fundamentales se presentan la definición de lo que es un SIG y sus principales componentes, que es un Framework, los estándares especificados por la OGC, entre otros.

3.1. Sistema de Información Geográfica (SIG)

De acuerdo a (Tomlinson, R.F., 2005), un Sistema de Información Geográfica (SIG) es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información.

En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. Los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

Hasta hace unas décadas, la manipulación, la síntesis y la representación de la información geográfica se limitaba a los mapas en papel y estas tareas se limitaban a los procesos no interactivos manuales. La mejora exponencial en el rendimiento de las tecnologías computarizadas y la creciente demanda para la manipulación y análisis interactivo de la información geográfica han creado la necesidad de los sistemas de información geográfica (SIG) como lo explican ([Brisaboa, N.R., Cotelo Lema, 2001](#)). Una característica importante de los sistemas de información geográfica es que son más herramientas para producir mapas de papel.

Mientras que en la cartografía tradicional del mapa de papel es la base de datos, en un SIG el mapa es una proyección de una visión particular de una base de datos Geográfica en un momento dado. Esto permite que el usuario final SIG pueda revisar un número ilimitado de alternativas de análisis, y pueda hacer mapas desde diferentes puntos de vista haciendo hincapié en diferentes aspectos de la información. Como consecuencia, los requisitos funcionales para un SIG son muy amplias y pueden ir mucho más allá de las de los sistemas de información tradicionales explican ([Longley, P., Goodchild M., 2001](#)).

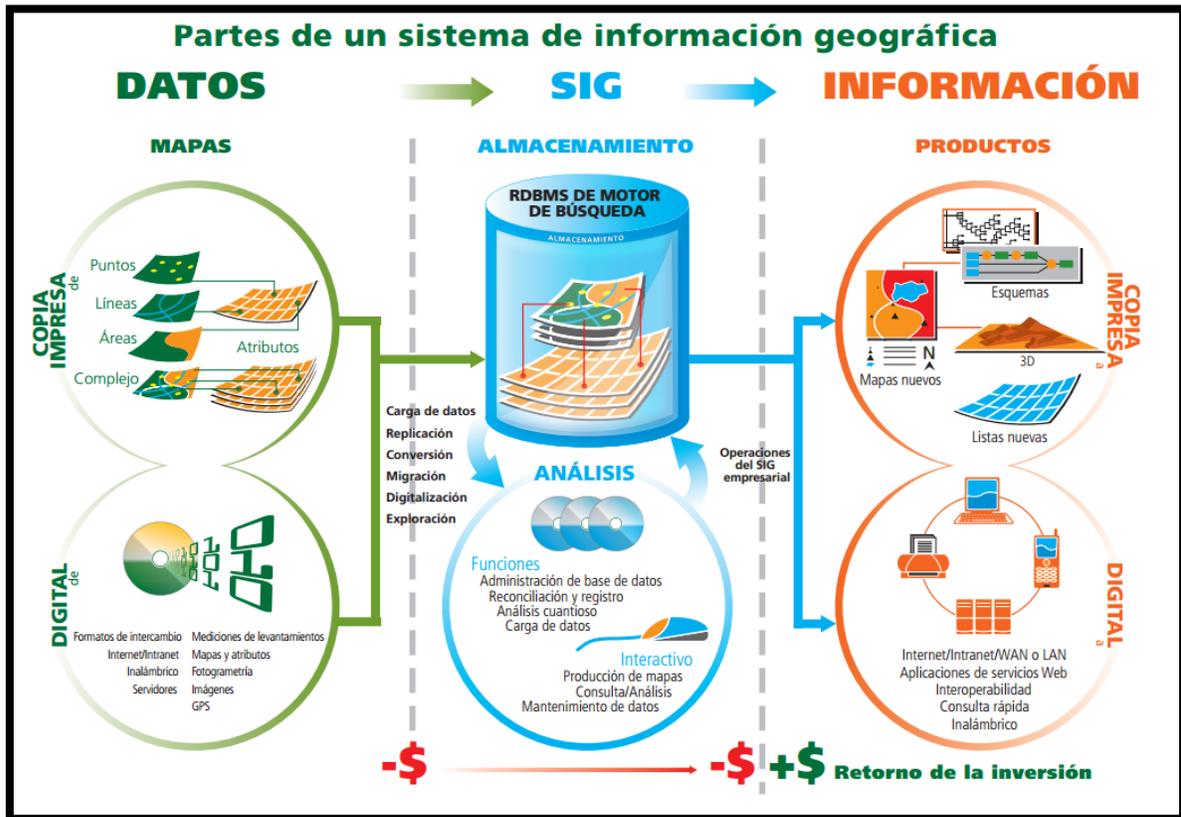


Figura 12 Partes de un Sistema de Información Geográfica

Después de muchos años de investigación y desarrollo, ahora se acepta generalmente que la arquitectura de los sistemas de información de propósito general debe constar de tres capas distintas: la capa de presentación, la capa de lógica de aplicación (o capa de lógica de negocio), y la capa de datos. La principal ventaja de esta arquitectura es que impone una estricta separación de la funcionalidad del sistema en tres módulos independientes diferentes que interactúan sólo en las interfaces bien definidas. Esto permite a un desarrollador al modificar cada uno de estos tenga un impacto mínimo en los demás. Por lo tanto, esta arquitectura ofrece mayor rendimiento, flexibilidad, mantenimiento, reusabilidad y escalabilidad.

A pesar de que la arquitectura de tres niveles para los sistemas de información de propósito general es adecuada para los SIG, la naturaleza especial y las características exclusivas de la información geográfica plantean requisitos funcionales especiales sobre la arquitectura en términos conceptual, estructuras de datos, métodos de acceso, las técnicas de análisis o procedimientos de visualización.

3.1.1. Características especiales de la información geográfica

La naturaleza especial de la información geográfica impone algunos requisitos en la arquitectura del sistema de información. Se enumeran algunos de estos requisitos en esta sección.

3.1.1.1. Representación de la información geográfica

Los modelos conceptuales utilizados para los sistemas de información (por ejemplo, el modelo entidad-relación) no tienen construcciones para modelar esquemas de aplicaciones que tienen que ver con la información geográfica. Además, los modelos lógicos (por ejemplo, el modelo relacional) están fuertemente orientados a las aplicaciones de negocios que manipulan grandes pero simples conjuntos de datos, y no incluyen la funcionalidad para representar la información geográfica.

Además de los tipos de datos para los números, los textos y las fechas, la representación de la información geográfica en una computadora requiere nuevos tipos de datos, curvas, superficies o tipos de colección como colección de puntos, o

colección de geometrías. Por último, los modelos físicos para sistemas de información tradicionales no son capaces de representar la información geográfica de manera eficiente de acuerdo al (ISO/TC 211, 2002).

Los SIG organizan los datos en capas. Cada capa contiene información temática específica. Las capas pueden interrelacionarse mediante procedimientos de análisis más o menos complejos.

Lo que convierte a la tecnología SIG en única es su capacidad para integrar en una misma capa información geográfica (espacial) con bases de datos (datos tabulares, atributos). De este modo cada capa contiene a la vez información geográfica (dónde está tal cosa) y atributos (qué características podemos encontrar en un lugar concreto).

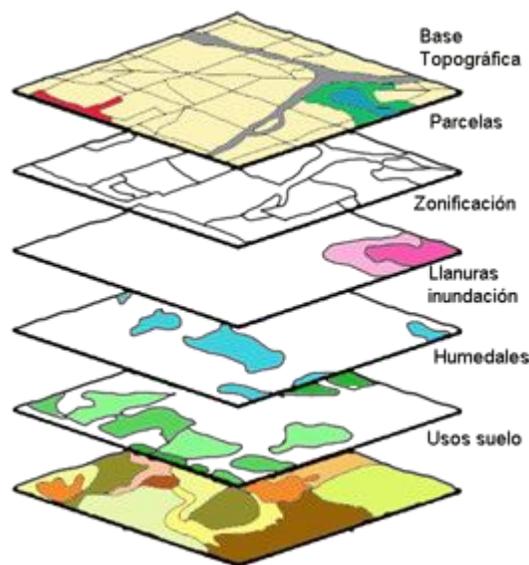


Figura 13 Representación de las Capas Temáticas de un SIG

En resumen, los sistemas de información tradicionales deben ampliarse a todos los niveles del modelo conceptual al modelo físico con el fin de representar la información geográfica adecuadamente.

3.1.1.2. Procesamiento de la Información Geográfica

Hay un rico conjunto de transformaciones especiales, técnicas aplicables a la información geográfica de manipulación y análisis, que debe integrarse en el sistema de información. Esto debe hacerse al proporcionar un exhaustivo conjunto de operaciones primitivas en las abstracciones de datos del modelo conceptual, el cual debe integrarse en un lenguaje de consulta para recuperar y manipular las abstracciones de datos del modelo conceptual. Finalmente, las técnicas de resolución de problemas se deben utilizar para la siguiente categoría de problemas:

- ¿Qué pasa en este lugar, y dónde se encuentra este?
- ¿Cuál es la relación espacial entre estos objetos?
- ¿Con qué sitio satisface estos requisitos?
- ¿Cuál será la situación en el futuro?

Entre otros.

3.1.1.2.1. Funciones de Análisis Espacial

El análisis de datos con SIG tiene por finalidad para descubrir estructuras espaciales, asociaciones y relaciones entre los datos, así como para modelar fenómenos geográficos. Los resultados reflejan la naturaleza y calidad de los datos así como la pertinencia de los métodos y funciones aplicadas.

(Longley, P. A, 2001) Explican que existen diversos criterios para clasificar las funciones de los SIG, especialmente las de análisis espacial. Los grupos siguientes tratan de seguir la secuencia de transformaciones de los datos, desde su formación y transformaciones sucesivas hasta producir resultados, es decir, información.

- **Interrogaciones:** Se pueden utilizar criterios temáticos o espaciales y combinar ambos; las preguntas pueden ser simples y bien definidas (¿Qué hay en tal localización? ¿Cuántos objetos de un tipo se encuentran dentro a menos de 1 km de un punto determinado?) o más vagas (¿cuál es la localidad más cercana a un punto?).
- **Medidas:** Estas funciones producen valores numéricos que describen algunas propiedades esenciales de los objetos, como su longitud, área, forma, pendiente, orientación o la distancia y dirección entre dos o más entidades.
- **Transformaciones:** Comprenden un conjunto de métodos simples de análisis espacial que cambian (transforman) las entidades originales, mediante comparaciones o combinaciones. Estas funciones utilizan principios y reglas geométricas, aritméticas o lógicas, y también operadores de conversión de datos vectoriales a raster y viceversa. Este grupo incluye la creación de corredores (buffer), las operaciones de “punto en polígono”, superposición de polígonos e interpolación espacial.
- **Sumarios:** Son funciones que resumen los datos en unos cuantos indicadores. Los más usados son los de estadística descriptiva (media aritmética, mediana, moda, desviación típica, varianza...) y sus equivalentes espaciales, como el centro de gravedad y la desviación típica de las distancias.

- **Optimización:** Se agrupan aquí diversas técnicas de naturaleza normativa cuya finalidad es la de seleccionar localizaciones que cumplen determinados criterios. Son muy utilizadas en los estudios de mercado y en la planificación de los equipamientos públicos, para estimar la localización óptima de establecimientos con arreglo a determinados supuestos. También se incluyen los métodos de análisis de redes, como la búsqueda de rutas óptimas entre dos o más localizaciones en una red.

3.1.1.3. Visualización de la información geográfica

La visualización de la información geográfica es un rasgo distintivo de las aplicaciones SIG en comparación con los sistemas de información de uso general. La información geográfica tiene algunas peculiaridades que tienen un impacto en el proceso de presentación:

- Las diferentes abstracciones deben ser utilizados para la representación y la visualización de la información.
- Es multi-dimensional.
- Es voluminosa.
- Se requiere a diferentes escalas y tamaños.
- Se requiere de diferentes perspectivas.
- Se proyecta sobre una superficie plana (es decir, en una pantalla de ordenador o un documento).

Por lo tanto, hay una necesidad de operaciones apropiadas para manipular la información geográfica en la interfaz de usuario del sistema. Estas operaciones

deben incorporar procesos dinámicos tales como zoom, la adición y eliminación de la información, etc.

3.1.1.4. Arquitectura del Sistema y otras cuestiones

La naturaleza especial de información geográfica hace más importante el cumplimiento de algunos requisitos de los sistemas de información de carácter general, tales como flexibilidad, extensibilidad, reusabilidad, escalabilidad, confiabilidad y seguridad. Con el fin de proporcionar estas características, la arquitectura de los SIG debe basarse en una SGBD¹² que extienda servicios de gestión de la información geográfica, y una colección de modular altamente distribuida, servicios de procesamiento y visualización de información geográfica.

La información geográfica plantea otras exigencias importantes en la arquitectura del sistema. En primer lugar, como cualquier otro dominio de aplicación para el sistema de información, el campo de los SIG necesita elementos especiales de metadatos¹³ para describir las particularidades del dominio de la aplicación.

Los metadatos en la arquitectura de un SIG permiten encontrar conjuntos de datos de información geográfica, para describir cómo se puede utilizar la información, y para comprobar si la información cumple con algunos requisitos.

Además de esto, la cantidad de posibles técnicas de análisis para información geográfica es ilimitado. Por lo tanto, ninguna herramienta de desarrollo de SIG

¹² Un sistema Gestor de bases de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos

¹³ Metadatos (del griego μετα, meta, 'después de, más allá de' y latín datum, 'lo que se da', «dato»), literalmente «sobre datos», son datos que describen otros datos.

puede proporcionar toda posible operaciones y técnicas de resolución de problemas. Como consecuencia, **“la arquitectura de un Herramienta de desarrollo de SIG debe ser extensible para apoyar la incorporación de nuevos análisis de datos”** (Longley, P., Goodchild, M., 1999). El proceso de ampliación de la funcionalidad de una herramienta de desarrollo de SIG es generalmente referido como personalización.

3.1.2. Bases de Datos Espaciales

Como se mencionó antes para poder representar la información geográfica se requiere de tipos de datos especiales, a los que se les conoce como *“datos espaciales”* y las bases de datos que almacenan esta información se les conoce como bases de datos espaciales.

En este tipo de bases de datos es imprescindible establecer un cuadro de referencia (un SRE, Sistema de Referencia Espacial) para definir la localización y relación entre objetos, ya que los datos tratados en este tipo de bases de datos tienen un valor relativo, no es un valor absoluto. Los sistemas de referencia espacial pueden ser de dos tipos: georreferenciados (aquellos que se establecen sobre la superficie terrestre. Son los que normalmente se utilizan, ya que es un dominio manipulable, perceptible y que sirve de referencia) y no georreferenciados (son sistemas que tienen valor físico, pero que pueden ser útiles en determinadas situaciones).

La construcción de una base de datos geográfica implica un proceso de abstracción para pasar de la complejidad del mundo real a una representación simplificada que pueda ser procesada por el lenguaje de las computadoras actuales. Este proceso de abstracción tiene diversos niveles y normalmente comienza con la

concepción de la estructura de la base de datos, generalmente en capas; en esta fase, y dependiendo de la utilidad que se vaya a dar a la información a compilar, se seleccionan las capas temáticas a incluir.

La estructuración de la información espacial procedente del mundo real en capas conlleva cierto nivel de dificultad. En primer lugar, la necesidad de abstracción que requieren los computadores implica trabajar con primitivas básicas de dibujo, de tal forma que toda la complejidad de la realidad ha de ser reducida a puntos, líneas o polígonos.

En segundo lugar, existen relaciones espaciales entre los objetos geográficos que el sistema no puede obviar; la topología, que en realidad es el método matemático-lógico usado para definir las relaciones espaciales entre los objetos geográficos puede llegar a ser muy compleja, ya que son muchos los elementos que interaccionan sobre cada aspecto de la realidad (Rigaux, P. Scholl, 2002).

3.1.2.1. Datos espaciales

Un modelo de datos geográfico es una abstracción del mundo real que emplea un conjunto de objetos dato, para soportar el despliegue de mapas, consultas, edición y análisis. Los datos geográficos, presentan la información en representaciones subjetivas a través de mapas y símbolos, que representan la geografía como formas geométricas, redes, superficies, ubicaciones e imágenes, a los cuales se les asignan sus respectivos atributos que los definen y describen.

Un dato espacial es una variable asociada a una localización del espacio. Normalmente se utilizan datos vectoriales, los cuales pueden ser expresados mediante tres tipos de objetos espaciales (Kraak M. J., Ormeling F., 2011).

