



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE CIENCIAS

Un sistema de caché para el despliegue  
de imágenes geográficas estáticas

TESIS

que para obtener el título de:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

presenta:

JOSAFAT ISAÍ GUERRERO IÑIGUEZ



Director de tesis:

L. EN C.C. KARLA RAMÍREZ PULIDO

2008



# Un sistema de caché para el despliegue de imágenes geográficas estáticas

---

1. Datos del alumno
Guerrero Iñiguez Josafat Isaí 55 79 69 86 Universidad Nacional Autónoma de México Facultad de Ciencias Ciencias de la Computación 96206234
2. Datos del tutor
L. en C.C. Karla Ramírez Pulido
3. Datos del sinodal 1
Dr. Jorge Prado Molina
4. Datos del sinodal 2
M. en C. José Antonio Quintero Pérez
5. Datos del sinodal 3
M. en I. María de Luz Gasca Soto
6. Datos del Sinodal 4
Dra. Elisa Viso Gurovich
7. Datos del trabajo escrito
Un sistema de caché para el despliegue de imágenes geográficas estáticas 136 p. 2008

A mi familia: Alicia, Julio, Azucena y Julio Jr.



# Agradecimientos

A mi familia que quiero tanto y nunca quitó el dedo del renglón.

A Bere, quien saltó conmigo durante todos estos años y ayudó a quitarme los pensamientos borrosos. ¡Poing!

A todos los amigos del laboratorio (Miguel, Miguel, Juan, Juan, René, René, Maye, Moy, ...) que me sufrieron, apoyaron y acompañaron durante la realización de ésta tesis, pero especialmente Ana, Iliá y Luis.

A mis colegas de generación Mary Carmen, Gustavo, Ruy, John, Pax y Grecia, por ser compañía, apoyo e inspiración para titularme. A Anayanzi, por interactuar conmigo y fomentar la sana competencia. A José Luis por sembrar la semilla platónica y conservar la amistad pese a la distancia.

A Elisa, cuyo coraje admiro.

A Pepe Q. por apoyar la realización de este trabajo.

A mi querida Lucy G. por creer siempre en mí, brindándome su apoyo incondicional.

A Karla, por todo el tiempo invertido, lágrimas derramadas y la amistad ganada.

A Jorge, quien la hizo de *cuchillito de palo* durante todos y cada uno de estos días, pero agradezco infinitamente.

Y por último, a todos aquellos que criticaron (Vero T.) o apoyaron este trabajo, aunque por descuido he olvidado colocar su nombre.

Mi deuda con ustedes es inmensa. ¡Gracias!

Josafat.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Sistemas de Información Geográfica . . . . .	2
1.2. Infraestructuras de datos espaciales . . . . .	3
1.3. Aplicaciones para la visualización de mapas . . . . .	5
1.4. Planteamiento del problema . . . . .	10
1.5. Contexto del trabajo . . . . .	11
<b>2. Sistemas de información geográfica e infraestructuras de datos espaciales</b>	<b>13</b>
2.1. Sistemas de Información Geográfica . . . . .	13
2.2. Infraestructuras de Datos Espaciales . . . . .	16
2.2.1. El Modelo de Referencia ISO para el Procesamiento Abierto y Distribu- tribuido. . . . .	18
2.2.2. Requisitos arquitectónicos . . . . .	20
2.2.3. Conceptos arquitectónicos de una IDE basada en estándares abiertos.	25
2.2.4. Componentes arquitectónicos de una IDE basados en estándares abiertos . . . . .	28
<b>3. Análisis y Diseño del sistema</b>	<b>33</b>

---

3.1. Mapeo en línea . . . . .	33
3.1.1. <i>OGC Web Map Service</i> . . . . .	34
3.2. Análisis del sistema . . . . .	42
3.3. Diseño del sistema . . . . .	48
3.4. Requerimientos del sistema . . . . .	51
3.5. Casos de uso . . . . .	52
<b>4. Desarrollo del sistema</b>	<b>55</b>
4.1. Tecnología . . . . .	55
4.2. Implementación del <i>Cascading Map Server</i> . . . . .	58
4.2.1. Documento <i>Capabilities</i> . . . . .	58
4.2.2. <i>GetCapabilities</i> : obtener información del servicio. . . . .	61
4.2.3. <i>GetMap</i> : obtener una imagen georeferenciada. . . . .	68
4.2.4. <i>GetFeatureInfo</i> : Obtener información sobre rasgos presentes en el mapa . . . . .	79
4.2.5. Manejo de errores . . . . .	81
<b>5. Evaluación del desempeño del sistema</b>	<b>85</b>
5.1. Contexto del proyecto . . . . .	88
5.2. Ambiente de evaluación . . . . .	88
5.3. Criterios de evaluación . . . . .	90
5.4. Diseño y planeación de las pruebas . . . . .	93
5.4.1. Obtención de métricas . . . . .	95
5.5. Configuración del ambiente de pruebas . . . . .	100
5.5.1. Implementación de las pruebas . . . . .	105
5.6. Análisis de resultados . . . . .	108

5.7. Conclusiones de la evaluación . . . . .	124
<b>6. Conclusiones</b>	<b>127</b>
<b>A. Ambiente físico y lógico.</b>	<b>131</b>
<b>Bibliografía</b>	<b>133</b>

# Capítulo 1

## Introducción

La información es la materia prima de muchas actividades humanas y su valor está relacionado directamente con su disponibilidad. En el caso de la información geográfica, su análisis y manipulación generan diversos productos indispensables como los mapas temáticos<sup>1</sup>, esenciales para la toma de decisiones en muchos aspectos de la vida diaria y en distintos niveles organizacionales. Estos productos son aprovechados en el ámbito de la educación, en la administración del ambiente, en la planeación de estrategias ante desastres naturales, en la agricultura, la salud, la planeación del transporte, la distribución de agua potable, la energía eléctrica, el uso de suelo, la seguridad pública, entre otros.

La adopción de Sistemas de Información Geográfica (SIG) a nivel mundial y la investigación en sistemas distribuidos, junto con el uso masivo de computadoras enlazadas en red y su conexión por medio de la red Internet, han llevado a los SIG a un desarrollo tal que es posible crear nuevas formas de almacenar, buscar, procesar y distribuir información de forma arbitraria desde lugares remotos. Toda esta tecnología aplicada al quehacer geográfico produce el desarrollo de herramientas para acceder a la información geográfica en línea por

---

<sup>1</sup>Es un mapa diseñado para representar información sobre un tema, como lo es la densidad de población o la geología.

medio de la utilización de componentes estandarizados, planteando nuevas posibilidades y beneficios para la toma de decisiones a distintos niveles, ya sean locales, regionales o globales.