3.1.2.1.1. Puntos

Se encuentran determinados por las coordenadas terrestres medidas por latitud y longitud. Por ejemplo, ciudades, accidentes geográficos puntuales, hitos.

3.1.2.1.2. Líneas

Objetos abiertos que cubren una distancia dada y comunican varios puntos o nodos, aunque debido a la forma esférica de la tierra también se le consideran como arcos. Líneas telefónicas, carreteras y vías de trenes son ejemplos de líneas geográficas.

3.1.2.1.3. Polígonos

Figuras planas conectadas por distintas líneas u objetos cerrados que cubren un área determinada, como por ejemplo países, regiones o lagos.

De esta forma la información sobre puntos, líneas y polígonos se almacena como una colección de coordenadas (x, y). La ubicación de una característica puntual, pueden describirse con un sólo punto (x, y). Las características lineales, pueden almacenarse como un conjunto de puntos de coordenadas (x, y). Las características poligonales, pueden almacenarse como un circuito cerrado de coordenadas. La otra forma de expresar datos espaciales es mediante rasterización, la cual, a través de una malla que permite asociar datos a una imagen; es decir, se pueden relacionar paquetes de información a los píxeles de una imagen digitalizada.

3.2. Estándares internacionales para los SIG

Teniendo en cuenta que cada aplicación tiene un punto de vista sobre la información geográfica diferente, cada desarrollador ha definido modelos conceptuales, modelos de datos geográficos, los formatos de almacenamiento, operaciones de análisis, o procedimientos de representación especialmente adaptados a los requisitos de la aplicación. Como consecuencia, existe hoy en día un problema de la interoperabilidad entre las herramientas de desarrollo de los SIG; no es fácil de usar una herramienta para analizar la información recopilada con otra herramienta.

3.2.1. ISO y OGC

Con el fin de resolver este problema, un número de gobiernos, la investigación y la industria fundaron el Consorcio OpenGIS (OGC) en 1994 agrupa (en febrero de 2009) a 372 organizaciones públicas y privadas. Las raíces del OGC se encuentran en la software fuente libre GRASS y la subsiguiente fundación OGF (Open GIS Foundation) fundada en 1992. Su fin es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica y de la World Wide Web. Persigue acuerdos entre las diferentes empresas del sector que posibiliten la interoperación de sus sistemas de geoprocésamiento y facilitar el intercambio de la información geográfica en beneficio de los usuarios. Anteriormente fue conocido como Open GIS Consortium. Antes de firmar como Consortium firmó como fundación (OGC, 2003).

Algunas de sus más importantes especificaciones son las siguientes:

- GML - Lenguaje de Marcado Geográfico
- KML - Keyhole Markup Language

- WFS - Web Feature Service
- WMS - Web Map Service
- WCS - Web Coverage Service
- CSW - Web Catalogue Service

Otra organización de estándares que ha dedicado muchos esfuerzos a las aplicaciones de SIG es la Organización Internacional de Normalización (ISO) por medio del Comité Técnico ISO 211 ([ISO/TC 211, 2002](#)), llamado Información Geográfica/Geomática. El objetivo de ambas organizaciones es crear especificaciones de las normas relativas a la información geográfica con el detalle suficiente para permitir a los desarrolladores crear implementaciones conformes a tales normas que interoperen sin problemas.

Durante muchos años, la OGC y la ISO estaban trabajando de forma independiente para llegar a objetivos superpuestos, pero hoy en día ambos cuerpos tratan de converger hacia una solución común.

El OGC y la ISO/TC 211 están llevando a cabo una tarea muy importante en el campo de los sistemas de información geográfica. Se están sentando las bases para una nueva generación de aplicaciones para SIG y herramientas de desarrollo que serán capaces de cooperar en mayor medida en muchos niveles diferentes.

Además, en lugar de proponer una capa de software monolítico que implementen todas las funcionalidades de las aplicaciones SIG, estas organizaciones están proponiendo romper la funcionalidad de una vasta colección de servicios con una funcionalidad muy específica que interactúen sólo en las interfaces. Esto permite

una arquitectura de aplicaciones SIG muy flexible porque los servicios que implementen la funcionalidad pueden estar ejecutándose en un único equipo, o distribuido a lo largo de una red de área amplia, de una manera totalmente transparente.

3.2.1.1. Especificaciones OGC

En el siguiente apartado se describen algunas de las más importantes especificaciones publicadas por la OGC.

3.2.1.1.1. WMS

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

El estándar define tres operaciones:

1. Devolver metadatos del nivel de servicio.
2. Devolver un mapa cuyos parámetros geográficos y dimensionales han sido bien definidos.
3. Devolver información de características particulares mostradas en el mapa (opcionales).

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

El servicio WMS permite así la creación de una red de servidores distribuidos de mapas, a partir de los cuales los clientes pueden construir mapas a medida. Las operaciones WMS también pueden ser invocadas usando clientes avanzados SIG, realizando igualmente peticiones en la forma de URLs. Existe software libre, como las aplicaciones GRASS, uDIG, gvSIG, Kosmo y otros, que permite este acceso avanzado a la información remota, añadiendo la ventaja de poder cruzarla con información local y disponer de una gran variedad de herramientas SIG como se explica en ([OGC 06-042, 2006](#)).

3.2.1.1.2. WFS

Web Feature Service o **WFS** del Consorcio Open Geospatial Consortium o OGC es un servicio estándar, que ofrece una interfaz de comunicación que permite interactuar con los mapas servidos por el estándar WMS, como por ejemplo, editar la imagen que nos ofrece el servicio WMS o analizar la imagen siguiendo criterios geográficos.

Para realizar estas operaciones se utiliza el lenguaje GML que deriva del XML, que es el estándar a través del que se transmiten las órdenes **WFS**.

WFS no transaccional permite hacer consultas y recuperación de elementos geográficos. Por el contrario WFS-T (Web Feature Service Transactional) permite además la creación, eliminación y actualización de estos elementos geográficos del mapa ([OGC WFS, 2005](#)).

3.2.1.1.3. WCS

La interfaz estándar Web Coverage Service (WCS) o Servicio de Cobertura *web* de Open Geospatial Consortium proporciona una interfaz que permite realizar peticiones de cobertura geográfica a través de la *web* utilizando llamadas independientes de la plataforma. Las coberturas son objetos (o imágenes) en un área geográfica mientras que la interfaz WMS o portales de mapas online como Google Maps devuelven sólo una imagen, que los usuarios no pueden editar o analizar espacialmente.

El grupo OGC define y mantiene la especificación WCS. GeoServer sirve como implementación de referencia del estándar.

El servicio de cobertura *web* básico permite realizar consultas y obtener coberturas. WCS describe las operaciones de descubrimiento, petición o transformación de datos. El cliente genera una petición y la envía a l servidor *web* utilizando el protocolo HTTP. El servidor *web* ejecuta la petición. La especificación WCS utiliza

HTTP como plataforma de computación distribuida, aunque no es un requisito obligatorio.

Hay dos tipos de codificación definidas para las operaciones de WCS:

- XML (apto para HTTP POST/SOAP)
- Pares Clave-Valor (apto para HTTP GET/Remote procedure call)

En la clasificación de servicios *web*, WCS está categorizada como un servicio de tipo RPC no-tranquilo.

Los datos pueden estar disponibles en varios formatos, como DTED, GeoTIFF, HDF-EOS, o NITFS.

Se soportan varios tipos de capas de datos:

- series de puntos, como ejemplos de localizaciones
- red regular de píxeles o puntos, que representan una foto
- conjunto de curvas segmentadas, a menudo utilizadas para rutas de carretera
- conjunto de polígonos de Thiessen utilizados para analizar los datos distribuidos espacialmente como medidas de precipitación
- red irregular triangulada (TIN), a menudo utilizada para modelos de terreno

Se pueden añadir rangos de información a las localizaciones, como la velocidad media del viento o rendimientos del cultivo ([OGC 09-110, 2012](#)).

3.2.2. Arquitectura genérica para los SIG

En la *Figura 14* se representa la arquitectura de una aplicación de SIG con dos diferentes interfaces de usuario construidas sobre las especificaciones ya descritos anteriormente. La primera interfaz consiste en un SIG de escritorio basado en el análisis de datos geográficos, y el segundo es un SIG basado en la *web* para la representación geográfica de la información. La *Figura 14* muestra las especificaciones ya adoptada en cajas blancas, y las piezas que faltan de la arquitectura en cajas grises.

El trabajo desarrollado por estas organizaciones tiene también algunos pequeños inconvenientes, por ejemplo, a pesar de que las especificaciones proporcionan un nivel de gestión de la información completa, gran parte de los algoritmos de procesamiento y técnicas de representación gráfica de información deben ser definidas por un desarrollador. No hay especificaciones que se hayan definido para el nivel de procesamiento, y muy poco trabajo ha sido producido para el nivel de interfaz humana.

Otro pequeño problema es que la naturaleza intrínseca de cualquier organización de estándares hace que el proceso de desarrollar una especificación tiende a ser bastante lento. Antes de una especificación que se adopte, debe ser propuesto, escrito en un proyecto de estado, discutida y votada por los miembros. Se trata de un largo proceso que no puede ser asumido por las empresas de software que debe producir nuevos productos a un ritmo mucho más rápido.

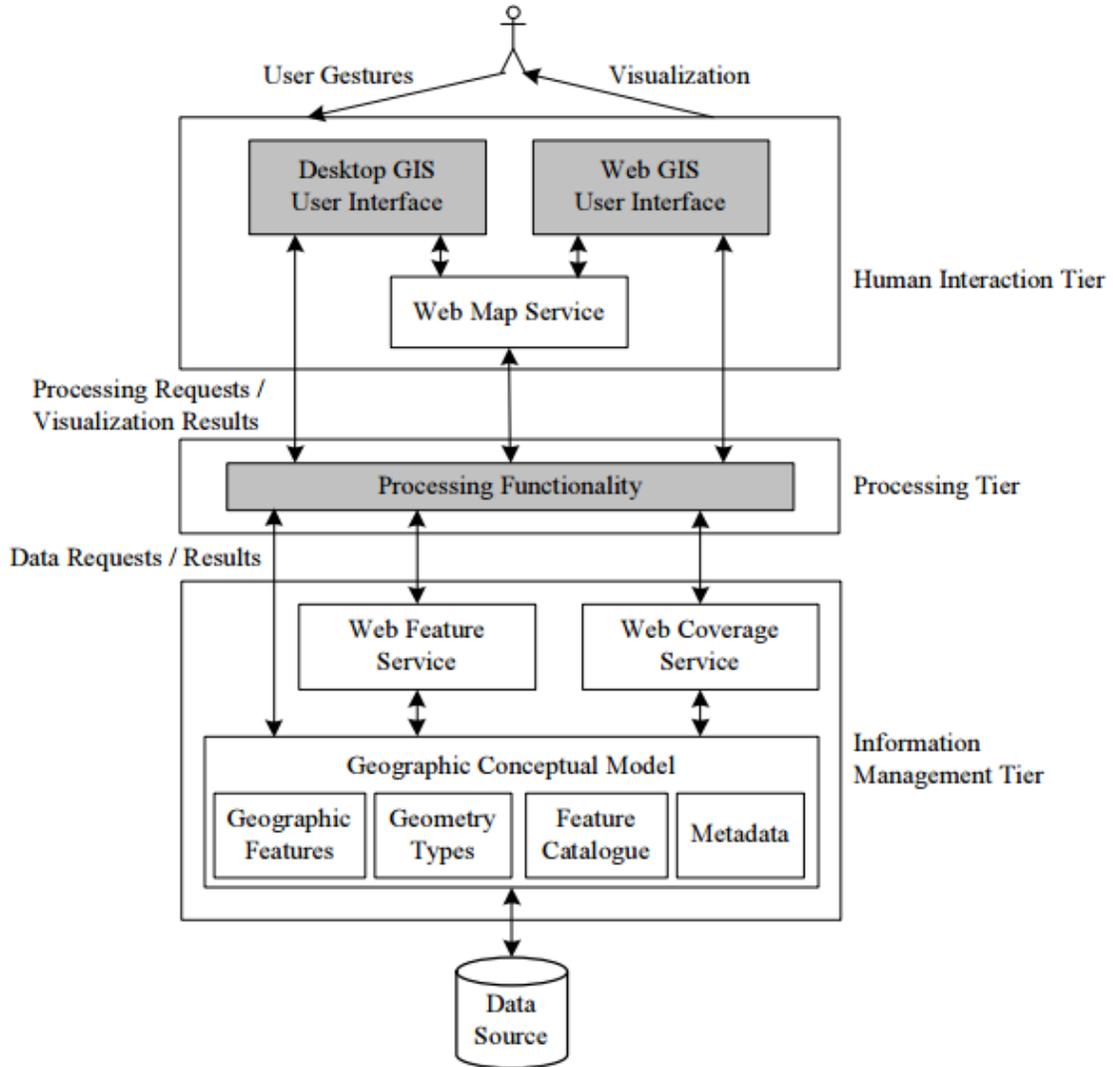


Figura 14 Servicios OGC en la Arquitectura SIG

3.3. Framework

La palabra inglesa *framework* (marco de trabajo) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un *framework* o infraestructura digital, es una

estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

De acuerdo a [\(Riehle, Dirk, 2000\)](#), los frameworks contienen características distintivas que los separan de las bibliotecas normales:

- La inversión de control: En un *framework*, a diferencia de las bibliotecas o aplicaciones de usuario normal, el flujo del programa general de control no está dictada por la persona que llama, sino por el *framework*.
- Comportamiento predeterminado: Un *framework* tiene un comportamiento predeterminado. Este comportamiento por defecto debe ser un comportamiento útil.
- Extensibilidad: Un *framework* puede ser ampliado por el usuario por lo general por sobre escritura selectiva o especializada del código de usuario para proporcionar una funcionalidad específica.
- Código de *framework* no modificable: El código del *framework*, en general, no se supone que debe ser modificado, si bien acepta extensiones implementadas por el usuario. En otras palabras, los usuarios pueden ampliar el *framework*, pero no deben modificar su código.

Los diseñadores de *frameworks* de software tienen por objeto facilitar el desarrollo de software al permitir que los diseñadores y programadores puedan dedicar su tiempo a cumplir con los requisitos de software en lugar de tratar con los detalles de bajo nivel de proveer un sistema de trabajo, reduciendo así el tiempo de desarrollo general (DocForge, 2008). Por ejemplo, un equipo utilizando un *framework* de aplicaciones *web* para desarrollar un sitio *web* bancario puede centrarse en la escritura de código en particular a la banca en lugar de la mecánica de la administración de solicitudes y la gestión de estados.

3.3.1. Problemas de los *frameworks*

Frameworks a menudo aumentan el tamaño de los programas, un fenómeno denominado "*code bloat*". Además, debido a la complejidad de sus API's¹⁴, la reducción prevista en el tiempo total de desarrollo a veces no puede lograrse, debido a la necesidad de dedicar más tiempo a aprender a usar el *framework*, esta crítica es claramente válida cuando un *framework* especial o nuevo se encuentra por primera vez por el equipo de desarrollo. Si este *framework* no se utiliza en trabajos posteriores, el tiempo invertido en el aprendizaje del mismo puede costar más que el desarrollo del software final requerido.

Sin embargo, una vez que se aprende un *framework*, los proyectos futuros pueden ser más rápidos y más fáciles de completar; el concepto de un *framework* es hacer soluciones a la medida para todos, y con la familiarización, la producción de código lógicamente incrementar.

¹⁴ Interfaz de programación de aplicaciones o API es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Capítulo 4. Metodología

En este capítulo se describe la metodología propuesta para llevar a cabo la implementación del *framework* para desarrollo de aplicaciones Web SIG. Retomando los conceptos descritos en capítulos anteriores, partiremos de que las aplicaciones Web SIG que generará el *framework* serán de una arquitectura de 3 capas (presentación, lógica de negocios y datos). En esta metodología se describe a su vez como se integrarán las distintas herramientas, ya existentes, y además que son OpenSource tales como el servidor de mapas GeoServer, la biblioteca de despliegue de mapas OpenLayers, la base de datos Postgres con su extensión para el manejo de datos espaciales PostGIS, entre otras.

4.1. Enfoque general de solución



Figura 15 Arquitectura Entrada-Salida del Framework

La herramienta propuesta debe de lograr crear Aplicaciones Web SIG completas, y además que el desarrollador no tenga que escribir demasiado código de

programación (*Figura 15*), para lograr esto se propone una interfaz Front End *web* con la cual el desarrollador de la aplicación únicamente tendrá que interactuar con la misma sin tener que escribir líneas de código.

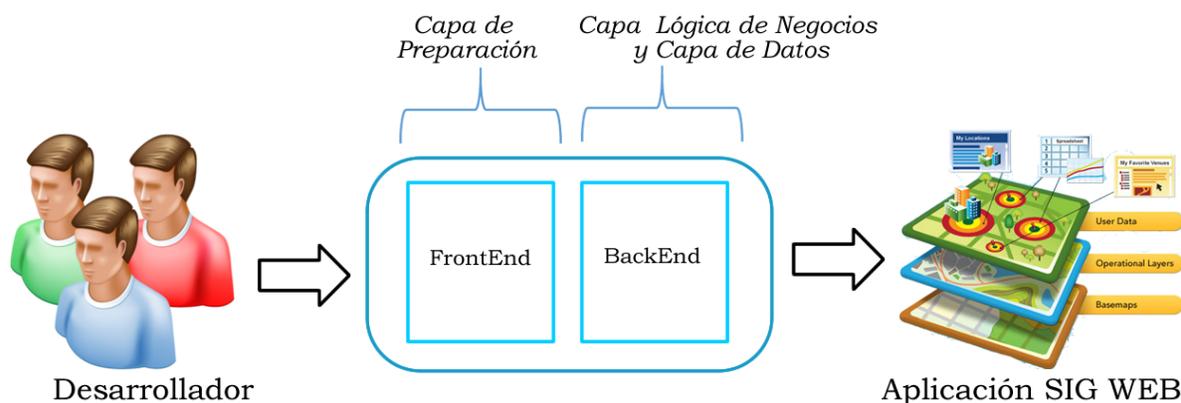


Figura 16 Representación de Back End y Front End

Por otro lado el back end¹⁵ a desarrollar se encargará, en un módulo del lado del cliente, de la generación del código con base en plantillas de código JavaScript, que permitirán agregar componentes a cada aplicación a desarrollar, dichos componentes serán los encargados de la creación de la aplicación SIG. Los componentes mencionados son: el visor de mapas, la barra de herramientas para dibujo de objetos espaciales, el selector de capas y la barra de herramientas de operaciones de análisis espacial. Estos componentes serán desarrollados a partir de la biblioteca OpenLayers de JavaScript, aunado a ésta, se usarán dos extensiones más, GeoExt y ExtJS, extensiones de JavaScript que brindan apariencias más amigables y componentes más interactivos con el usuario.

Además de lo anterior, el *back end* se compone también de un módulo del lado del

¹⁵ back-end es la parte que procesa la entrada desde el front-end

servidor, con base en un servidor *web* Apache Tomcat que trabaja con Java Server Pages junto con la biblioteca Java Topology Suit para lograr las funciones de análisis espacial.

La gestión de las fuentes de los datos espaciales será llevada a cabo mediante el servidor de mapas GeoServer, el cual administra los datos espaciales provenientes de distintas fuentes, como pueden ser BDE, archivos Shape, servicios WMS. La base de datos a utilizar será PostgreSQL con su extensión para datos espaciales PostGIS.

4.2. Arquitectura propuesta

Como en capítulos anteriores se han mencionado los requisitos especiales en la arquitectura de los SIG, en esta sección se presenta una descripción de los componentes, las interacciones y las dependencias en una arquitectura genérica propuesta para los SIG que cumpla con los requisitos mencionados. El diseño de esta arquitectura está fuertemente influenciado por las propuestas de la ISO / TC 211 y el OGC ya que el trabajo de estas organizaciones supone especificaciones de años de trabajo. La *Figura 17* muestra la propuesta de una arquitectura genérica para los SIG sobre la *web* que el *framework* generará.

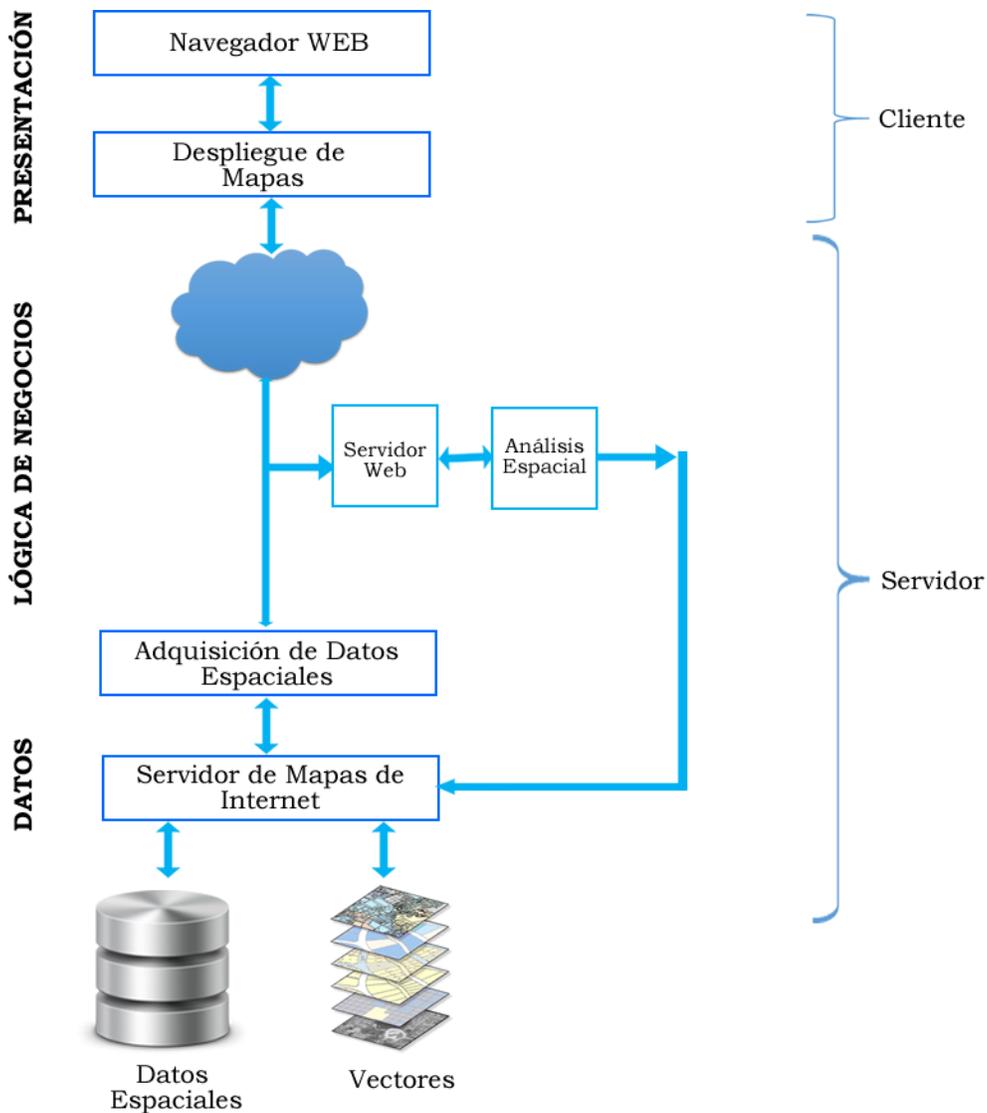


Figura 17 Arquitectura del Framework Web SIG

La arquitectura separa la funcionalidad del sistema en tres capas independientes, las cuales son: la capa de datos, la capa de lógica de aplicación o negocio y la capa de presentación. La capa de datos proporciona funcionalidad de gestión de datos de forma independiente de la tecnología de software. La capa de presentación se encarga de ejecutar la interfaz de usuario del sistema, la visualización de los mapas y proporcionar cierta funcionalidad sobre ellos así, como algunas de las operaciones de análisis espacial. La capa de lógica de negocios implementa la

funcionalidad de resolución de las operaciones entre la capa de presentación y la capa de datos.

A fin de permitir la reutilización y la flexibilidad de la arquitectura del sistema, las funcionalidades de estas capas deben ser implementadas de manera independiente de cualquier aplicación particular. La estrategia utilizada consiste en encontrar y aislar las características que son independientes.

La arquitectura construida con estos módulos será la utilizada para las aplicaciones SIG creadas con el Framework únicamente llenando en los detalles específicos de cada aplicación a crear. Describimos ahora cada capa en más detalle.

4.2.1. Capa de Datos

El propósito de la capa de datos es proporcionar funcionalidad de gestión de la información geográfica, independientemente de la tecnología de software utilizado para almacenarla. Este nivel recibe las solicitudes de recuperación de información y manipulación utilizando un lenguaje de consulta, evalúa la consulta y devuelve un conjunto de objetos de datos que se representan mediante un lenguaje de intercambio de información.

Teniendo en cuenta que puede haber muchos tipos diferentes de fuentes de datos, dentro de la arquitectura debe existir un mediador para administrar las diversas fuentes de datos espaciales y entregar o publicar los datos hacia el cliente. Dicho mediador que administra la adquisición de las fuentes de datos es llevado a cabo por GeoServer.

4.2.1.1. GeoServer

GeoServer como se ha descrito anteriormente es un servidor de mapas en Internet de código abierto, escrito en Java, que permite compartir la información geoespacial proveniente de la mayoría de los orígenes de datos espaciales y además entregar la información en distintos formatos de despliegue, ya sea en formato raster o en formato vectorial, a través del uso de los estándares como WMS, WFS, GML y KML ya anteriormente explicados.

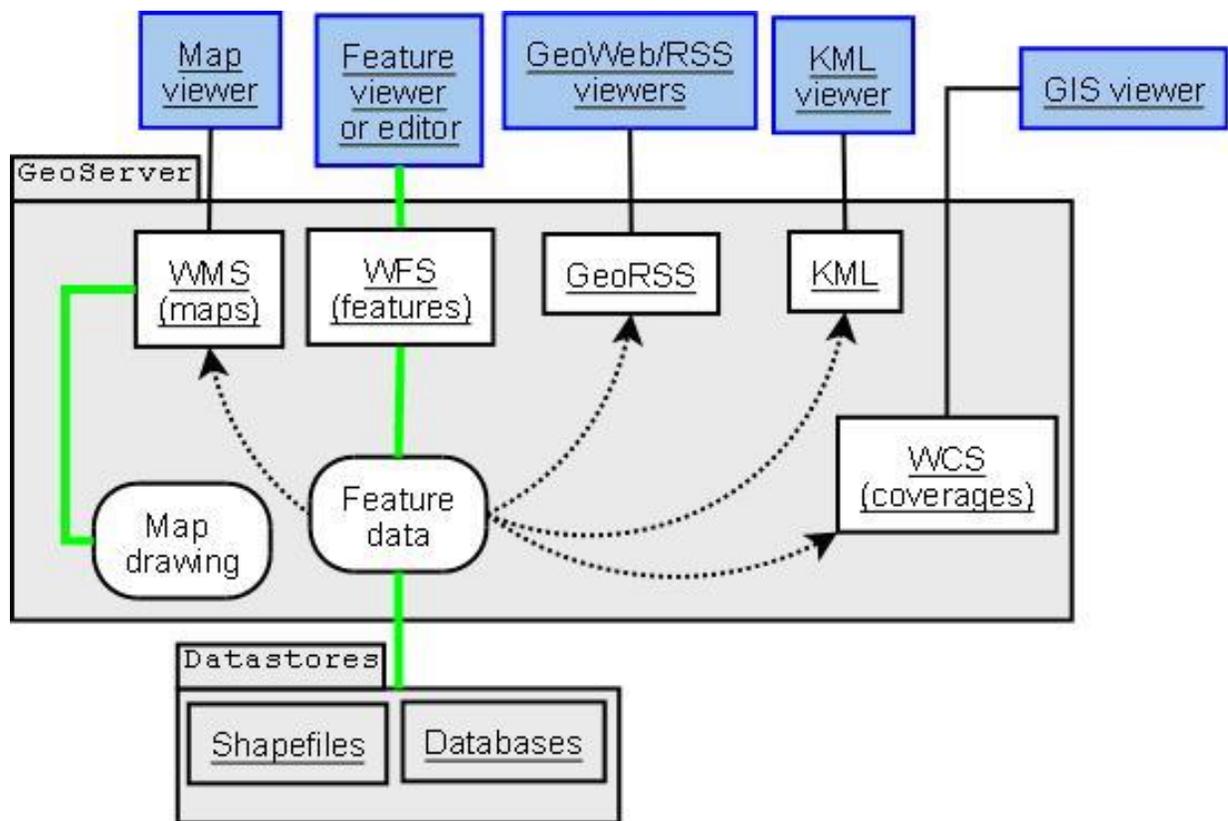


Figura 18 Arquitectura del funcionamiento de Geoserver

Dentro de las fuentes de datos que GeoServer soporta tenemos las siguientes:

- PostGIS
- Shapefile
- ArcSDE y Oracle
- MySQL
- MapInfo
- WFS

Como se puede observar se cuenta con una variedad de fuentes de entrada para GeoServer, haciendo a su vez que nuestro Framework propuesto sea flexible en este aspecto. En este trabajo únicamente se emplean 2 fuentes de datos espaciales: PostGIS de Postgres y ShapeFiles.

4.2.1.2. PostgreSQL/PostGIS

Como una de las fuentes de datos para nuestra arquitectura del Framework se utilizan las BDE administradas por PostgreSQL con su extensión PostGIS.

PostgreSQL, a menudo simplemente "Postgres", es un sistema de gestión de bases de datos objeto-relacional (ORDBMS), y además libre bajo licencia BSD¹⁶, con énfasis en la extensibilidad y normas de cumplimiento. Como servidor de base de datos, su función principal es la de almacenar los datos, de forma segura y el apoyo a las mejores prácticas, y recuperarlo más tarde, a petición de otras aplicaciones de software, ya sea aquellos en el mismo equipo o en los que se ejecuta en otro equipo a través de una red (incluido el de Internet). Se puede manejar las cargas de trabajo que van desde pequeñas aplicaciones de un solo

¹⁶ La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution), La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

máquinas para grandes aplicaciones orientadas a Internet con muchos usuarios concurrentes. Las versiones recientes también ofrecen la réplica de la base de datos en sí para la seguridad y la escalabilidad.

PostgreSQL implementa la mayoría del estándares SQL, maneja las consultas SQL complejas utilizando muchos métodos de indexación que no están disponibles en otras bases de datos; tiene vistas actualizables, *triggers*, claves foráneas; apoya las funciones y los procedimientos almacenados y tiene un gran número de extensiones escritos por terceras partes. Además de la posibilidad de trabajar con las principales bases de datos de fuentes propietarias y abiertas, PostgreSQL soporta la migración de los mismos, por su amplio soporte de estándar SQL y herramientas de migración disponibles ([PostgreSQL, 2014](#)).

Para solventar el manejo de datos espaciales, Postgres utiliza una extensión de la misma llamada PostGIS. PostGIS es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en Sistema de Información Geográfica. Se publica bajo la Licencia Pública General¹⁷ de GNU.

PostGIS ha sido desarrollado por la empresa canadiense Refraction Research, especializada en productos "Open Source" entre los que habría que citar a Udig. PostGIS es hoy en día un producto veterano que ha demostrado versión a versión su eficiencia. En relación con otros productos, PostGIS ha demostrado ser muy superior a la extensión geográfica de la nueva versión de MySQL, y a juicio de

¹⁷ La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License (o simplemente sus siglas del inglés GNU GPL) es la licencia más ampliamente usada¹ en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.

muchos, es muy similar a la versión geográfica de la base de datos Oracle ([PostGIS, 2014](#))

4.2.1.3. Shapefile

Como segunda fuente de datos utilizada dentro de la arquitectura propuesta para nuestro Framework se utiliza el shapefile. El formato shapefile (SHP) es un formato de archivo informático propietario de datos espaciales desarrollado por la compañía ESRI, quien crea y comercializa software para SIG como Arc/Info o ArcGIS. Originalmente se creó para la utilización con su producto ArcView GIS, pero actualmente se ha convertido en un estándar para el intercambio de información geográfica entre SIG por la importancia que los productos ESRI tienen en el mercado SIG y por estar muy bien documentado.