Un ejemplo práctico de la utilidad del acceso global a estos acervos, se observa en el desarrollo de la Infraestructura de Datos Espaciales Canadiense (Canadian Geospatial Data Infrastructure o CGDI) [Inf05b], cuyo lema puede resumirse en: “Mejorando la calidad de vida a través de la toma de mejores decisiones”. Este tipo de iniciativas permiten resolver una gran cantidad de problemas sociales a través de la distribución relevante y oportuna de la información. Su alcance es de ámbito nacional y proporciona un servicio para acceder gran cantidad de datos a diversos tipos de usuarios, cada uno con objetivos y metas distintas, y por ende con requerimientos diferentes. Las características propias de este proyecto plantean problemas computacionales importantes debido a la gran cantidad de usuarios potenciales, al volumen de información por distribuir y al tamaño respectivo de la información. De manera concreta, el acceso, la manipulación y visualización de información geográfica en forma arbitraria y masiva compromete sistemas de cómputo y reduce su disponibilidad. Por tanto, la puesta en marcha de una Infraestructura de Datos Espaciales (IDE) requiere de una planeación y ejecución adecuada para poder asegurar el correcto funcionamiento a corto, mediano y largo plazo.

## **1.1. Sistemas de Información Geográfica**

Los Sistemas de Información Geográfica pueden verse como sistemas para capturar, almacenar, manipular, analizar y desplegar datos espacialmente referidos a la Tierra[Mas05].

La importancia de los SIGs radica en que éstos logran separar los datos o rasgos geográficos del análisis y de la representación visual a través del uso de abstracciones. De

esta forma, se puede acceder posteriormente a la información, modificarla y visualizarla de manera simple. Esto permite que fuentes de datos diferentes puedan mezclarse, analizarse y visualizarse en distintas proyecciones. Esta separación produce una abstracción respecto de la aplicación [Mas05] y por lo tanto eleva el valor de la información geográfica original ya que su utilidad sobrepasa los objetivos originales.

La importancia de los SIGs se observa de manera amplia en el ámbito gubernamental, porque ahí es donde se toman las decisiones clave para los intereses locales, regionales y nacionales. La toma de decisiones requiere contar con información geográfica, junto con mecanismos para administrarla, analizarla y visualizarla. Esta actividad se relaciona con el concepto de infraestructura, como las carreteras, los caminos o el servicio eléctrico. Una infraestructura debe ser un ambiente confiable que permita la comunicación e interconexión entre distintos elementos heterogéneos, basándose en conjuntos establecidos de protocolos y estándares.

## **1.2. Infraestructuras de datos espaciales**

La posibilidad de transmitir y publicar información geográfica por medio de Internet abre la posibilidad de compartir acervos geográficos albergados por distintos usuarios, dependencias, instituciones, etcétera, que se ubiquen distantes geográficamente.

El término Infraestructura de Datos Espaciales (IDE) denota una colección básica de tecnologías, políticas y acuerdos institucionales para facilitar la disponibilidad y el acceso a la información geoespacial. En este sentido, provee las bases para el descubrimiento, evaluación y acceso a la información espacial para usuarios en general. El empleo de la palabra infraestructura promueve el concepto de un ambiente confiable para facilitar el acceso a la información geográfica al seleccionar las prácticas a emplear en el sistema, los

protocolos y las especificaciones adecuados a la creación de dicho ambiente.[Neb04]

Desde el punto de vista arquitectónico, una IDE se compone de un conjunto de datos, servicios basados en estándares y protocolos, usuarios y aplicaciones[inf05a]. En la Figura 1.1 podemos observar que los distintos elementos de la IDE tienen una relación vertical entre sí, lo que corresponde al tipo de interacción esperada.



Figura 1.1: Relación entre los elementos de una IDE.

Una IDE considera un conjunto mínimo de usuarios, los cuales se pueden agrupar en las siguientes categorías [inf05a],[Mas05]:

- Proveedores de información.
- Administradores de la IDE
- Desarrolladores.
- Usuarios Finales.

La relación entre usuarios y servicios es estrecha, ya que los servicios que proporciona una IDE se sustentan en las necesidades básicas de los diversos usuarios y se caracterizan por los siguientes puntos[inf05a],[Neb04],[Mas05]:

- **Búsqueda de información relevante:** los usuarios de la IDE deben contar con mecanismos para explorar la información contenida.
- **Visualización:** conociendo el contenido de una IDE, la forma tradicional de presentar la información es por medio de mapas.
- **Evaluación:** después de visualizar el mapa, el usuario busca información concreta acerca de los datos consultados, como lo son el estado actual de la información, la fecha de proceso y el linaje de la información<sup>2</sup>.
- **Acceso a la información:** una vez establecida la utilidad de la información, es necesario obtener los datos actuales.

La conexión del usuario final con los servicios propuestos por una IDE se realiza a través de las aplicaciones desarrolladas para consumir los servicios propuestos por la IDE. Es entonces cuando podemos agruparlas en aplicaciones para visualización, descubrimiento, publicación y edición de la información.

### **1.3. Aplicaciones para la visualización de mapas**

Históricamente, la información geográfica se ha presentado por medio de mapas, lo cual permite al usuario final evaluar de manera rápida si la información es adecuada a sus necesidades particulares.

---

<sup>2</sup>El conjunto de procesos y modificaciones que ha sufrido la información desde su creación digital.

La visualización de mapas en línea<sup>3</sup> es una puerta de entrada a la información geográfica, ya que un solo vistazo a un mapa permite explorar la gran cantidad de información disponible sin necesidad de descargar el conjunto completo de datos que lo compone, el cual usualmente es de gran tamaño y puede contener información irrelevante a las necesidades particulares del usuario.

Por esta razón, es importante que las Infraestructuras de Datos Espaciales permitan y habiliten el acceso a la información geográfica por medio de servicios y aplicaciones para la visualización de mapas, accediendo así a grandes volúmenes de información de manera inmediata y oportuna.

La *Open Geospatial Consortium (OGC)* es una organización internacional sin fines de lucro que dirige el desarrollo de estándares para la creación de servicios Web geoespaciales y de geolocalización. Los productos de la *OGC* son los estándares *OpenGIS* [opea], los cuales son documentos técnicos detallados respecto a interfaces para la creación de servicios Web, cuyo objetivo primario es lograr la interoperabilidad entre distintos proveedores de datos geográficos. Las especificaciones no describen implementación alguna, pero contienen gran cantidad de detalles que permiten lograr la interoperabilidad.

El presente documento se concentra en una de estas especificaciones cuya funcionalidad es crear y desplegar mapas a partir de imágenes superpuestas provenientes de diferentes fuentes. El estándar se llama *Web Map Service (WMS)* [Inc02] y define las siguientes operaciones:

- *GetMap*: expone la forma de hacer peticiones de imágenes así como el mecanismo de acceso a las mismas.
- *GetFeatureInfo*: permite obtener información concreta de un rasgo en el mapa, obte-

---

<sup>3</sup>A través de Internet.

niendo los atributos que describen dicho rasgo.

- *GetCapabilities*: proporciona información acerca de los tipos de mapas que un servidor *WMS* puede producir, incluyendo extensión geográfica, los formatos de imágenes, las proyecciones permitidas para el servicio, entre otras.

Algunos ejemplos de aplicaciones que pueden consumir este tipo de servicios en línea son:

- *NASA WorldWind*[wor]: proporciona una interfaz virtual del planeta Tierra donde se despliegan imágenes satelitales y datos de elevación provenientes del *SRTM*<sup>4</sup>.
- *OpenLayers*[opeb]: es un conjunto de bibliotecas *Open Source* para la creación de aplicaciones Web, las cuales permiten el despliegue de mapas dentro de navegadores Web empleando los estándares *OGC OpenGIS* .