Un shapefile es un tipo de archivo que contiene datos espaciales en un formato vectorial de almacenamiento digital, donde se guarda la localización de los elementos geográficos y los atributos asociados a ellos. No obstante carece de capacidad para almacenar información topológica. Es un formato multiarchivo, es decir está generado por varios archivos ([ESRI E., 1998](#)). El número mínimo requerido es de tres y tienen las extensiones siguientes:

- .shp - es el archivo que almacena las entidades geométricas de los objetos.
- .shx - es el archivo que almacena el índice de las entidades geométricas.
- .dbf - es la base de datos, en formato dBASE, donde se almacena la información de los atributos de los objetos.

Además de estos tres archivos requeridos, opcionalmente se pueden utilizar otros para mejorar el funcionamiento en las operaciones de consulta a la base de datos,

información sobre la proyección cartográfica, o almacenamiento de metadatos. Estos archivos son:

- .prj - Es el archivo que guarda la información referida al sistema de coordenadas en formato WKT.
- .sbn y .sbx - Almacena el índice espacial de las entidades.
- .fbn y .fbx - Almacena el índice espacial de las entidades para los shapefiles que son inalterables (solo lectura).
- .ain y .aih - Almacena el índice de atributo de los campos activos en una tabla o el tema de la tabla de atributos.
- .shp.xml - Almacena los metadatos del shapefile.

4.2.2. Capa Lógica de Negocio

La capa de lógica de negocio comprende las operaciones de negocio entre la capa de datos y la capa de presentación del sistema. Este nivel está compuesto por un servidor *web*, independiente del servidor de mapas, encargado de atender las peticiones correspondientes para la realización de las operaciones de análisis espacial: buffer, unión, intersección, diferencia. Tales peticiones son procesadas por el módulo de análisis espacial, encargado de realizar las operaciones mediante la API de java JTS que más adelante se explicará a detalle.

Con la separación de este servicio dentro de la arquitectura podemos permitirle a los desarrolladores y usuarios finales construir nuevos servicios o módulos.

4.2.2.1. Servidor Web Apache Tomcat

Un servidor *web* o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.

Apache Tomcat (o simplemente Tomcat, anteriormente Jakarta Tomcat) es un servidor *web* de código abierto y contenedor de servlets desarrollado por la Apache Software Foundation (ASF). Tomcat implementa las especificaciones Servlet Java y Java Server Pages (JSP), especificaciones de Sun Microsystems, y proporciona un entorno de servidor *web* HTTP "Java puro" para el lenguaje Java. El proceso se ejecuta una máquina virtual Java (JVM). Cada petición HTTP de un navegador para Tomcat se procesa en el proceso de Tomcat en un hilo separado ([Apache Tomcat, 2014](#)).

4.2.2.2. Módulo de Análisis Espacial

Para este módulo se emplea un servidor *web* montado sobre Apache Tomcat, el cual, siguiendo la filosofía de este trabajo, también es de código abierto, donde sobre dicho servidor *web* se montan las Java Server Pages o JSP's, código en Java del lado del servidor, el cual junto con la API Java Topology Suite (JTS) se encargarán de realizar las operaciones de análisis espacial soportadas por el Framework mediante el modulo creado totalmente en Java.

4.2.2.1.1. Java Server Pages (JSP)

Java Server Pages (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas *web* dinámicas basadas en HTML, XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.

La principal ventaja de JSP frente a otros lenguajes es que el lenguaje Java es un lenguaje de propósito general que excede el mundo *web* y que es apto para crear clases que manejen lógica de negocio y acceso a datos de una manera prolija. Esto permite separar en niveles las aplicaciones *web*, dejando la parte encargada de generar el documento HTML en el archivo JSP.

Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios. Es común incluso que los desarrolladores trabajen en una plataforma y que la aplicación termine siendo ejecutada en otra.

Los servlets y las JSP son dos métodos de creación de páginas *web* dinámicas en servidor usando el lenguaje Java. En ese sentido, son similares a otros métodos o lenguajes tales como el PHP, ASP o CGIs, programas que generan páginas *web* en el servidor. Sin embargo, tienen algunas diferencias. Para comenzar, las JSP y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de computadora, siempre que exista una máquina virtual Java para él. Cada servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en su propio hilo, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar

programa +intérprete). Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo (JSP, 2014).

Las JSP son en realidad una forma alternativa de crear servlets ya que el código JSP se traduce a código de servlet Java la primera vez que se le invoca y en adelante es el código del nuevo servlet el que se ejecuta produciendo como salida el código HTML que compone la página *web* de respuesta. En la *Figura 19* se muestra el funcionamiento de los JSP's.

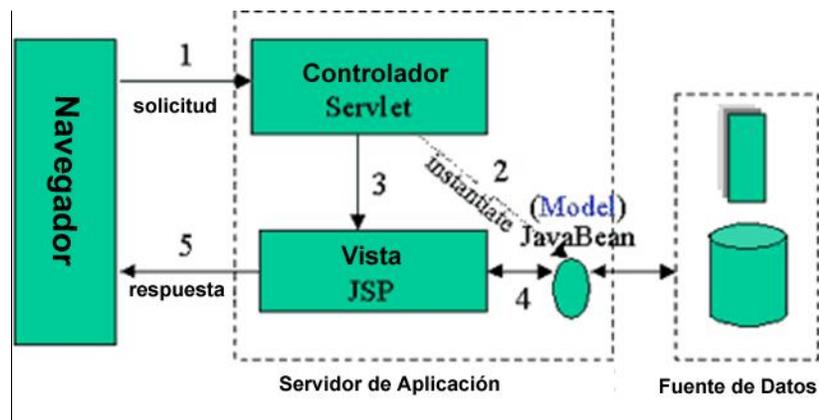


Figura 19 Funcionamiento de JSP

4.2.2.1.2. Java Topology Suite (JTS)

Java Topology Suite (JTS) es una API que proporciona un modelo de objetos espaciales y funciones geométricas 2D fundamentales. JTS ha sido desarrollada por la empresa Vivid Solutions y está implementada íntegramente en el lenguaje de programación Java, distribuyéndose bajo licencia LGPL. Cumple con la especificación Simple Features Specification para SQL (OGC SFSQL, 2004) publicada por el Open Geospatial Consortium y proporciona una implementación completa, consistente y robusta de algoritmos espaciales bidimensionales.

Esta biblioteca es ampliamente utilizada en el software SIG de código libre con funciones de análisis espacial, consultas avanzadas y creación de topología. Existe una versión de esta biblioteca en C++, llamada GEOS (Geometry Engine - Open Source) (JTS, 2006).

4.2.3. Capa de Presentación

La capa de presentación es responsable de la interfaz de usuario del sistema que permite la visualización, manipulación y la entrada de datos. La capa de presentación recibe las interacciones del usuario, en el caso de la arquitectura en acciones provenientes desde el navegador *web* en forma de clics del ratón, entradas de teclado u otras entradas del dispositivo. Estas entradas se evalúan y se realizan las operaciones adecuadas invocadas en el la capa de lógica de negocio.

Cuando se devuelven los resultados, que se muestran al usuario utilizando los controles de interfaz de usuario apropiados.

El componente más importante de la capa de presentación es el servicio de interpretación, que está a cargo de la conversión de una colección de características geográficas en una colección de objetos cartográficos que se pueden representar en un dispositivo de visualización en este caso en un mapa dentro de un navegador *web*. El proceso de representación es controlada por un conjunto de definiciones de estilo, que debe definir con precisión la forma en la que cada característica geográfica deben ser prestados.

Los objetos cartográficos resultantes se muestran utilizando un servicio de

visualización del mapa, para que el usuario final pueda manipular el mapa que se está desplegando. Por ejemplo, la escala del mapa, las capas a mostrar, dibujado de objetos espaciales, selección de objetos, eliminación de objetos, etc.

Además de las manipulaciones de visualización, la pantalla del mapa debe permitir si así se requiere, y si se agregan los controles adecuados desde el *front end* del *framework*, que el usuario final pueda manipular los objetos cartográficos que se muestran en el mapa para realizar operaciones geográficas y para solicitar operaciones de procesamiento de la capa de lógica de negocio al módulo de análisis espacial. Dicho módulo asocia estas acciones a los eventos de interfaz de usuario que se producen en el mapa.

Si toda la funcionalidad del sistema se implementa en la capa de lógica de negocio, el tiempo de respuesta del sistema puede no ser lo suficientemente rápido. Con el fin de lograr una mejor respuesta en tiempos del sistema, en la capa de presentación se pueden implementar algunas funciones utilizando los objetos cartográficos para los cálculos. Esto se implementa en la capa de presentación en el módulo despliegue de mapas, y consiste en herramientas para las siguientes tareas:

- Control de la escala del mapa y la posición. Esto permite al usuario final para centrar en el mapa fenómenos de interés por medio del *zoom* y mover el mapa.
- Gestión de la leyenda de las capas. Se permite entender el mapa describiendo las entidades del mundo real que se describen por cada símbolo.

- Barra de herramientas de dibujo de objetos. Eso permite al usuario dibujar objetos espaciales tales como polígonos, líneas y puntos sobre una nueva capa de dibujo.
- Selector de Capas. Con esto se le concede al usuario la posibilidad de seleccionar que capas quiere visualizar sobre el mapa y cuáles no.

4.2.3.1 Despliegue de Mapas

Para la parte de este módulo el intérprete usado para la visualización de objetos espaciales sobre un mapa dentro del navegador *web* se utiliza la herramienta de OpenLayers, la cual brinda esa capacidad de poder obtener los objetos geográficos provenientes del servidor que entrega, ya sea la capa de datos desde los servicios WFS y WMS extendidos por GeoServer, o desde la capa de Lógica de Negocio de los datos provenientes del módulo de análisis espacial entregados por el servidor *web*.

Para lograr la parte de componentes de mayor interacción con el usuario se utilizan la biblioteca de GeoExt en conjunto con OpenLayers, la cual extiende a su vez la funcionalidad de ExtJS que es una biblioteca de Java Script para el desarrollo de aplicaciones *web* interactivas.

4.2.3.1.1 OpenLayers

OpenLayers es una biblioteca de JavaScript de código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores *web*. OpenLayers ofrece una API para acceder a diferentes fuentes de información

cartográfica en la red: Web Map Services, Mapas comerciales (tipo Google Maps, Bing, Yahoo), Web Features Services, distintos formatos vectoriales. La *Figura 20* muestra como OpenLayers puede obtener para cargar los Mapas de los servicios actualmente existentes.

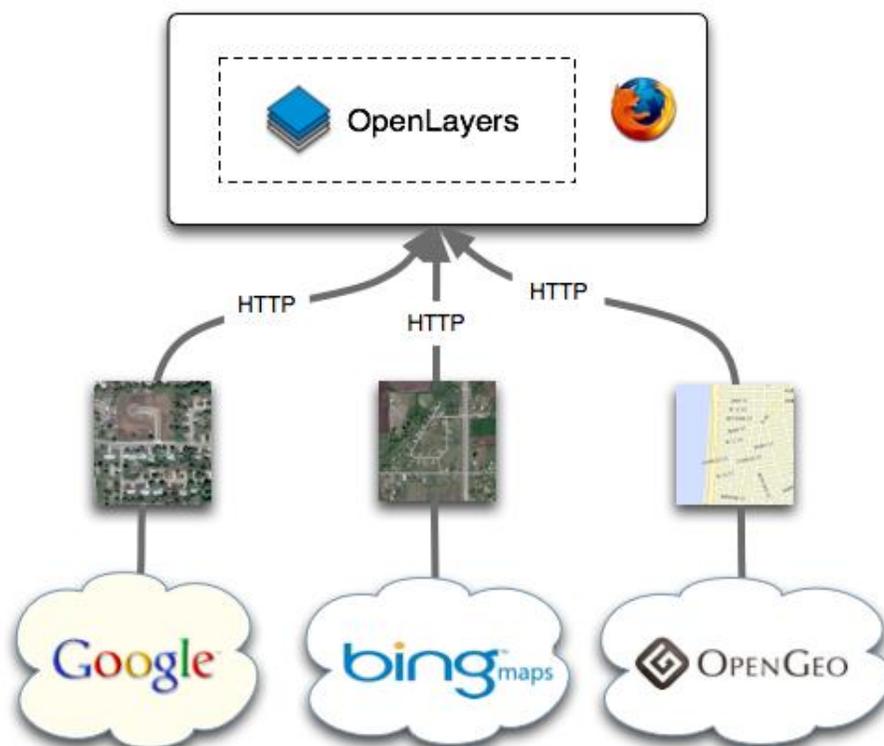


Figura 20 Interacción OpenLayers con Servicios de Mapas

4.2.3.1.2. ExtJS

ExtJS es una biblioteca de JavaScript para el desarrollo de aplicaciones *web* interactivas usando tecnologías como AJAX, DHTML y DOM. Fue desarrollada por Sencha. Originalmente construida como una extensión de la biblioteca YUI por Jack Slocum, en la actualidad puede usarse como extensión para

las bibliotecas jQuery y Prototype. Desde la versión 1.1 puede ejecutarse como una aplicación independiente (ExtJS, 2013).

Dispone de un conjunto de componentes (*widgets*) para incluir dentro de una aplicación *web*, como:

- Cuadros y áreas de texto.
- Campos para fechas.
- Campos numéricos.
- Combos.
- Radiobuttons y checkboxes.
- Editor HTML.
- Elementos de datos (con modos de sólo lectura, datos ordenables, columnas que se pueden bloquear y arrastrar, etc.).
- Árbol de datos.
- Pestañas.
- Barra de herramientas.
- Menús al estilo de Windows.
- Paneles divisibles en secciones.
- *Sliders*.
- Gráficos

4.2.3.1.3. GeoExt

La biblioteca GeoExt es una herramienta utilizada para construcción de aplicaciones *web* de mapas basada en JavaScript. Combina los controles geoespaciales de OpenLayers con los componentes de interfaz de usuario de Ext JS en un *framework* que nos permite construir aplicaciones SIG de estilo similar a las de escritorio, pero en un navegador (GeoExt, 2014).

Por último en la *Figura 21* podemos ver la Arquitectura nuevamente, pero ahora con las dependencias de las API's y Frameworks utilizados.

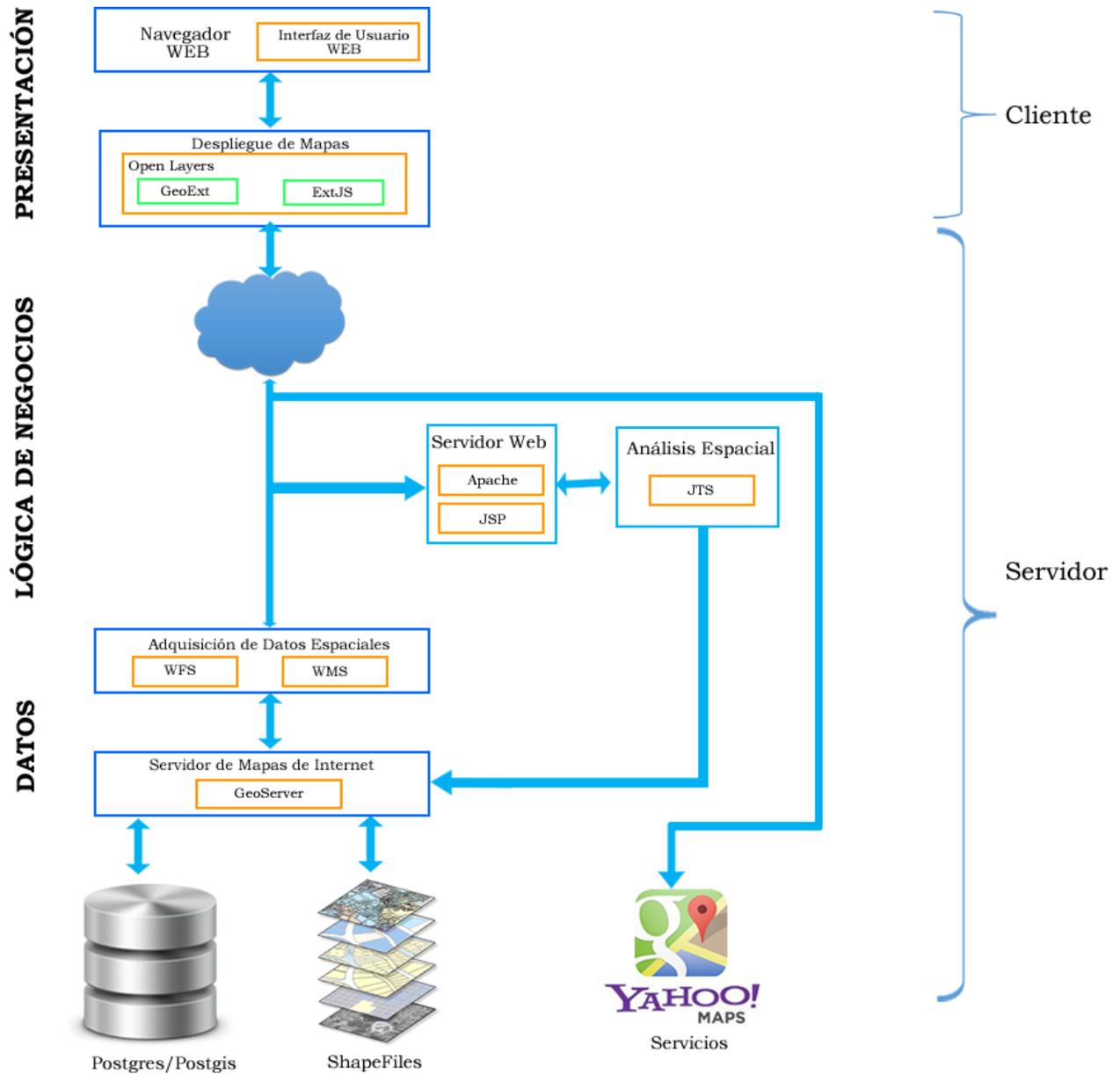


Figura 21 Dependencias de la Arquitectura Propuesta

Capítulo 5. Resultados

En este capítulo se presentan algunos diagramas UML de los resultados obtenidos para entender mejor los flujos de la aplicación, una explicación del funcionamiento de la herramienta desarrollada y la comparación en cuanto a tiempos contra *frameworks* similares existentes en el mercado mediante la implementación de un caso de estudio.

5.1. Diseño del Sistema

En este apartado se muestran algunos diagramas de Caso de Uso y Secuencia para visualizar con mayor claridad los flujos del *framework* y la interacción del usuario con el Front End.

5.1.1. Diagrama de Caso de Uso

En la [Figura 22](#) se muestra el diagrama general de Caso de Uso para entender mejor la interacción del usuario, en este caso el desarrollador de aplicaciones web SIG, con el Front End del *framework*.

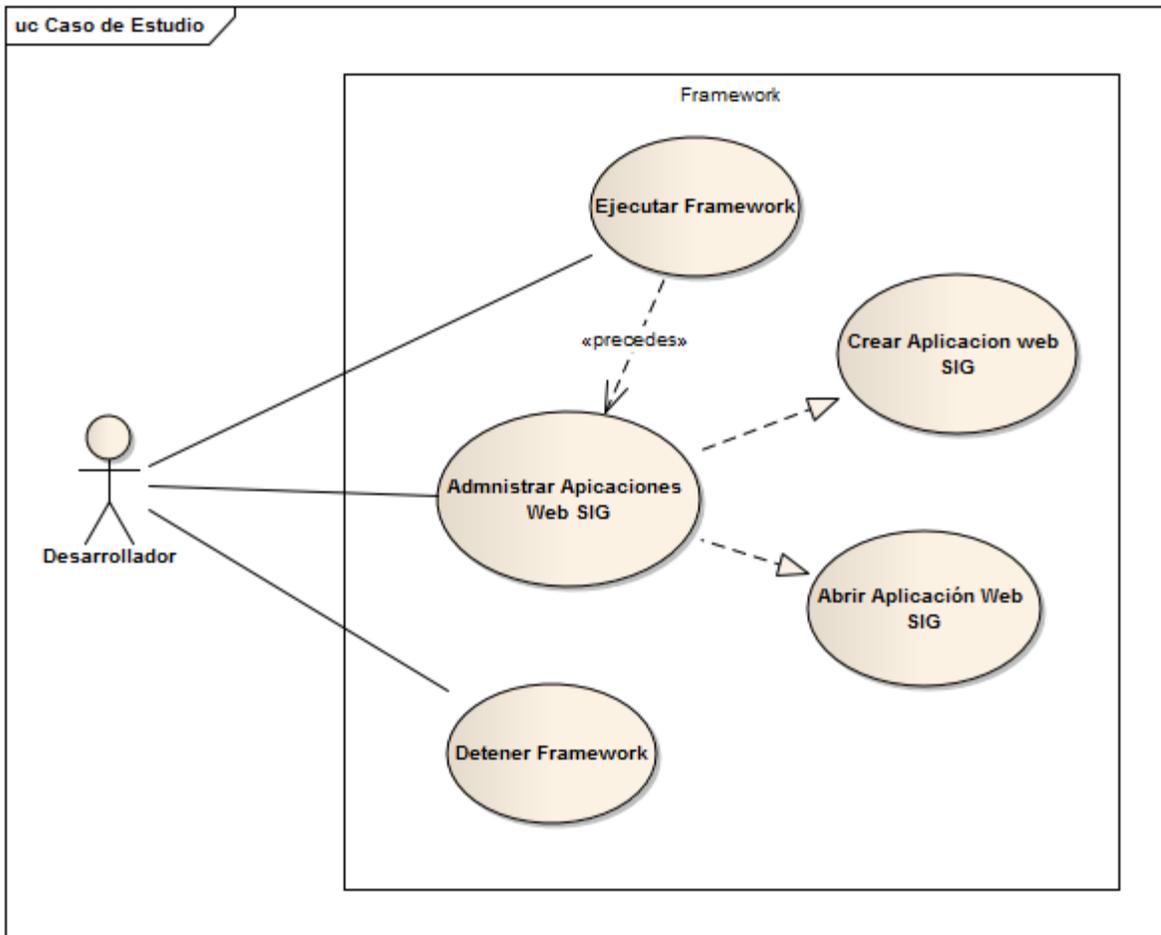


Figura 22. Diagrama General de Caso de Uso

5.1.2. Diagramas de Secuencia

Se muestran los diagramas de secuencia que ayudarán a comprender los flujos de las actividades y la participación del Back End y Front End en cada una de ellas.

5.1.2.1. Diagrama de Secuencia: Ejecutar Framework

En la [Figura 23](#) se muestra el Diagrama de Secuencia correspondiente a la actividad de Ejecutar el Framework.

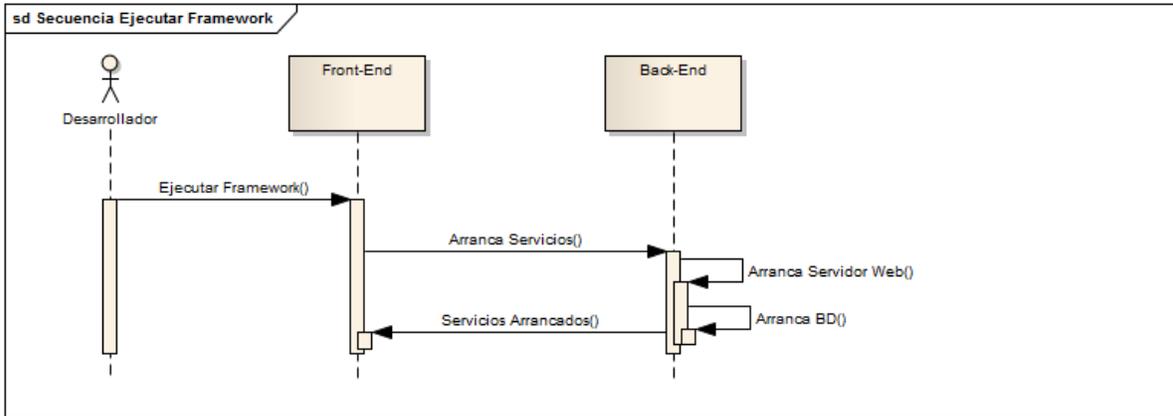
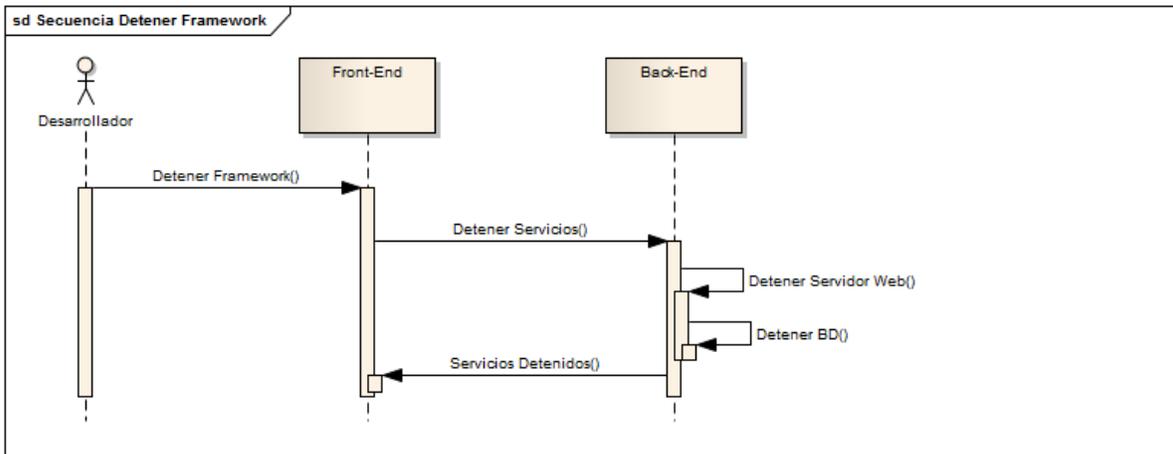


Figura 23. Diagrama de Secuencia: Ejecutar Framework

5.1.2.2. Diagrama de Secuencia: Detener Framework

En la *Figura 24* se muestra el Diagrama de Secuencia correspondiente a la actividad de Detener el Framework.

Figura 24. Diagrama de Secuencia: Detener Framework



5.1.2.3. Diagrama de Secuencia: Crear Aplicación

En la *Figura 25* se muestra el Diagrama de Secuencia correspondiente a la actividad de Crear una nueva aplicación web SIG.

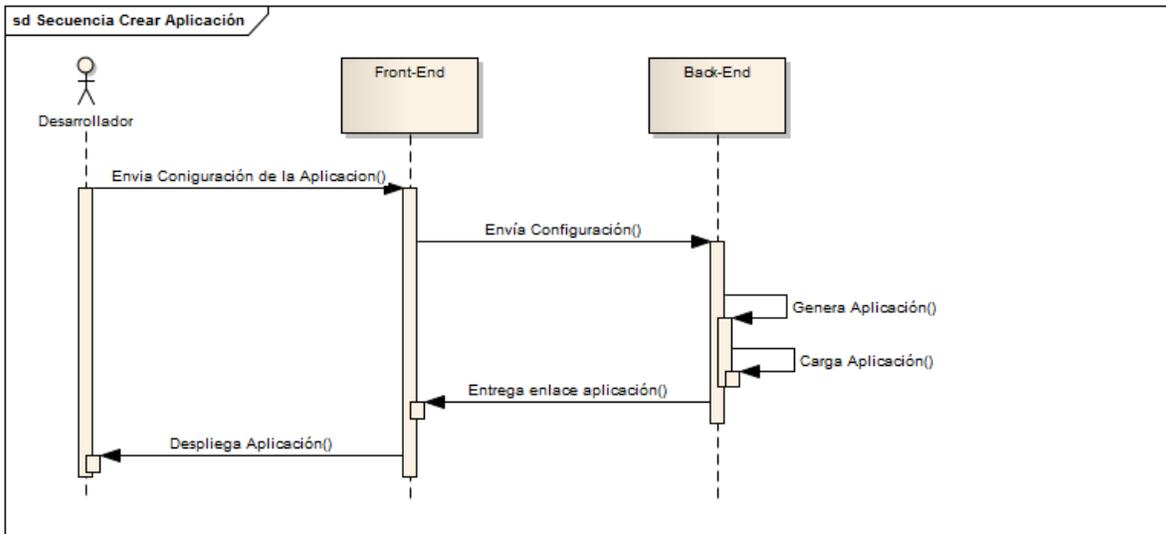


Figura 25. Diagrama de Secuencia: Crear Aplicación

5.2. Ejecutando el *framework*

Para ejecutar el Framework basta con ejecutar un archivo por lotes, esta explicación se muestra en el Anexo 1 de este documento.

Una vez ejecutados todos los servicios necesarios del *framework* se podrá abrir el navegador en la dirección <http://localhost:8080> para observar la interfaz *web* inicio del Front End del *framework* como se observa en la *Figura 26* desde donde podremos administrar las aplicaciones Web SIG creadas, así como la administración de las fuentes de datos espaciales que es llevada a cabo desde la interfaz *web* de Geoserver.

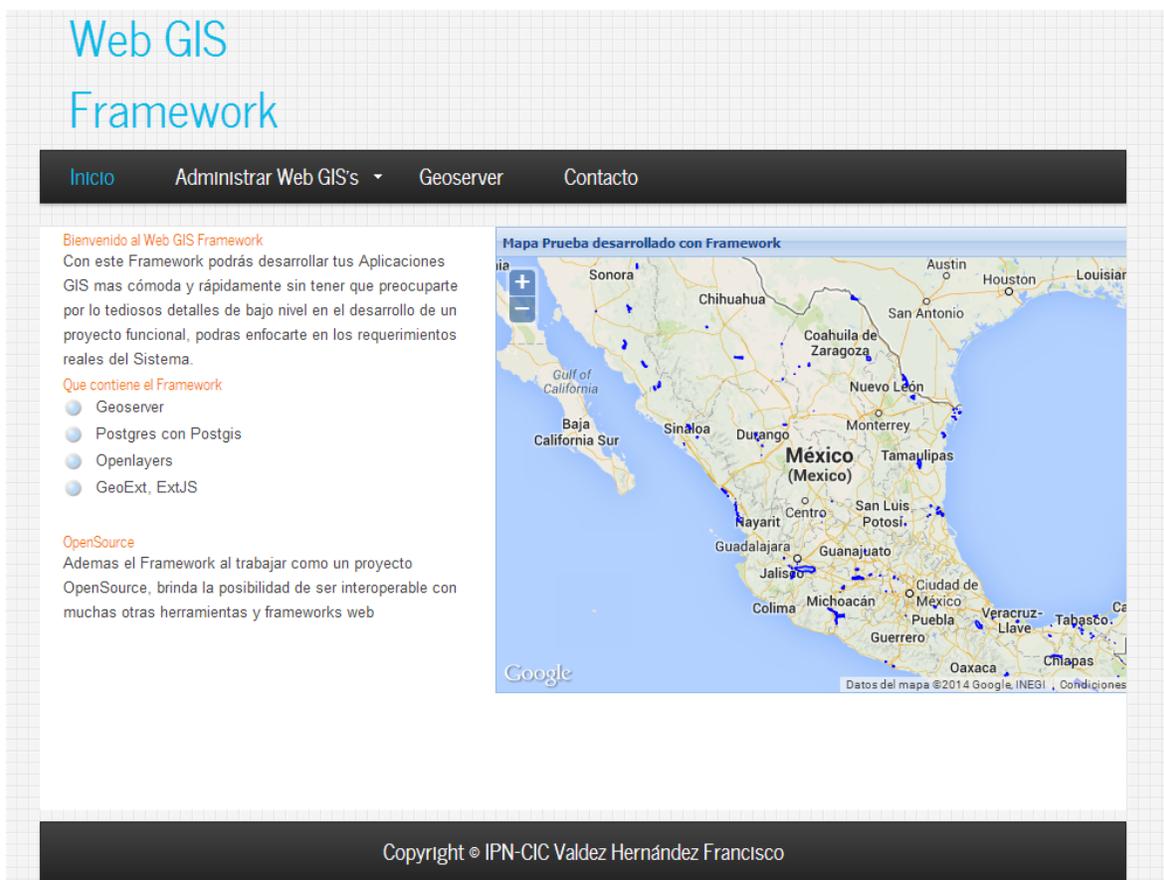


Figura 26. Página de inicio del Framework

5.3. Administrando las Fuentes de Datos Geoespaciales

Como se ha descrito anteriormente la gestión de las fuentes de datos geoespaciales es llevada a cabo por el servidor de mapas Geoserver, el cual como se ha mencionado en capítulos anteriores permite la administración de fuentes de datos geoespaciales entre las que destacan Bases de datos Espaciales como Oracle spatial, PostgreSQL con su extensión de PostGIS, entre otras, también directamente de

archivos shape de ESRI. Geoserver cuenta con una interfaz *web* muy amigable para facilitar dicha administración, podemos acceder a ella directamente desde la página inicial del *framework* *Figura 27*



Figura 27. Acceso a Geoserver

Al acceder veremos la siguiente pantalla de inicio de Geoserver *Figura 28*

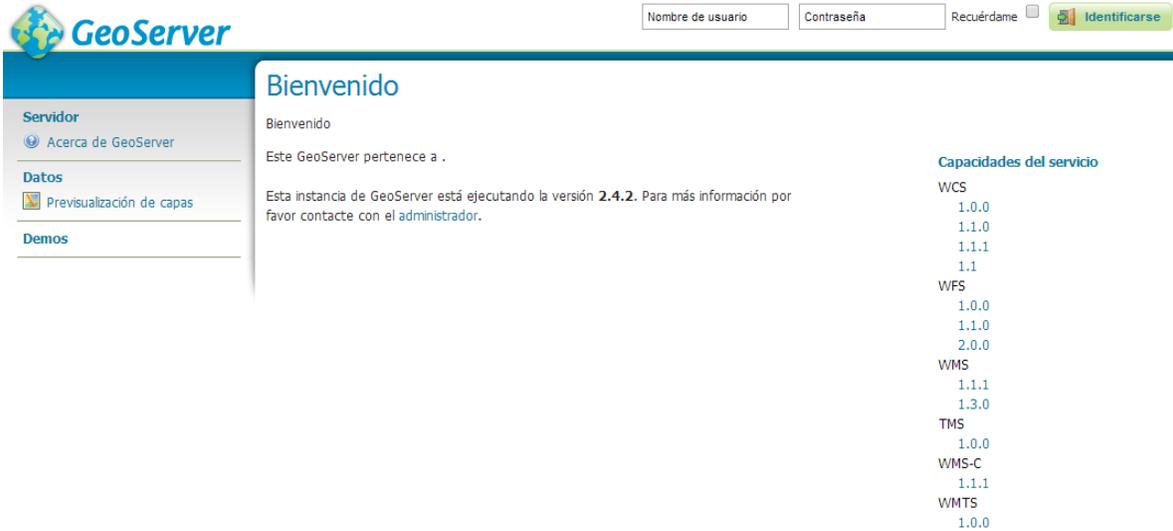


Figura 28. Página inicial Geoserver

Para acceder al sistema se requiere de un login que por default tenemos como nombre de usuario: **admin** y password: **geoserver**, al introducir el login veremos la pantalla inicial del administrador de geoserver que como principales funciones tendremos las de agregar capas nuevas, agregar almacenes de datos, pre visualizar capas, etc.