Estas aplicaciones permiten presentar información geográfica diversa, en algunos casos vectorial y en otros casos en formato de imágenes o rasters por medio de mapas; están pensadas para ser usadas por personas de todo el mundo y requieren transferir volúmenes moderados de información por medio de la red.

La utilidad de estas aplicaciones se observa comparando la cantidad de información necesaria para transferir el conjunto completo de datos con la cantidad de tiempo requerida para transferir la imagen de un mapa. Un ejemplo de esto se presenta al crear servicios de imágenes *WMS* para desplegar ortofotos<sup>5</sup> que cubren México. Suponiendo una resolución

---

<sup>4</sup>Siglas de Shuttle Radar Topography Mission, que significa Misión Topográfica de Radar

<sup>5</sup>Fotografía aérea en la cual las distorsiones ocasionadas por la inclinación de la cámara y el relieve del terreno se eliminan. Una ortofoto conserva la escala a través de toda la fotografía y puede emplearse como mapa[gis06].

de 3 metros por pixel para cada ortofoto y empleando una retícula escala 1:20,000, se requiere almacenar aproximadamente 55 GB de datos <sup>6</sup> para todo el conjunto.

Estas aplicaciones emplean esquemas optimizados de consumo y presentación de información, los cuales solicitan únicamente la información requerida y en escalas predefinidas. La práctica común es segmentar la tierra con una malla regular<sup>7</sup> a una escala fija donde cada celda está indexada. Si se emplea una escala mayor<sup>8</sup> se requiere un mayor número de celdas y, a la inversa, si se emplea una escala menor se requiere un número menor de celdas.

En general, cuando se requiere desplegar el mapa correspondiente a un área cualquiera de la tierra, se emplea solamente el conjunto de imágenes correspondientes al área de interés empleando una escala adecuada y se obtienen únicamente las celdas que se requieran visualizar. Estos conjuntos de imágenes a diferentes escalas pueden verse como pirámides de imágenes, donde cada nivel contiene las mismas imágenes pero a distintas escalas.

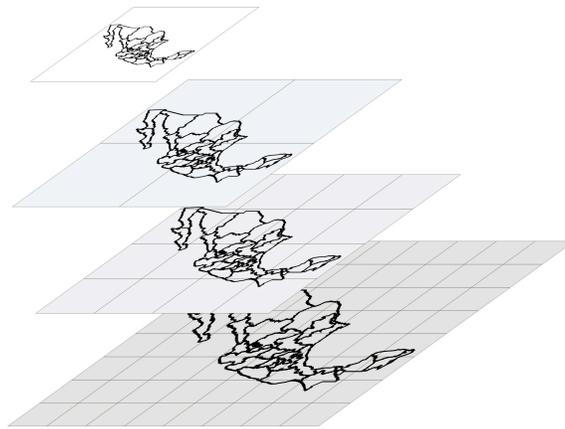


Figura 1.2: Pirámide de mosaicos.

<sup>6</sup>Aproximadamente 4MB por ortofoto. Hay más de 13,000 ortofotos cubriendo solamente el área continental.

<sup>7</sup>Una retícula cuyos mosaicos sean paralelos tanto a los meridianos como a los paralelos de la Tierra.

<sup>8</sup>Una menor área visible.

Aunque esta ingeniosa solución disminuye la cantidad de información que se debe transferir a través de la red, aparecen nuevos problemas al aumentar la cantidad de usuarios y, por lo tanto, el número de peticiones que debe atender un servidor de imágenes georeferenciadas.

La generación de imágenes crea problemas de sobrecarga en servidores de imágenes, porque cada petición que se recibe y procesa genera un acceso a la fuente de datos, la cual debe procesarse para generar una representación visual de la información geográfica. La complejidad asociada a la generación de tales imágenes varía dependiendo de la cantidad de información involucrada, así como del número de operaciones geométricas necesarias, la complejidad de la consulta, las opciones de despliegue del mapa, entre otras operaciones.

El esquema de consumo basado en pirámides de mosaicos posee la ventaja de emplear retículas fijas que pueden aprovecharse para reducir la carga asociada de los sistemas que proveen los mapas: si la herramienta únicamente solicita imágenes cuya extensión corresponde a los mosaicos de una retícula constante, entonces cada imagen puede almacenarse para algún uso posterior.

Una solución ingenua conduce a la generación de todas las posibles imágenes para una retícula concreta, reduciendo el tiempo de procesamiento necesario para atender cada petición y por consiguiente el tiempo de respuesta del sistema. Sin embargo, esta solución no considera la cantidad excesiva de espacio requerido para almacenar el conjunto total de imágenes, lo que nos lleva a buscar soluciones equilibradas acorde a las posibilidades de espacio disponible, así como a las capacidades de cómputo actuales.

## 1.4. Planteamiento del problema

El problema se describe en los siguientes términos:

- Existe una sobrecarga de trabajo en los servidores de imágenes basados en el estándar *WMS* debido a la complejidad inherente a la generación de imágenes georeferidas.
- La sobrecarga de trabajo se refleja en un mayor tiempo de respuesta para los clientes y por consiguiente en una menor utilidad de las aplicaciones para visualizar mapas.
- El empleo de soluciones ingenuas para producir todas las imágenes no conduce a soluciones útiles debido a que la cantidad de espacio necesaria para generar tales conjuntos es enorme y la cantidad de recursos necesarios no siempre está disponible.
- El número de usuarios potenciales de las aplicaciones para visualizar mapas aumenta a una velocidad mayor que las capacidades de cómputo disponibles.
- La baja disponibilidad de los servidores *WMS* resulta en un limitado acceso a la información en línea, razón por la cual debe encontrarse un mecanismo equilibrado que permita acceder a la información contenida en forma de mapas, aumente la disponibilidad de los servidores *WMS* y disminuya el tiempo de respuesta.

La creación de mecanismos para acelerar el acceso a mapas en línea es de reciente aparición y su desarrollo fomenta la socialización (uso público) de la información geográfica. Aún no se han desarrollado soluciones estándares que permitan disminuir la sobrecarga de los servidores de imágenes, razón por lo cual, deben crearse soluciones *ad – hoc* para satisfacer las necesidades de cada aplicación para la visualización de mapas en línea.

Este documento está dirigido a quienes planeen distribuir información geográfica estática a gran escala en forma de imágenes por medio del estándar *WMS*, así como a los desarrolladores de aplicaciones para la visualización de mapas.

El objetivo de la tesis es plantear y desarrollar formalmente una solución al problema de la sobrecarga de trabajo del procesador en caso de tratarse de peticiones de imágenes estáticas, a escalas fijas y correspondientes a una retícula.

## 1.5. Contexto del trabajo

El presente trabajo nace dentro del proyecto SIBA (Sistema de Informática para la Biodiversidad y el Ambiente), el cual es uno de los cinco megaproyectos incluidos dentro del programa IMPULSA. Dentro de SIBA se desarrollan dos sistemas: la Unidad de Informática para la Biodiversidad (UNIBIO) y la Unidad de Informática Geoespacial (UNIGEO).

La UNIGEO tiene dentro de sus objetivos desarrollar una Infraestructura de Datos Espaciales capaz de alojar y distribuir información geoespacial a través de Internet. Para cumplir estos objetivos fue necesario evaluar y adoptar diversas tecnologías ya existentes, y en ausencia de herramientas adecuadas, crear sistemas de software que complementen al conjunto de tecnologías empleadas.

El presente trabajo forma parte del conjunto de herramientas desarrolladas que permiten la creación de aplicaciones espaciales de alto desempeño, ya que permiten diseminar información geoespacial de manera eficiente, rápida y empleando recursos limitados.

# Capítulo 2

## Sistemas de información geográfica e infraestructuras de datos espaciales

La Geografía tiene por objetivo describir y analizar hechos y fenómenos que ocurren en el espacio geográfico. Estos objetivos se pueden cumplir con el planteamiento de marcos de trabajo adecuados y el uso de un lenguaje específico que permita transmitir el conocimiento acerca de los fenómenos terrestres.

### 2.1. Sistemas de Información Geográfica

La creación de los Sistemas de Información Geográfica (SIG) es el mayor paso realizado desde la invención del mapa rumbo a un manejo adecuado de la información geográfica [Mas05].

“Un Sistema de Información Geográfica es un software que permite capturar, integrar, verificar, administrar, analizar y visualizar información referida geospacialmente.” Esta definición engloba un amplio conjunto de actividades que un SIG puede realizar, pero también

permite ampliar el conjunto de información que se considera como información geográfica, por ejemplo, cualquier dato que pueda relacionarse con una ubicación geográfica por medio de una coordenada o algún registro postal, área administrativa o región, por nombrar algunos [Mas05].

Las tareas que realiza un SIG se logran gracias a la creación y empleo de abstracciones sobre los datos, acerca de su representación, los mecanismos de almacenamiento, las herramientas de visualización, el procesamiento de la información, la capacidad de búsqueda y la valoración de la información geográfica [wha07]. Estas abstracciones conforman los SIG y se describen brevemente en los siguientes puntos:

- **Representación de los datos:** los datos geográficos corresponden a observaciones, medidas, rasgos, descripciones, imágenes, tablas de datos, reportes, etc, que describen entidades geográficas. Los datos se representan y organizan por medio de modelos de datos geográficos que son descripciones matemáticas de las entidades geográficas. El paradigma matemático para la representación de objetos geográficos corresponde a conjuntos de puntos, líneas y polígonos. En el caso de la representación de superficies se emplea el concepto de *raster*, el cual es una una matriz con valores, donde cada celda representa algún rasgo de una superficie. Estos modelos motivan la creación de archivos de computadora y bases de datos geográficas para almacenar la información. Cuando se habla de bases de datos geográficas, se habla de modelos de datos a un nivel superior <sup>1</sup> que describen las capas temáticas que componen una aplicación, permiten generar reglas semánticas y crear relaciones entre la información geográfica, dependiendo de la aplicación que se pretenda desarrollar.
  
- **Mapas:** un mapa dentro de un SIG es la representación cartográfica de la información

---

<sup>1</sup>Se dice que un modelo de datos es superior a otro si permite modelar un mayor número de fenómenos o cualidades.

geográfica. La visualización de mapas en la computadora requiere del empleo de múltiples disciplinas computacionales como la geometría computacional, el diseño y análisis de algoritmos, bases de datos, visualización por computadora, entre otras. Todas ellas son necesarias para poder llevar a cabo el despliegue de la información de manera rápida y eficiente.

- Modelos de geoprocésamiento y rutinas de automatización: el tratamiento de la información geográfica usualmente requiere realizar operaciones simples de manera repetida. Un SIG debe proporcionar herramientas que permitan automatizar las tareas rutinarias para optimizar el trabajo.
- Métodos y flujos de trabajo: el manejo de información geográfica requiere del empleo de metodologías adecuadas para la generación de mapas, la edición de información, el análisis, visualización y administración. Sólo el uso de procedimientos adecuados permite el uso correcto de los SIGs.
- Metadatos: son la llave para el descubrimiento de la información dentro de un SIG puesto que éstos documentan su contenido. Al organizar y exponer los metadatos por medio de catálogos, se permite a los usuarios evaluar la utilidad de los datos, es decir, conocer el estado de la información, la calidad de la misma, los procesos realizados con ella, la escala, su resolución y el origen de la información.

El impacto de los SIG en la producción y la utilización de la información geográfica es grande ya que éstos separan los datos de su representación en el mapa. Esta separación permite que la información empleada al generar el mapa pueda reutilizarse para generar productos diferentes a los originalmente planeados [Mas05], incrementando su valor y haciéndola independiente del contexto de su creación.

## Representación digital de la información geográfica

La información geográfica se representa usualmente por medio de alguno de los siguientes tipos de datos [Zei99].

- **Vectores:** son un modelo que representa los rasgos de manera precisa y de forma compacta como un conjunto ordenado de coordenadas. Esta representación permite realizar operaciones geométricas, como calcular longitudes y áreas, identificar traslapes e intersecciones entre rasgos o encontrar rasgos cercanos, entre otras operaciones posibles. Se emplean puntos para representar rasgos unidimensionales en el mapa u objetos que no tienen longitud o área. Las líneas representan objetos geográficos angostos que no requieren ser representados por medio de un área, pero cuya longitud es importante. Los polígonos se emplean para representar objetos geográficos que requieren considerar su área.
- **Imágenes *raster*:** son retículas bidimensionales que contienen celdas, las cuales poseen un valor que representa algún atributo. Algunos ejemplos de imágenes raster son las imágenes satelitales, imágenes aéreas y modelos digitales de terreno, por citar algunos.
- **Tablas:** es información asociada con los rasgos y objetos geográficos. Esta información puede relacionarse a su vez con otras tablas y rasgos.

### 2.2. Infraestructuras de Datos Espaciales

El acceso a la información geográfica es un factor clave para cualquier actor involucrado en la toma de decisiones tanto a nivel local, regional o global. Entre estos actores podemos

mencionar a las agencias de seguridad pública interesadas en el control del crimen, consultoras especializadas en el desarrollo de negocios, agencias de gobierno encargadas de la prevención y respuesta ante desastres naturales como inundaciones o terremotos, ante desastres antrópicos como la deforestación o para atender los problemas de la administración del territorio, como es el caso de la asignación del uso de suelo potencial.

La información geográfica es un recurso cuyo valor se relaciona directamente con su costo de adquisición, lo cual conduce a su baja disponibilidad. Debido a este factor es común que distintas organizaciones dedicadas a la generación de productos geográficos posean conjuntos de datos particulares pertinentes a sus actividades, provenientes de distintas fuentes, con un procesamiento, contenido y calidad distinta. Esto conlleva a la adopción de soluciones particulares para la organización y distribución de la información geoespacial, limitando su reutilización.

La tecnología moderna proporciona herramientas para la adquisición, distribución y utilización de información geográfica a través de redes de computadoras. La posibilidad de proporcionar acceso remoto facilita la creación de productos nuevos, al tiempo que reduce su costo de adquisición. Al ser reutilizada la información, aumenta su valor, reduce la duplicidad y disminuyen los costos asociados con su administración.