Figura 29. Menú principal de Geoserver

5.4. Caso de Estudio

En el siguiente apartado se demuestra el uso del framework desarrollado aplicando un caso de estudio utilizando datos geoespaciales de vialidades y transportes del Distrito Federal.

Para la creación la aplicación Web SIG utilizando el framework de este trabajo, se parte del enlace en la página de inicio como se muestra en la *Figura 30*



Figura 30. Crear nueva Web SIG

Al acceder a la parte de Crear nueva Web SIG, se observa la siguiente pantalla *Figura 31*, con un formulario desde donde se configuran los componentes y capas que integrarán la aplicación.

Crear Aplicación Web SIG

Configuración de Mapa

Nombre de la Aplicación:

Tipo de Mapa: Fijo Flotante

Tamaño del Mapa: H: px W: px

Capas de Datos

Geoserver WMS:

cuerpos_agua4m
metrobus_rutas

calles_alvaro_obregon
calles_azcapotzalco
calles_cuauhtemoc
contorno_area_estudio
metro_estaciones
metro_rutas

[Administrar capas \(Geoserver\)](#)

Agregar Capa base Google Maps

Controles

Panel de Selección de Capas

Panel de Leyendas

Barra de Navegación

Barra de Herramientas de Dibujo
 Punto Línea Polígono Texto

Constructor de Consultas

Barra de Herramientas de Funciones
 Buffer Unión Intersección

Figura 31. Formulario para crear nueva Web SIG

Como se observa en *Figura 31* tenemos un formulario compuesto de 3 secciones: Mapa, Capas de Datos y Controles. En la parte de Mapa se pide el nombre que

llevará la aplicación, el tipo de mapa que puede ser fijo o flotante; un mapa flotante sería colocado en una ventana movable dentro del navegador, en el ejemplo mostrado, se selecciona mapa flotante; también se requiere especificar un tamaño de mapa en cuanto alto y ancho en pixeles, en el ejemplo lo definimos a 500x1000 px.

En la parte de Capas de Datos, se seleccionan las capas temáticas que se incluirán a mostrar en el mapa, dichas capas son obtenidas del servicio brindado por Geoserver, WMS, por lo cual se observan las capas que previamente hayan sido gestionadas desde Geoserver, en el caso de estudio a implementar se integran las alguna capas dadas de alta en Geoserver: Vialidades de algunas delegaciones del Distrito federal y así como las rutas y estaciones del transporte público metro, y una capa de la red del transporte público metro, justo debajo de la lista de capas disponibles se tiene un enlace que lleva directo a la plataforma web de Geoserver.

Para la parte de controles se seleccionan las herramientas que serán agregadas junto con el mapa, dentro de las posibles opciones se tienen: un panel para selección de capas, un panel para mostrar las leyendas de las capas temáticas, una barra de navegación, una barra de herramientas para dibujado de elementos espaciales dentro de los cuales se tienen las opciones de polígono y línea, punto y texto, y un Constructor de Consultas (*Query Builder*), y por último se tiene la opción de mostrar una barra de herramientas con algunas opciones desarrolladas para el uso de operaciones de análisis espacial de entre las cuales se tiene: buffer, unión e intersección. Para el caso de estudio se seleccionan todas las opciones de controles a mostrar, para así mostrar el funcionamiento de cada una de ellas.

Al finalizar y dar click al botón de crear se muestra inmediatamente la aplicación creada como se ve en la *Figura 32*.

Vialidades_DF

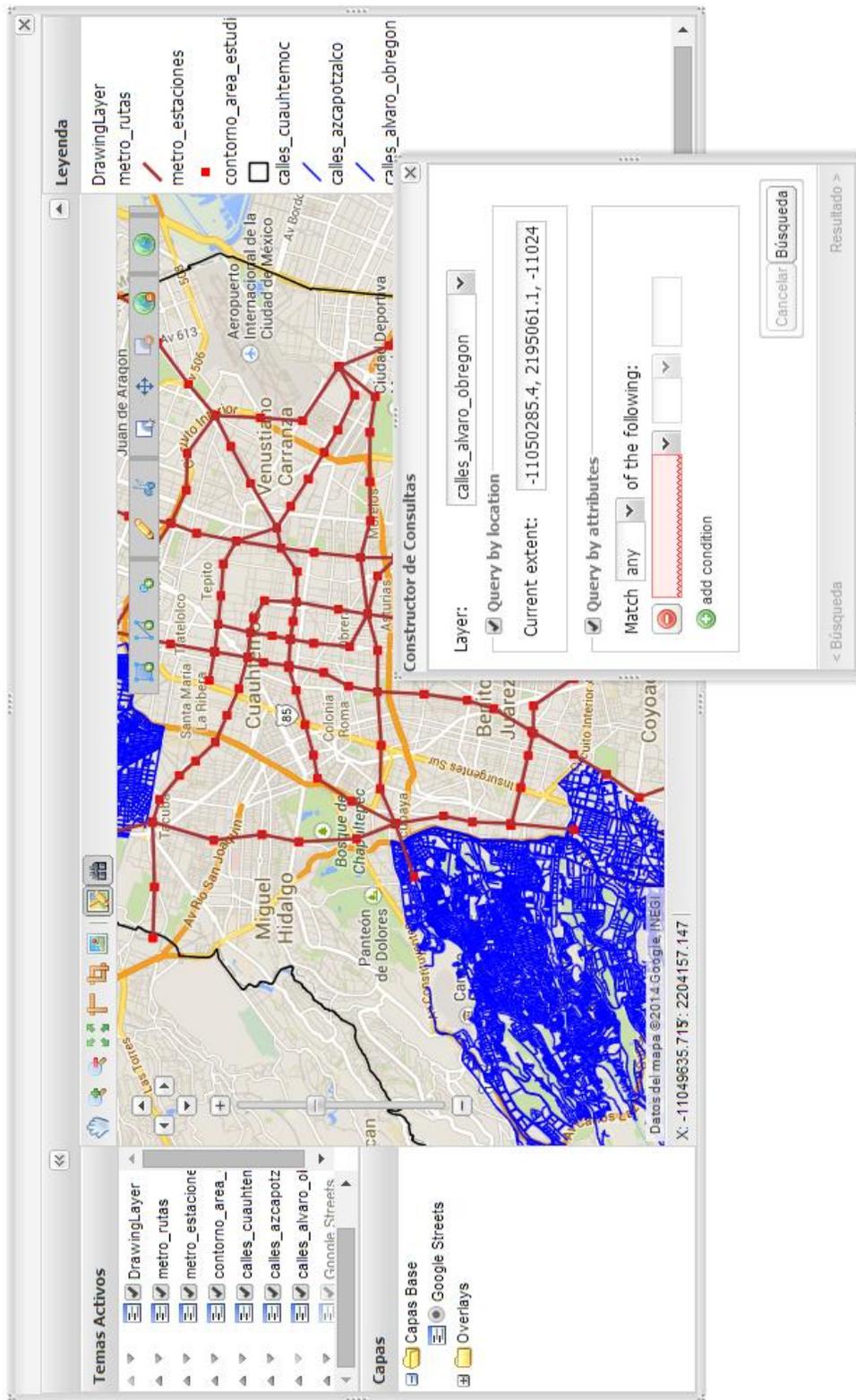


Figura 32. Aplicación creada de Caso de Estudio

Como se observa en la imagen anterior podemos ver que la aplicación web SIG generada cuenta con todas las herramientas seleccionadas y así como las capas temáticas a mostrar, y todos los controles solicitados a incluir.

5.3.1. Dibujado de elementos espaciales

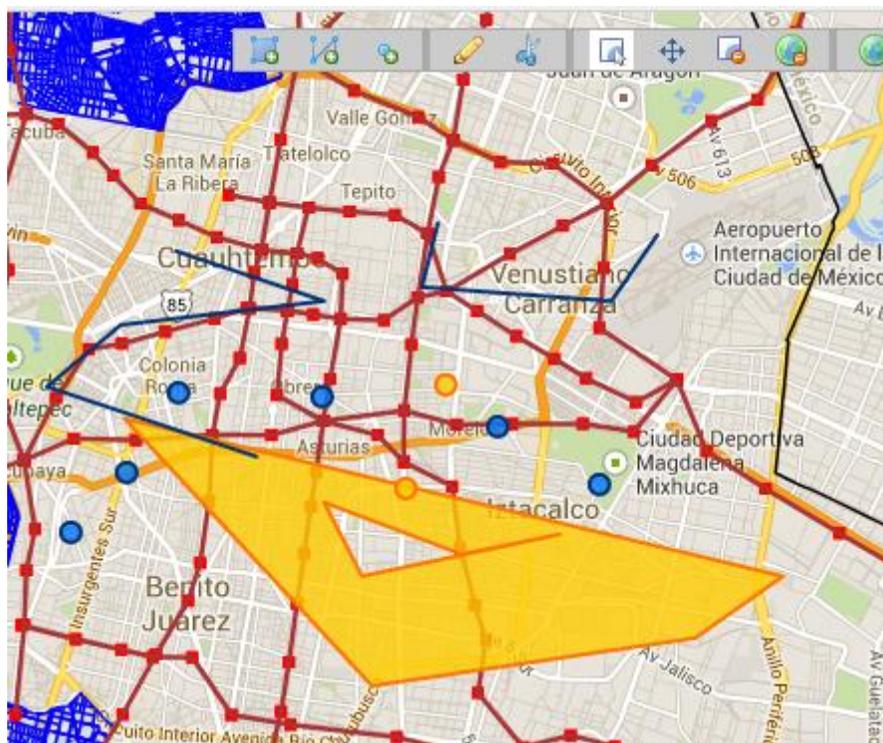


Figura 33. Dibujando Polígonos y Líneas

La *Figura 33* muestra cómo se han utilizado las herramientas de dibujo para insertar en el mapa algunos polígonos y líneas.

Cabe mencionar, que todos los elementos dibujados se van realizando dentro de una capa especial llamada *Capa de Dibujado*.

5.3.2. Panel de Selección de Capas

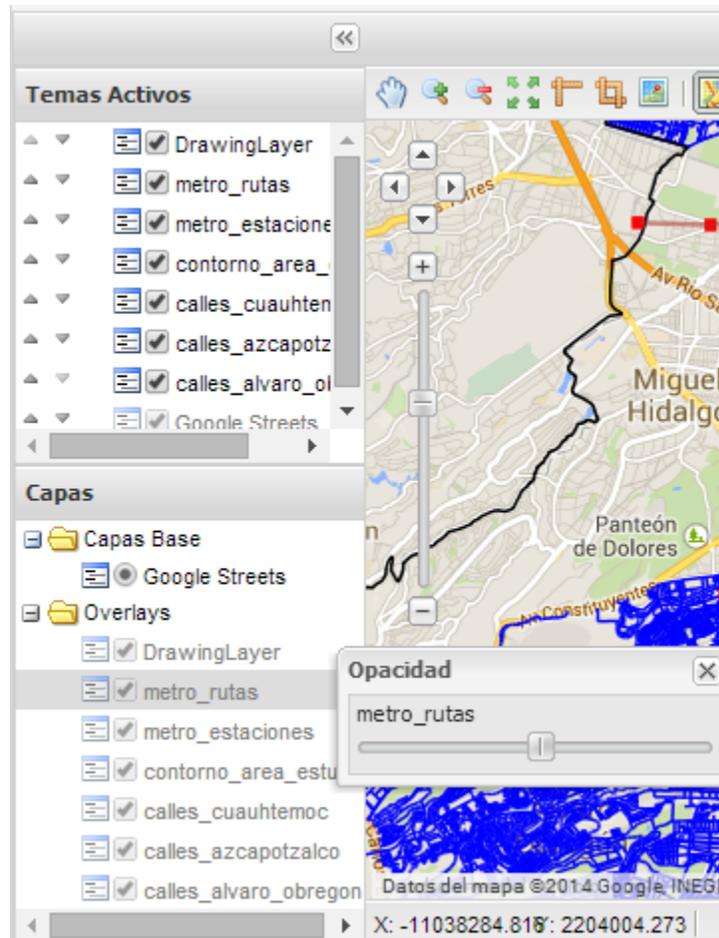


Figura 34 Panel de Selección de Capas

Como se muestra en la Figura 30, y como una de las opciones seleccionadas en nuestra aplicación de caso de estudio, se tiene un panel de Selección de Capas, el cual nos permite habilitar o inhabilitar las capas temáticas que hayamos seleccionado y adicionalmente también cambiar la opacidad de las mismas.

5.3.3. Panel de Leyendas

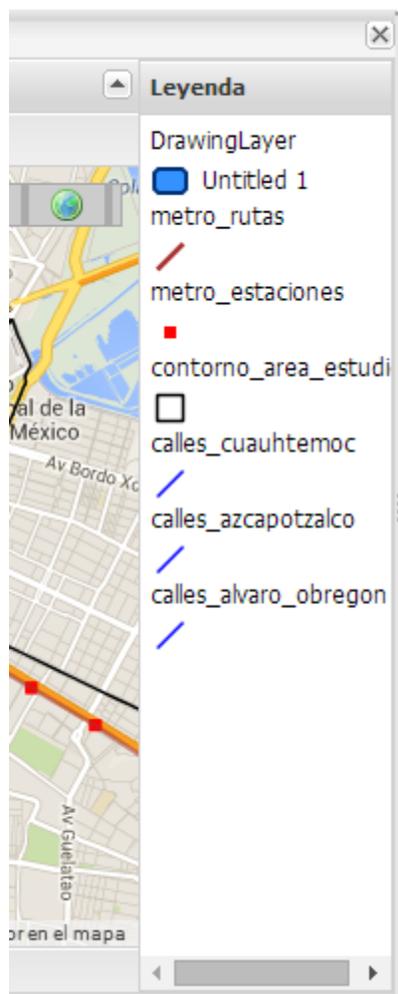


Figura 35 Panel de Leyendas

En la Figura 31 se muestra justo del lado derecho de la aplicación un panel donde se observan las Leyendas de las capas temáticas que se estén visualizando en el momento.