El término Infraestructura de Datos Espaciales (IDE) denota un conjunto básico de tecnologías, políticas y acuerdos institucionales creados para facilitar la disponibilidad y acceso a la información espacial. Una IDE debe proporcionar los mecanismos básicos que permitan tanto el descubrimiento de la información, como realizar una evaluación de su contenido, así como brindar los medios para disponer de tales acervos. La información geográfica es un bien que debe ser administrado cuidadosamente para alcanzar su valor potencial y en este sentido, la administración de los bienes se encuentra estrechamente vinculada con la idea de infraestructura.

El empleo de la palabra *infraestructura* promueve la adopción de un ambiente confiable para facilitar el acceso a la información geográfica a través del establecimiento y adopción de un conjunto mínimo de prácticas, protocolos y especificaciones [Neb04], facilitando la creación de productos y aplicaciones geoespaciales.

La adopción masiva de computadoras, junto con el incremento del poder de cómputo y el aumento de las capacidades de comunicación, promueven el desarrollo de infraestructuras propicias para el desarrollo de aplicaciones distribuidas en sistemas abiertos [Mor01]. En particular, el uso de Internet como medio de comunicación tiene una influencia importante sobre el diseño de la arquitectura de las IDE, debido a que estas emplean mecanismos para acceder a recursos remotos por medio de redes de computadoras. El resultado de interconectar IDE geográficamente remotas para compartir y acceder tanto a la información como a los recursos informáticos produce lo que se conoce como Infraestructuras de Datos Espaciales Distribuidas. Una IDE distribuida permite que sistemas independientes puedan comunicarse y colaborar entre ellos.

### **2.2.1. El Modelo de Referencia ISO para el Procesamiento Abierto y Distribuido.**

Los sistemas distribuidos poseen características específicas que la Ingeniería de Software tradicional no ha podido afrontar correctamente ni tampoco de forma completa [Mor01].

La *International Standards Organization (ISO)* y la *International Telecommunication Union (ITU-T)* plantean el Modelo de Referencia de Procesamiento Abierto y Distribuido o *Reference Model for Open Distributed Processing (RM-ODP)*, el cual "... proporciona un marco de coordinación para la normalización del desarrollo de aplicaciones abiertas y distribuidas, creando una arquitectura capaz de soportar de forma integrada aspectos tales

como la distribución, interoperabilidad o portabilidad de los sistemas, objetos y componentes” [Mor01].

Este modelo proporciona las bases conceptuales y también los patrones o reglas de implementación para la creación de sistemas distribuidos. El modelo de referencia está definido por un conjunto de normas básicas que especifican su estructura y elementos básicos.

De estas normas se extrae una serie de conceptos fundamentales acerca del modelo para la normalización de sistemas abiertos y distribuidos. Estos conceptos sintetizan los elementos requeridos para la creación de Sistemas Abiertos y Distribuidos y se describen en los siguientes puntos [Mor01]:

- El sistema se debe describir a partir de especificaciones basadas en distintos puntos de vista interrelacionados.
- Debe emplearse un modelo de objetos común a cualquier punto de vista.
- Debe definirse una infraestructura que elimine y oculte la complejidad inherente a los sistemas distribuidos, simplificando la especificación y diseño de aplicaciones.
- Estos sistemas deben especificarse y definirse a partir de funciones comunes de utilidad general.
- Los sistemas deben contar con mecanismos para auto evaluar la conformidad con las normas de un sistema OPD.

Los puntos de vista definidos por el modelo permiten atender diferentes aspectos de los sistemas distribuidos y facilitan la división de responsabilidades. Este modelo muestra que estos sistemas son complejos y no pueden ser descritos por una sola representación. Para conseguir resultados satisfactorios es necesario considerar distintos puntos de vista [Mor01]:

- Punto de vista empresarial o de negocios, define la motivación, el alcance, los objetivos perseguidos, responsabilidades, políticas y casos de uso.
- El punto de vista informático hace énfasis en la información contenida por una IDE, así como en el comportamiento que ésta tiene. Este comportamiento se debe entender a partir del conjunto de modelos de datos empleados, semánticas y esquemas adoptados.
- Computacionalmente, se hace énfasis en los componentes, interfaces, interacciones y restricciones, independientemente de los actores.
- La visión ingenieril describe la infraestructura y los mecanismos para la distribución de sus componentes, así como sus restricciones, interacciones y relaciones.
- Tecnológicamente, se definen los ambientes de implementación y desarrollo usando tecnología estándar y productos adecuados.

### **2.2.2. Requisitos arquitectónicos**

Toda Infraestructura de Datos Espaciales tiene como objetivo primario maximizar el uso de la información geográfica a partir del acceso inmediato a los acervos de información mantenidos por las distintas organizaciones relacionadas con ella. Esta diversidad de participantes implica coordinar distintas organizaciones para realizar correctamente el potencial de una IDE. Una IDE también involucra un conjunto amplio de actividades relacionadas con el manejo de información, tecnologías, estándares y mecanismos de distribución. Además, debe contemplar asuntos institucionales relacionados con la asignación de responsabilidades, recursos económicos y humanos. Por otra parte, los usuarios son los

que determinan gran parte de la funcionalidad de una IDE, ya que ellos son los principales consumidores de la información geográfica [Mas05].

El diseño de una Infraestructura de Datos Espaciales se guía por la necesidad de proporcionar acceso en línea a la información geoespacial, conduciendo al desarrollo de componentes de software, servicios y conjuntos de datos que permitan reducir la dificultad, el costo y el tiempo de desarrollo de aplicaciones gracias a la creación de una red de clientes y proveedores tanto de información como de servicios y aplicaciones.

Para diseñar Infraestructuras de Datos Espaciales es necesario:

- Proporcionar acceso en línea a los conjuntos de información geográfica y a los servicios implementados.
- Permitir la integración de información geográficamente dispersa.
- Favorecer el intercambio de información y la coordinación entre partes.
- Descentralizar las actividades de desarrollo, permitiendo que cada integrante de una IDE realice sus objetivos particulares e interactúe de forma independiente.
- Facilitar la generación y distribución de documentación acerca de la información geoespacial.

El correcto cumplimiento de estos requisitos está condicionado a las capacidades de interoperabilidad<sup>2</sup> de la IDE, ya que ésta es la que permite a los usuarios encontrar información, servicios y aplicaciones independientemente de su ubicación física, ayudando a comprender y emplear tanto las herramientas como la información independientemente de la plataforma empleada, asegurando con esto la generación de ambientes de desarrollo libres de restricciones propietarias.

---

<sup>2</sup>Habilidad de un sistema o de sus componentes para acceder a recursos heterogéneos por medio de interfaces previamente definidas.

Desde el punto de vista de interoperabilidad entre sistemas, hay que considerar ciertos aspectos para el diseño de una arquitectura[inf05a]:

- Uso de protocolos de red que permitan la comunicación básica entre componentes.
- Empleo de interfaces comunes.
- Protocolos de transporte de datos.
- Interoperabilidad semántica.

La interoperabilidad entre sistemas es uno de los requisitos para el correcto desarrollo de una Infraestructura de Datos Espaciales Distribuida y el empleo de interfaces estándares y especificaciones precisas es necesaria para cumplir con los objetivos de la misma. La estandarización requiere del consenso y la aceptación de una solución técnica por parte de las personas que trabajan en el desarrollo y dirección de una cierta tecnología. El proceso de estandarización requiere del uso y revisión constante de trabajos previos para refinar y mejorar los estándares y especificaciones establecidos [inf05a].

El desarrollo de las Infraestructuras de Datos Espaciales basadas en estándares abiertos está ligado estrechamente a dos organizaciones, la primera es el *Open Geospatial Consortium (OGC)*, encargada de definir especificaciones para la implementación de soluciones geoespaciales, mientras que la segunda es la *International Organization for Standardization(ISO)*, encargada de definir estándares abstractos [inf05a].

La *OGC* es un comité técnico encargado de la creación de especificaciones para interfaces de red interoperables, está conformado por la industria, la academia y el gobierno. El comité tiene por intención promover y desarrollar estándares abiertos para la creación de soluciones geoespaciales distribuidas. Estas especificaciones proveen de herramientas para

el acceso a información geoespacial a los usuarios de las IDEs, así como una colección de Servicios Web geoespaciales para los desarrolladores de sistemas.

La ISO es una organización encargada de definir y publicar estándares internacionales y poseen un apartado importante de especificaciones relativas al desarrollo de IDEs, entre las que destacan esquemas para la descripción de metadatos<sup>3</sup> y para la codificación de rasgos geográficos<sup>4</sup>.

El uso de Internet abre la posibilidad de realizar comunicaciones entre sistemas de cómputo remotos, pero debido a que esta red se compone de un gran número de sistemas heterogéneos, los cuales pueden emplear distintos protocolos de comunicación, protocolos de red, mecanismos de seguridad, codificaciones, entre otros, es difícil tener estándares en la representación de la información, razón por la cual la estandarización es un asunto importante para lograr una colaboración efectiva.

Los servicios Web surgen como una solución para los problemas de interoperabilidad, ya que su empleo en el diseño e implementación de una IDE permite resolver las necesidades de interoperabilidad, y al mismo tiempo proporcionan una metodología o filosofía para la creación de aplicaciones y servicios desacoplados de la implementación.

Un servicio Web se define informalmente como un sistema de software diseñado para permitir la interacción entre máquinas a través de una red; puede verse como un sitio en la red concebido para ser consumido por un programa en lugar de por personas [Gro04]. El intercambio de mensajes y operaciones se realiza por medio de protocolos y especificaciones preestablecidas que puedan ser procesadas por una máquina.

El empleo de una arquitectura basada en Servicios Web proporciona un ambiente distribuido donde los servicios son utilizables por medio de protocolos, estándares e interfaces

---

<sup>3</sup>ISO 19115: Geographic Information - Metadata

<sup>4</sup>ISO 19110: Geographic Information - Methodology for feature cataloguing.

basadas en especificaciones concretas.

### ***OGC Basic service model***

La *OGC* proporciona un Modelo Básico de Servicios Web (*OGC Basic Service Model* [inf05a]) que proporciona una plataforma para el desarrollo de servicios Web que se resume en los siguientes puntos:

- Un servicio se debe de llamar a través de un *URL*<sup>5</sup> accesible por medio de un servidor Web.
- Los servicios deben de ser solicitados a través de una petición *HTTP*<sup>6</sup> empleando el método *get* o *post*.
- Cuando un servicio se invoca a través del método *get*, los parámetros se especifican usando las convenciones *CGI*<sup>7</sup>.
- Cuando se llama a un servicio por medio del método *post*, los parámetros se codifican en *XML*.
- El servicio debe poseer una operación *getCapabilities* que devuelva información en forma de metadato codificado en *XML*.

La propuesta de la *OGC* para la creación de Servicios Web difiere de la propuesta del *World Wide Web Consortium (W3C)*, debido a que esta última está basada en otros estándares como *SOAP*, *Web Services Description Language (WSDL)* y *Universal Description, Discovery and Integration (UDDI)*, entre otros [Tu06]. Los Servicios Web propuestos

---

<sup>5</sup>Siglas de *Uniform Resource Locator* y se traduce como Localizador Uniforme de Recursos

<sup>6</sup>Siglas de *Hyper Text Transfer Protocol*: Protocolo de Transferencia de HiperTexto.

<sup>7</sup>Siglas de *Common Gateway Interface*: Interfaz Común de Entrada.

por la *OGC* son servicios especializados para el ámbito geográfico que poseen mecanismos para el descubrimiento, transporte y codificación de información.

Dentro de una IDE, la información geográfica y los Servicios Web poseen la misma importancia, razón por la cual ambos cuentan con mecanismos de descubrimiento, al contrario de los Servicios Web tradicionales que enfatizan los servicios y solamente poseen mecanismos de descubrimiento [inf05a].

### **2.2.3. Conceptos arquitectónicos de una IDE basada en estándares abiertos.**

Una Infraestructura de Datos Espaciales se forma a partir de la combinación de elementos que permiten describir la información, realizar búsquedas en los acervos, brindar acceso a los datos y proporcionar elementos para evaluar la calidad y utilidad de la información.

Los elementos que componen una IDE son los siguientes [inf05a]:

- **Datos:** la información es la parte fundamental de toda IDE ya que ésta permite generar nuevos productos geográficos a partir de la mezcla de la información disponible, produciendo así resultados acordes a las necesidades de cada usuario.

Una IDE puede contener conjuntos de datos geográficos base relevantes, así como conjuntos temáticos acerca de temas especializados como la precipitación, la geología o la población.

En términos de información base, es deseable tener conjuntos de datos continuos y completamente integrados para así agrupar la información en alguna de las tres categorías siguientes [inf05a]:

- **Capas de alineación:** corresponden a todos los elementos de control geométrico

empleados para posicionar correctamente la información. Aunque no representan rasgos particulares, son críticas para el empleo del resto de la información. Ejemplos de estos datos son las retículas de diferentes tamaños, trazas de satélites, entre otras.

- Rasgos físicos: corresponden a rasgos naturales o rasgos creados por el hombre, y en ambos casos no están sujetos a interpretación o especulación. Ejemplos de éstos son caminos, ríos, curvas de nivel, datos de elevación, etcétera.
  - Capas conceptuales: corresponden a información descriptiva y administrativa acerca de un país. Pueden corresponder a observaciones físicas, económicas o también a factores sociales. Ejemplos de estas capas son los límites administrativos, distritos electorales o áreas ecológicas.
- Servicios: los servicios Web emergen como una tecnología clave para el intercambio de información en Internet y su uso facilita el acceso y posibilita la distribución de información geoespacial. Los servicios Web se pueden entender como aplicaciones que permiten realizar un conjunto específico de tareas; éstos se construyen a partir de interfaces definidas en XML, lo que permite interoperabilidad; la funcionalidad y la información se separan de la interfaz del usuario.

El empleo de servicios Web permite que sistemas independientes puedan comunicarse y colaborar entre sí, pero para garantizar su éxito se requiere la adopción de estándares y especificaciones abiertas.

Los servicios Web son los bloques de construcción de las aplicaciones, y cada aplicación puede emplear uno o más servicios Web por cada acción a realizar.