5.3.4. Constructor de Consultas

Los Archivos vectoriales o shape files, como se ha descrito anteriormente, muchas veces contienen atributos asociados a ellos, que nos muestran cierta información relevante, en nuestro caso de estudio se utilizaron datos de ciertos elementos referentes a vialidades y transporte en el Distrito Federal , que además de la información vectorial contienen atributos que pueden ser consultados mediante el control de Constructor de Consultas o Query Builder, desde donde podremos crear consultas de manera sencilla sin requerir de un lenguaje de consultas especializado. Dicho control se observa en la [¡Error! No se encuentra el origen de la referencia.2](#) desde donde se pueden realizar consultas por locación o extensión actual visible, y también realizar consultas por atributos de los vectores, se pueden agregar tantas condiciones como sean necesarias y realizar la consulta de acuerdo a coincidencia por alguna o todas las condiciones agregadas.

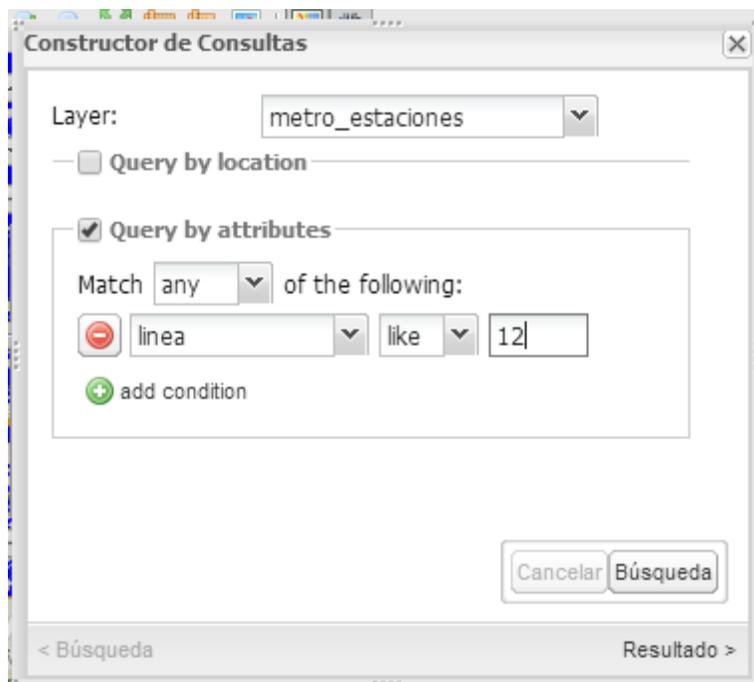
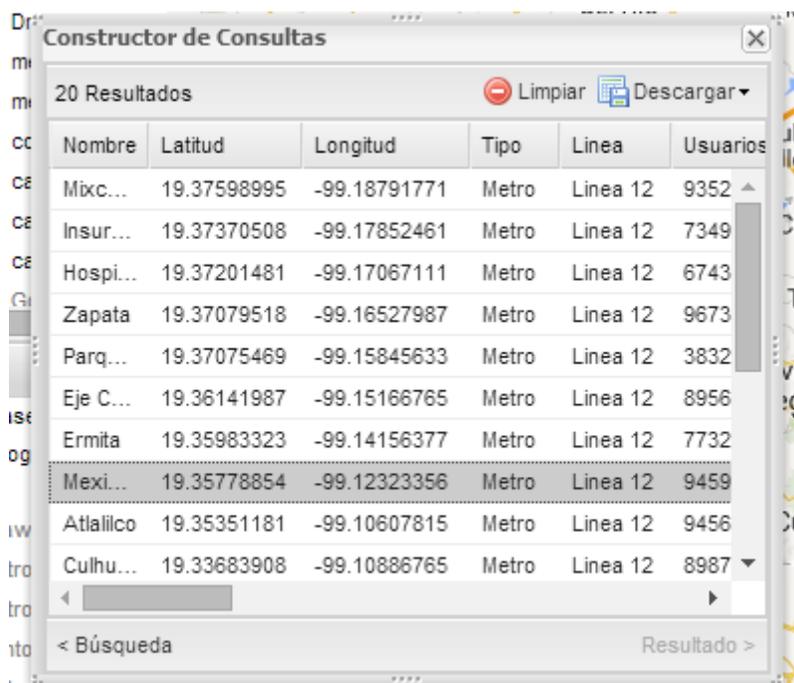


Figura 36. Constructor de Consultas

Y el resultado se muestra en el mismo panel como se observa en la Figura 33.



The screenshot shows a window titled 'Constructor de Consultas' with a table of 20 results. The table has columns for 'Nombre', 'Latitud', 'Longitud', 'Tipo', 'Linea', and 'Usuarios'. The results are for Linea 12 Metro stations. The 'Mexi...' row is highlighted.

Nombre	Latitud	Longitud	Tipo	Linea	Usuarios
Mixc...	19.37598995	-99.18791771	Metro	Linea 12	9352
Insur...	19.37370508	-99.17852461	Metro	Linea 12	7349
Hospi...	19.37201481	-99.17067111	Metro	Linea 12	6743
Zapata	19.37079518	-99.16527987	Metro	Linea 12	9673
Parq...	19.37075469	-99.15845633	Metro	Linea 12	3832
Eje C...	19.36141987	-99.15166765	Metro	Linea 12	8956
Ermita	19.35983323	-99.14156377	Metro	Linea 12	7732
Mexi...	19.35778854	-99.12323356	Metro	Linea 12	9459
Atlalico	19.35351181	-99.10607815	Metro	Linea 12	9456
Culhu...	19.33683908	-99.10886765	Metro	Linea 12	8987

Figura 37. Resultados Consulta

5.5. Comparación de Resultados

Se ha demostrado la funcionalidad del trabajo desarrollado en este documento, para finalizar se ha desarrollado la misma aplicación que en el caso de estudio pero utilizando algunos de los *frameworks* comerciales presentados en el estado del arte, con la finalidad de comparar el tiempo de desarrollo y las funciones que aportan cada *framework*. Los resultados se muestran en la siguiente Tabla:

Tabla 1. Comparación de Resultados

Framework	Tiempo de Desarrollo	Análisis Espacial	SGBDE	Seguridad
Framework desarrollado	10 mins.	si	PostgreSQL-PostGIS	No
Mapbender	1 hora 12 mins.	No	PostgresSQL-PostGIS	No
Geomajas	2 horas 24 mins.	Si (con Plugin)	PostGIS, Oracle Spatial, Microsoft SQL Server, MySQL	No
Mapfish	3 horas 31 mins.	No	Postgres-PostGIS	Si

Como se observa en la tabla anterior, se creó una herramienta que es capaz de minimizar los tiempos a la hora del desarrollo de una aplicación Web SIG en comparación con los *frameworks* comerciales, un punto muy fuerte del *framework* desarrollado es que se integró la parte de Análisis espacial que aunque aún son pocas las operaciones posibles a realizar, es el único *framework* que tiene integrado la parte de Análisis espacial, Geomajas lo logra pero a través de un plugin extra.

Por otra parte se tienen ciertas limitantes como son la parte de Seguridad que no fue implementada y que otro *framework* como lo es MapFish tiene un módulo de Seguridad bien implementado. Dicho módulo de Seguridad se refiere al aseguramiento de los datos que viajan de la red, Mapfish desarrolló su propio protocolo de transferencia de información geoespacial que está completamente cifrado.

Se cumple con el objetivo de minimizar el esfuerzo de aprendizaje del *framework* dado que no se requiere de tener que codificar una sola línea de código para lograr crear una aplicación Web SIG completa y funcional y de reducir los tiempos de desarrollo, además de recordar que se han utilizado herramientas completamente

Open Source, por lo cual la herramienta queda a ser extendida por la comunidad de desarrolladores en el área de Geoespacial.

Capítulo 6. Conclusiones

En el siguiente capítulo se muestran las conclusiones del trabajo haciendo énfasis en las aportaciones del trabajo, los alcances y limitaciones del mismo; así como los trabajos a futuro, tomando como base el trabajo de esta tesis.

La información geográfica se está convirtiendo poco a poco en un elemento importante en los sistemas informáticos.

Muchas aplicaciones se están desarrollando para tareas industriales, administrativas y de investigación en los que la información geográfica es el componente central.

Sin embargo, la información geográfica es un tipo especial de información que no puede ser representada, manipulada y visualizada usando los métodos que se utilizan tradicionalmente con otro tipo de información. La información geográfica requiere métodos especiales de análisis y modelado.

La primera contribución de este trabajo es un análisis de las características especiales que hacen que los sistemas de información geográfica difieran de los sistemas de información tradicionales.

Además, a pesar de que la OGC y la ISO han trabajado en las especificaciones de las normas relativas a la información geográfica y han sentado las bases para la generación de aplicaciones SIG y herramientas de desarrollo que son capaces de apoyar en los distintos niveles del desarrollo, actualmente no existe una arquitectura genérica para los Web SIG que tenga en cuenta la especial naturaleza y características de la información geográfica y, al mismo tiempo los requisitos bien conocidos para los sistemas de información de propósito general.

La principal contribución de este trabajo es una propuesta de una arquitectura genérica de 3 capas cuyo diseño está basado en el análisis de las características especiales de la información geográfica. Esta arquitectura cumple con las especificaciones y estándares con los que las distintas organizaciones han establecido.

Otro principal aporte ha consistido en la creación de una interfaz web desde donde el desarrollador puede crear las aplicaciones Web SIG con la arquitectura ya mencionada y sin tener que escribir una sola línea de código, logrando así un desarrollo ágil de aplicaciones Web SIG completas y funcionales.

Como caso de estudio se ha presentado el proceso del desarrollo completo de un Web SIG utilizando la herramienta de este trabajo y con información geográfica de vialidades y transporte del Distrito Federal, y también se ha desarrollado la misma aplicación con algunos de los *frameworks* comerciales, y se han comparado y analizado los resultados en cuanto a tiempos y facilidad de desarrollo, comprobando que la herramienta ayuda a agilizar la creación de aplicaciones Web SIG.

6.1. Alcances y Limitaciones

Como es de esperarse la metodología propuesta tiene sus limitaciones, por ello a continuación se presentan los alcances y limitaciones de las diferentes partes de la metodología:

- Con respecto al soporte de motores de bases de datos espaciales, y aunque en la arquitectura propuesta es posible trabajar con distintos SGBDE, solo se trabajó con conexión a PostgreSQL con su extensión PostGIS y con MySQL.

- Dado que se utiliza una interfaz web para la creación de las aplicaciones Web SIG para evitar la codificación por parte del desarrollador, se tiene la limitante de que las aplicaciones resultantes terminan siendo muy similares, gráficamente hablando, no existe una personalización.
- En cuanto a la parte de las funciones de análisis espacial, no se contemplan todas las existentes, solo se programaron algunas de las principales quedando pendiente la extensión o inclusión de más propiedades y/o funciones.
- Al utilizar archivos de procesamiento por lotes (.bat) para levantar todos los servicios, el framework está limitado a trabajar únicamente bajo la plataforma windows.

6.2. Trabajo a futuro

Como consecuencia de los alcances y limitaciones presentados, es posible puntualizar algunos aspectos que en un futuro pudieran ser desarrollados para ampliar el trabajo presentado:

- Como se mencionó, actualmente la herramienta únicamente brinda soporte a conexión con PostgreSQL con PostGIS, y a MySQL en un futuro debería brindar un soporte más amplio en cuanto a bases de datos espaciales, como puede ser Oracle Spatial, SpatialDB, DB2, etc.
- Para lograr una mayor personalización de las interfaces de las aplicaciones Web SIG creadas, es conveniente desarrollar una api o de ser posible un

nuevo lenguaje de alto nivel para la creación de las interfaces web del cliente.

- Como ya se ha expuesto queda pendiente la extensión o inclusión de nuevas funciones de análisis espacial que no fueron implementadas para la parte del módulo de análisis espacial.
- El framework no posee la cualidad de ser multiplataforma debido a que los scripts de levantamiento del ambiente únicamente trabajan bajo la plataforma Windows, se pueden crear scripts que puedan ser ejecutados en otras plataformas, para así poder ejecutar el framework en cualquier Sistema Operativo.

Capítulo 7. Referencias

Andreas Neumann (2007). *Web Mapping and Web Cartography*. Pag. 1262.

Apache (2014). Apache Tomcat. Consultado el 19 de febrero de 2014. Disponible en: <http://tomcat.apache.org/>

Baumann, P. (2012). OGC WCS 2.0 interface standard –core. Open Geospatial Consortium Inc., Wayland, MA, USA, OpenGIS® Interface Standard OGC.

Brisaboa, N. R., Lema, J. A. C., Luaces, M. R., & Viqueira, J. R. (2001). State of the Art and Requirements in GIS, 3er. Encuentro Internacional de Ciencias de la Computación (ENC'01). Aguascalientes, México.

Byong-Lyol L, Young-Chan K, Jin-I Y (1998) *Web interface for GIS in Agriculture*. By The Asian Federation for Information Technology in Agriculture.

De la Beaujardière, J. (2006). OpenGIS® *Web Map Server Implementation Specification*. Version 1.3. 0, OGC 06-042, Open Geospatial Consortium. Inc. 85.

DocForge. (2008). Framework. Consultado el 22 de Noviembre del 2013. Disponible en: <http://docforge.com/wiki/Framework>.

Dragicevic, S. (2004). The potential of *Web-based GIS*. *Journal of Geographical Systems*, 6(2), 79-81.

ESRI, E. (1998). Shapefile technical description. An ESRI White Paper.

ExtJS (2013). Sencha ExtJS. Consultado 13 de Febrero del 2014. Disponible en: <http://www.sencha.com/products/extjs/>

Gaffuri, J. (2012). Toward *web mapping with vector data*. In *Geographic Information Science* (pp. 87-101). Springer Berlin Heidelberg.

GeoExt (2013). JavaScript Toolkit for Rich Web Mapping Applications. Consultado el 13 de Febrero del 2014. Disponible en: <http://www.geoext.org/>

Geomajas (2012). Geomajas OSGeo Project. Consultado el 23 de Abril del 2013. Disponible en: <http://www.geomajas.org/>

Geoserver (2013). Geoserver OSGeo Project. Consultado el 35 de Abril del 2013. Disponible en: <http://geoserver.org/>

Geotools. (2013). Java GIS Toolkit. Consultado el 23 de Abril del 2013. Disponible en: <http://www.geotools.org/>

Gillavry E. M. (2000). Cartographic aspects of *WebGIS*-software, Doctoral dissertation Tese (PhD).

GML, O. (2003). Geography Markup Language (GML) 3.0 Implementation Specification. OpenGIS Consortium.

Helali H (2001) Design and Implementation of a *Web GIS* for the City of Tehran. MSc thesis, Department Of Geodesy And Geomatics Engineering K.N.Toosi University Of Technology, Tehran, Iran.

JSP (2014). Java Server Pages. Consultado el 10 de Febrero del 2014. Disponible en: http://es.wikipedia.org/wiki/JavaServer_Pages

JTS (2006) Java Topology Suite. Consultado el 8 de Febrero del 2014. Disponible en: <http://www.vividsolutions.com/jts/JTSHome.htm>

Kraak, M. J., & Ormeling, F. (2011). Cartography: visualization of spatial data. Guilford Press.

Leukert K, Reinhardt W (2000) GIS-Internet Architecture. International Archives of Photogrammetry and Remote Sensing. Vol. XXXIII, Part B4, Amsterdam 2000.

Longley, P., Goodchild, M., Maguire, D., Rhind, D. (1999). Geographical Information Systems: Principles, Techniques, Management and Applications. John Wiley & Sons ISBN: 0-471-32182-6.

Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2001). Geographic information system and Science. England: John Wiley & Sons, Ltd.

Mapbender (2013). Mapbender Project. Consultado el 18 de Abril del 2013. Disponible en: http://www.mapbender.org/Mapbender_Wiki

Mapfish (2011). Mapfish. Consultado el 18 de Abril del 2013. Disponible en: <http://mapfish.org/>

MapServer (2013). MapServer opensource web mapping. Consultado el 17 de Abril del 2013. Disponible en: <http://mapserver.org/>

Nayak, S (2000) GIS Data Dissemination: A New Approach through WEB Technology. Rolta India.

OpenLayers (2013). OpenLayers Free Maps for the web. Consultado el 17 de Abril del 2013. Disponible en: <http://openlayers.org/>

Open Geospatial Consortium. (2004). Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option 1.2.1.

Open Geospatial Consortium. (2006). OpenGIS Web Map Service version 1.3. 0.