- Usuarios: son los participantes dentro de las actividades de una IDE y se agrupan en

cuatro categorías:

- Proveedores de información geográfica y de servicios Web. Estos usuarios trabajan en la parte central de una IDE, proporcionando información base para el desarrollo de aplicaciones y para los servicios Web.
  - Desarrolladores: están encargados de crear aplicaciones que permitan la interacción de los usuarios con la información geoespacial. Las aplicaciones desarrolladas deben de emplear los servicios Web propuestos por la IDE.
  - Promotores: son los encargados de distribuir y promover el uso de las aplicaciones geoespaciales desarrolladas dentro de la IDE.
  - Usuarios finales: son los principales consumidores de la información geoespacial y poseen objetivos particulares. Aquellos dependen básicamente de las aplicaciones desarrolladas para acceder a los recursos albergados por una IDE.
- Aplicaciones: el desarrollo de herramientas o aplicaciones geoespaciales se guía por las necesidades de los usuarios de una infraestructura de datos. Las aplicaciones geoespaciales consumen servicios web para acceder a la información georeferida espacialmente o para procesar información geoespacial. Dentro del contexto de la IDE se distinguen los siguientes tipos de aplicaciones, aunque la IDE no se limita a utilizar alguna de las siguientes categorías [inf05a]:
- Visualizadores: muestran gráficamente información proveniente de servidores que producen mapas.
  - Herramientas de descubrimiento: son aplicaciones que permiten indagar acerca del contenido geoespacial ubicado dentro de un servidor y proporcionan los mecanismos para acceder a la IDE.

- Herramientas para la publicación: herramientas creadas para facilitar la publicación de la información disponible. Son necesarias para comenzar a compartir y distribuir información.
- Administradores: son herramientas que permiten agregar, eliminar o actualizar la información geográfica disponible en línea. Se encargan de mantener gran parte de la coherencia entre distintos tipos de información geoespacial.

La relación entre los distintos elementos de una IDE puede ejemplificarse en el diagrama de la Figura 2.1.

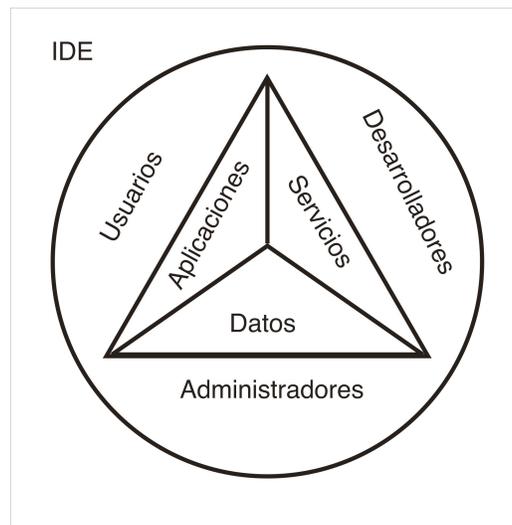


Figura 2.1: Relación entre los elementos de una IDE.

#### **2.2.4. Componentes arquitectónicos de una IDE basados en estándares abiertos**

La *OGC* trabaja en la especificación de interfaces abiertas enfocadas a la interoperabilidad entre sistemas, cuya última finalidad es la creación de soluciones geoespaciales dis-

tribuidas. Estas interfaces estandarizadas proveen un conjunto amplio de herramientas para el acceso y distribución de la información geográfica por medio del empleo de servicios Web geoespaciales.

A continuación se enlista un conjunto mínimo de estándares *OGC* empleado para el desarrollo de Infraestructuras de Datos Espaciales, clasificado según su utilidad:

- **Visualización:** la especificación *Web Map Server (WMS)* permite crear servicios para desplegar mapas a partir de capas superpuestas provenientes de diferentes fuentes. Estos mapas pueden desplegarse en línea usando de un navegador Web, permitiendo desplegar información geoespacial rápida y fácilmente, ya sea como auxiliares en la búsqueda de información geográfica, o para conocer el área de influencia de cierto conjunto de datos o simplemente para comparar visualmente dos conjuntos distintos de información, entre muchos otros casos de uso.

La especificación *WMS* define tres operaciones asociadas al “Mapeo en Línea”: *getMap*, *getCapabilities* y *getFeatureInfo*. Estas operaciones permiten respectivamente solicitar un mapa especificando sus parámetros espaciales, obtener información acerca de las capacidades del servicio, así como datos acerca de un rasgo presente en un mapa. A una implementación del servicio *WMS* que considera estas tres operaciones se le llama *WMS* básico [Inc02].<sup>8</sup>

- **Presentación:** la especificación llamada *Styled Layer Descriptor (SLD)* es complementaria y opcional a la especificación *WMS*. Permite tener control parcial sobre la representación visual de los rasgos dentro del mapa, al permitir crear y emplear simbología definida por el usuario. La definición de tales elementos se auxilia de un

---

<sup>8</sup>Aunque un *WMS* básico solamente requiere las operaciones *getMap* y *getCapabilities*, en la práctica resulta necesaria la operación *getFeatureInfo* aunque la especificación considere opcional su implementación.

lenguaje para especificar símbolos. El control que permite este lenguaje va desde la posibilidad de escoger los colores con los cuales se pinta una línea o el interior de un polígono, hasta la capacidad de seleccionar los rasgos que deseen desplegarse a partir de algún atributo [sld07].

- Almacenamiento: la especificación *Web Map Context (WMC)* sirve como soporte para el intercambio de la información espacial en forma de coberturas o mapas. Es semejante a *WMS* ya que permite escoger información a partir de varios criterios, en particular criterios espaciales. Es diferente de *WMS* porque mientras ésta permite especificar filtros y desplegar mapas estáticos, *WCS* proporciona los datos junto con descripciones detalladas, permitiendo realizar búsquedas complejas basadas en los datos originales [wmc05].
- Manipulación: la especificación llamada *Web Feature Service (WFS)* define un conjunto de interfaces para acceder y manipular datos geográficos empleando *HTTP* como una plataforma de cómputo distribuido. Estas interfaces permiten al usuario combinar, usar y administrar información geográfica. Las operaciones definidas por la especificación permiten crear y borrar rasgos, actualizar un rasgo particular, bloquear la manipulación de un rasgo, así como realizar consultas a partir de criterios espaciales y no espaciales [wfs02].
- Codificación: el lenguaje empleado para la codificación de rasgos geográficos se llama *Geographic Markup Language (GML)* y define una gramática escrita en *XML* adecuada para el modelado, transporte y almacenamiento de información geográfica. Este lenguaje proporciona una gran variedad de objetos para describir elementos geográficos, incluyendo rasgos, geometría, topología, sistemas coordinados de

referencia<sup>9</sup>, tiempo, unidades de medida y valores generalizados. Cada rasgo tiene atributos asociados que incluyen un nombre, tipo de dato y un valor [Por07].

- Consulta: la especificación *Filter Encoding* provee de un lenguaje definido en *XML* para establecer restricciones sobre los registros devueltos por una consulta. Estos filtros pueden manejar aspectos de la consulta tanto espaciales como no espaciales. Las expresiones definidas se basan en definiciones *BNF*<sup>10</sup> y pretenden describir un sistema neutral de consultas [fil05].

Por otra parte, el empleo de metadatos<sup>11</sup> es esencial para la creación, búsqueda y distribución de la información y servicios proporcionados por una IDE.