Ostensen, O. M., & Smits, P. C. (2002). ISO/TC211: Standardisation of geographic information and geo-informatics. In Geoscience and Remote Sensing Symposium, 2002. IGARSS'02. 2002 IEEE International (Vol. 1, pp. 261-263). IEEE.

Percivall, G., Reed, C., Leinenweber, L., Tucker, C., & Cary, T. (2003). OGC Reference Model. Open Geospatial Consortium Inc, 1-108.

Peterson, M. P. (2003). Maps and the Internet. Elsevier

PostGIS. (2013). Spatial and Geographic objects for PostgreSQL. Consultado el 7 de Mayo del 2013. Disponible en: <http://postgis.net/>

PostgreSQL. (2013). PostgreSQL. Consultado el 7 de Mayo del 2013. Disponible en: <http://www.postgresql.org.es/>

Riehle, D. (2000). *Framework design*. Doctoral dissertation, Diss. Technische Wissenschaften ETH Zürich.

Rigaux, P., Scholl, M., & Voisard, A. (2002). Spatial Databases with Application to GIS. *SIGMOD Record*, 32(4), 111.

Tomlinson, R.F. (2005) *Thinking About GIS: Geographic Information System Planning for Managers*. ESRI Press.

WFS, O. (2005). *Web Feature Service*. OGC. URL: <http://www.opengeospatial.org/standards/wfs>.

Anexos

A.1. Ejecutando el Framework

Para la fácil instalación del Framework desarrollado, se ha creado un único empaquetado .rar donde ya se integran las últimas versiones de las API's y *Frameworks* utilizadas para el correcto funcionamiento de la arquitectura mencionada en el metodología, no es necesaria la instalación de ninguna otra herramienta, solo se necesita desempaquetar el archivo en cualquier ubicación, mostrando así que incluso el trabajo de instalación del *framework* es casi nulo.

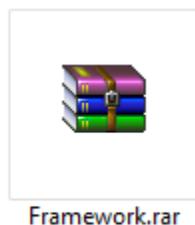


Figura 38. Archivo RAR del Framework

Al desempaquetar el archivo se contendrán los archivos mostrados en la [Figura 39](#).
Archivos incluidos dentro del RAR del Framework

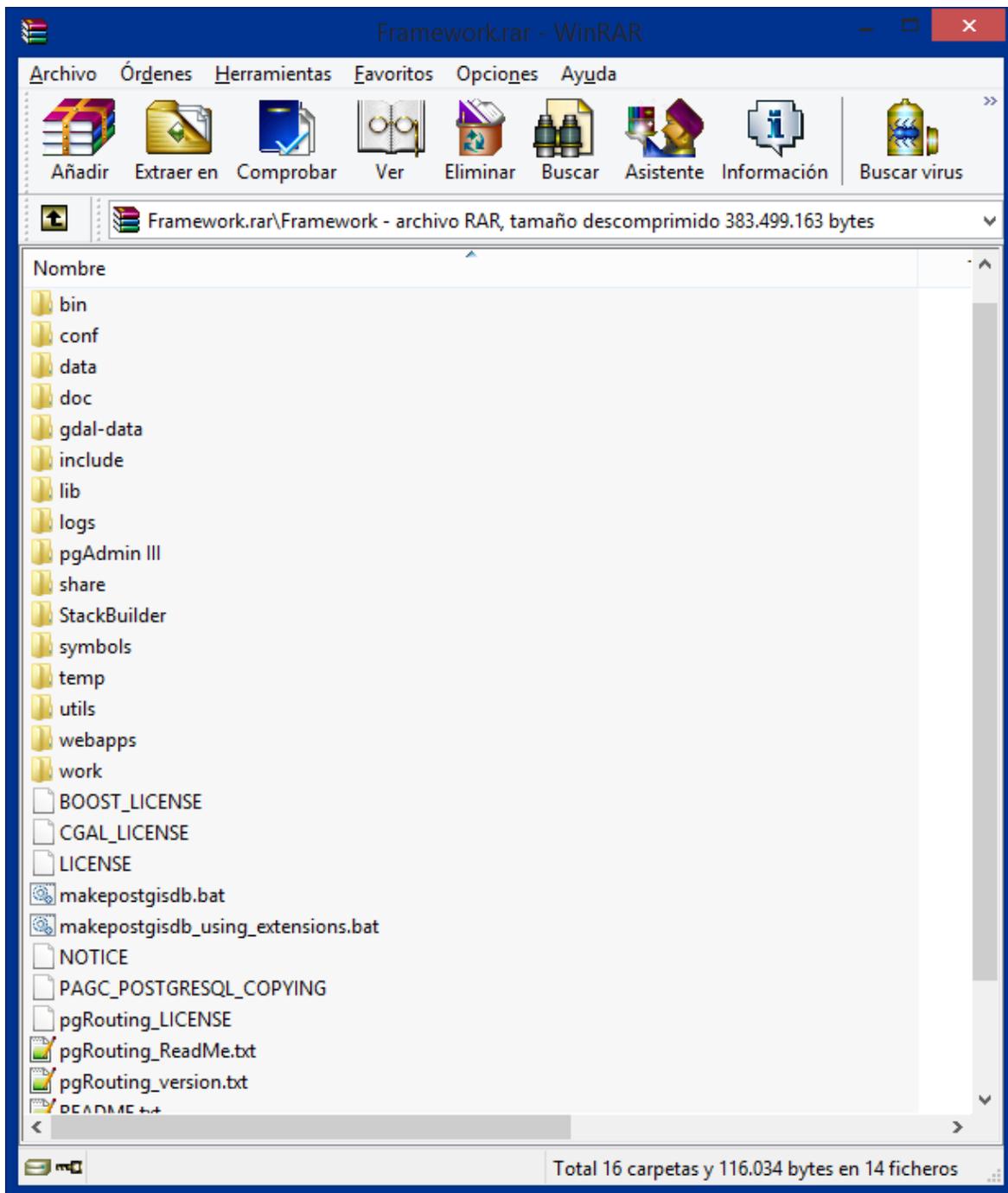


Figura 39. Archivos incluidos dentro del RAR del Framework

Para el levantamiento de los servicios requeridos por el *framework*, se creó un archivo batch `startup.bat`, en el cual únicamente al ejecutarlo se levantan los

servicios tanto de la BD Postgres con la extensión para Bases de Datos Espaciales Postgis, el servidor *web* Apache Tomcat y el Servidor de Mapas Geoserver; la ubicación del archivo se encuentra dentro de la carpeta bin como se muestra en la *Figura 40*.

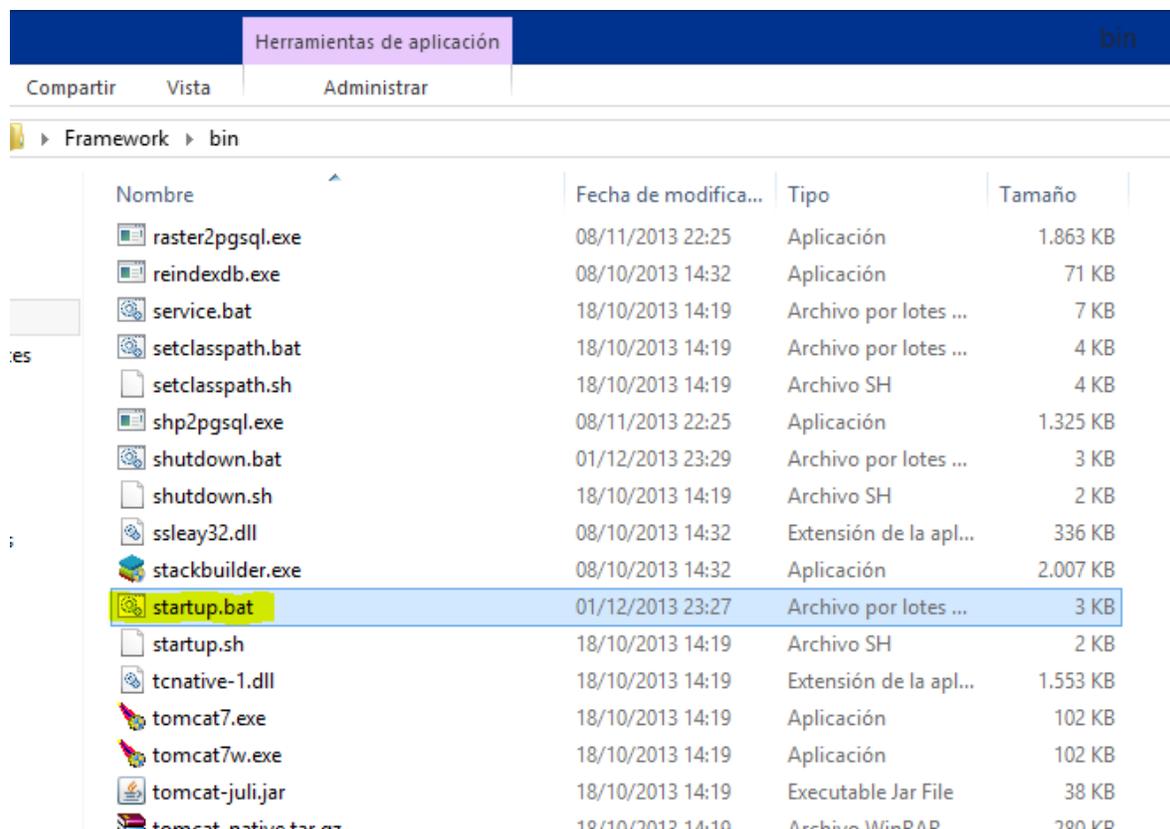
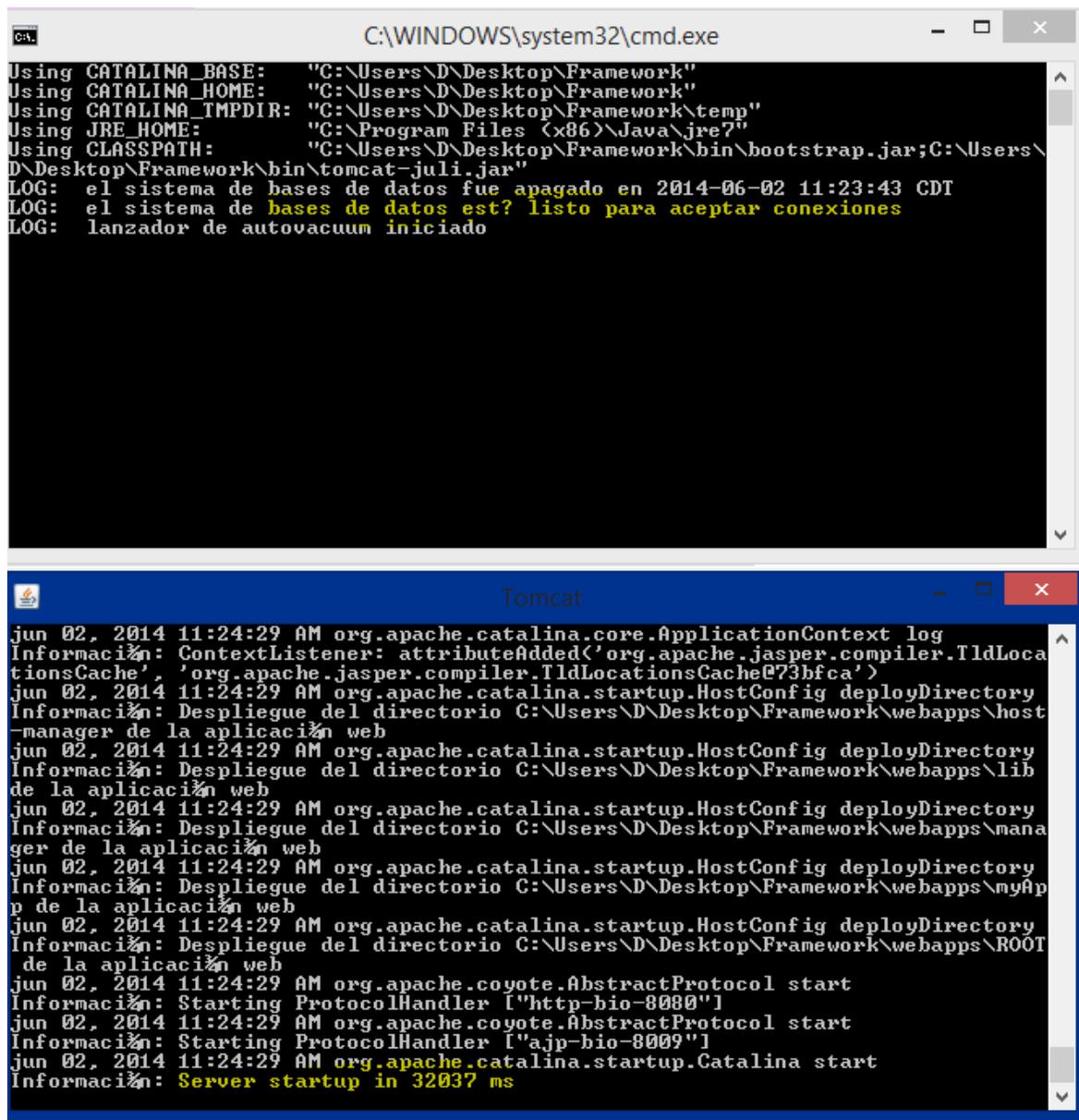


Figura 40. Ubicación del archivo startup.bat

Así mismo para detener el servicio basta con ejecutar el archivo shutdown.bat ubicado en la misma carpeta.

En la *Figura 41* se aprecia, una vez ejecutado startup.bat, el servicio de la base de datos postgres levantado así como el servidor web apache tomcat.



The image consists of two screenshots. The top screenshot shows a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The text in the window is as follows:

```
Using CATALINA_BASE: "C:\Users\D\Desktop\Framework"
Using CATALINA_HOME: "C:\Users\D\Desktop\Framework"
Using CATALINA_TMPDIR: "C:\Users\D\Desktop\Framework\temp"
Using JRE_HOME: "C:\Program Files (x86)\Java\jre7"
Using CLASSPATH: "C:\Users\D\Desktop\Framework\bin\bootstrap.jar;C:\Users\
D\Desktop\Framework\bin\tomcat-juli.jar"
LOG: el sistema de bases de datos fue apagado en 2014-06-02 11:23:43 CDT
LOG: el sistema de bases de datos est? listo para aceptar conexiones
LOG: lanzador de autovacuum iniciado
```

The bottom screenshot shows a Tomcat window titled 'Tomcat'. The text in the window is as follows:

```
jun 02, 2014 11:24:29 AM org.apache.catalina.core.ApplicationContext log
Informaci?n: ContextListener: attributeAdded('org.apache.jasper.compiler.TldLoca
tionsCache', 'org.apache.jasper.compiler.TldLocationsCache@73bfca')
jun 02, 2014 11:24:29 AM org.apache.catalina.startup.HostConfig deployDirectory
Informaci?n: Despliegue del directorio C:\Users\D\Desktop\Framework\webapps\host
-manager de la aplicaci?n web
jun 02, 2014 11:24:29 AM org.apache.catalina.startup.HostConfig deployDirectory
Informaci?n: Despliegue del directorio C:\Users\D\Desktop\Framework\webapps\lib
de la aplicaci?n web
jun 02, 2014 11:24:29 AM org.apache.catalina.startup.HostConfig deployDirectory
Informaci?n: Despliegue del directorio C:\Users\D\Desktop\Framework\webapps\mana
ger de la aplicaci?n web
jun 02, 2014 11:24:29 AM org.apache.catalina.startup.HostConfig deployDirectory
Informaci?n: Despliegue del directorio C:\Users\D\Desktop\Framework\webapps\myAp
p de la aplicaci?n web
jun 02, 2014 11:24:29 AM org.apache.catalina.startup.HostConfig deployDirectory
Informaci?n: Despliegue del directorio C:\Users\D\Desktop\Framework\webapps\ROOT
de la aplicaci?n web
jun 02, 2014 11:24:29 AM org.apache.coyote.AbstractProtocol start
Informaci?n: Starting ProtocolHandler ["http-bio-8080"]
jun 02, 2014 11:24:29 AM org.apache.coyote.AbstractProtocol start
Informaci?n: Starting ProtocolHandler ["ajp-bio-8009"]
jun 02, 2014 11:24:29 AM org.apache.catalina.startup.Catalina start
Informaci?n: Server startup in 32037 ms
```

Figura 41. Apache Tomcat y Postgres ejecutados