Un metadato es un archivo con información, normalmente presentado en *XML* que captura las características básicas de los datos o recursos informáticos. Estos permiten conocer el qué, quién, cuándo, dónde y el porqué de los recursos disponibles [Com06].

El estándar con mayor aceptación para la documentación de información geoespacial es el propuesto por la *Federal Geographic Data Committee (FGDC)* y recibe el nombre de *Content Standard for Digital Geospatial Metadata (CS-DGM)*. El estándar se desarrolla para ayudar a los usuarios a determinar la disponibilidad de la información geoespacial, la pertinencia de ésta para su uso particular, los mecanismos para disponer de la información, así como los mecanismos requeridos para transferirla [Gro98].

---

<sup>9</sup>Un sistema coordinado de referencia consiste de un conjunto coordinado de ejes asociado a la tierra a partir de un *datum*, el cual define el tamaño y forma de la Tierra.

<sup>10</sup>Backus Naur Form o gramáticas libres de contexto

<sup>11</sup>El término metadato en el contexto actual debe entenderse como un conjunto de información acerca de algo, ya sean datos espaciales, tablas, servicios, aplicaciones Web, etc.

# Capítulo 3

## Análisis y Diseño del sistema

### 3.1. Mapeo en línea

El acceso a la información geoespacial en forma de mapas digitales se ha visto influido por el empleo de tecnología como es el caso del uso de Internet, creando expectativas para el acceso a la información por medio de navegadores Web y herramientas en línea. Este acceso a mapas a través de la red o “mapeo en línea” debe desplegar y proporcionar información espacial en forma de mapas, de tal forma que sea fácil de interpretar usando únicamente las habilidades del usuario para leer mapas.

Una solución adecuada para proveer acceso a la información geográfica consiste en adoptar servicios Web geográficos para el mapeo en línea, que permitan obtener representaciones pictóricas de la información geoespacial o mapas geográficos de forma paramétrica, facilitando así el desarrollo de aplicaciones para la visualización de mapas en línea.

Las aplicaciones para la visualización de mapas en línea pueden dividirse en dos clases básicas: la primera proporciona mapas generales como fondo de alguna aplicación o para la ubicación de rasgos, mientras que la segunda clase trata de generar mapas interactivos y

emplear herramientas avanzadas de mapeo para consultar y modificar los mapas.

Las herramientas de visualización de mapas no se limitan a estas clases y en otros casos pueden aprovecharse para crear herramientas de búsqueda de información geoespacial, proporcionando una vista previa de la información en forma de mapa, junto con su extensión geográfica y atributos asociados.

Para aprovechar todas las ventajas que ofrecen las herramientas de visualización de mapas en línea, es necesario adoptar esquemas no propietarios de mapeo en línea con énfasis en la interoperabilidad, que permite que la información proveniente de diferentes fuentes pueda sobreponerse y ser operada a través de un conjunto de operaciones expuestas por interfaces comunes.

### 3.1.1. *OGC Web Map Service*

Para cumplir los objetivos deseados por las aplicaciones para la visualización de mapas en línea, se requiere tanto del planteamiento correcto de conceptos como del empleo de las herramientas adecuadas. Para el desarrollo de aplicaciones de visualización de mapas en línea es conveniente seguir el progreso de los grupos de trabajo del área, así como las recomendaciones y mejores prácticas<sup>1</sup> de tales organizaciones.

La *OGC* habla de interoperabilidad entre diversas IDE a partir del empleo de Servicios Web que proveen interfaces comunes para el intercambio de información.

El estándar propuesto por la *OGC* para el desarrollo de Servicios Web adecuados para la visualización de mapas georeferenciados se llama *Web Map Service* (*WMS* por sus siglas en inglés). Dicho estándar ha pasado por varias versiones, y la de más desarrollo y aceptación a la fecha es la versión 1.1.1. [Inc02]. En la siguiente lista se resumen sus características

---

<sup>1</sup>Conjunto de ideas administrativas sobre técnicas, métodos, procesos, actividades, incentivos o motivantes que resultan más efectivos para el cumplimiento de un objetivo particular, en contraste con otros conjuntos de ideas empleadas para el mismo objetivo.

más relevantes:

- Provee una sintaxis para cada una de las operaciones en forma de *URL*.
- Provee un metadato codificado en *XML*.
- Permite definir las capas solicitadas.
- Las imágenes se solicitan dentro de una extensión geográfica o *Bounding Box*.
- Permite especificar un Sistema de Referencia Espacial<sup>2</sup> o *SRS*<sup>3</sup>.
- Permite definir el formato de la imagen.
- Permite establecer el ancho y el alto de la imagen solicitada, además de poder definir un color para el fondo de la misma.

## Operaciones básicas

Las operaciones básicas que provee la interfaz son tres y se especifican dentro del parámetro *REQUEST*[Inc02]:

1. *GetCapabilities*: permite obtener información acerca del servicio Web, su contenido y el conjunto de parámetros que acepta. Esta operación se invoca por medio del protocolo *HTTP* y a través del método *GET*. Un ejemplo del *URL* y los parámetros necesarios para invocar la operación es el siguiente:

---

<sup>2</sup>Conjunto de parámetros que definen un nombre para un sistema de coordenadas, un identificador numérico, coordenadas que definen la máxima extensión de espacio posible, números que, cuando se utilizan en ciertas operaciones matemáticas, convierten las coordenadas recibidas en valores que se pueden procesar con máxima eficiencia. <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/opt/csbp3023.htm>

<sup>3</sup>Sigla de *Spatial Reference System*

```

http://host/wmsservice?
  VERSION=1.1.1&
  REQUEST=getCapabilities
  SERVICE=wms &

```

La descripción de los elementos de la operación *GetCapabilities* se observa en el cuadro3.1.

Elemento	Valor	Descripción
URL	http://host/wmsservice?	URL del servicio
VERSION	1.1.1	Versión de la especificación
REQUEST	getCapabilities	Operación solicitada de la interfaz
SERVICE	wms	Tipo de servicio solicitado

Cuadro 3.1: Descripción de los elementos de la operación *GetCapabilities*

El resultado de la operación es un documento *XML*, el cual es un metadato que describe el servicio *WMS*. Esta descripción contiene información de contacto, el tipo de peticiones que puede responder, las proyecciones que pueden emplearse para solicitar imágenes y el nombre de las capas disponibles, entre otros detalles. Al documento *XML* correspondiente a la respuesta se le llamará informalmente documento *Capabilities* y a continuación se presenta un ejemplo:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE WMT_MS_Capabilities SYSTEM
3 "http://schemas.opengis.net/wms/1.1.1/WMS_MS_Capabilities.dtd">
4 <WMT_MS_Capabilities version="1.1.1">
5 <!-- Detalles acerca del servicio: nombre del servicio ,
6 resumen, informacion de contacto, etc. -->
7 <Service>
8   <Name>OGC:WMS</Name>
9   <Title>Web Map Service NivelNacional2</Title>
10  <Abstract>ArcIMS 9.1.0 NivelNacional2 Web Map Service</Abstract>
11  <KeywordList><Keyword>ArcIMS</Keyword></KeywordList>
12  <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
13    xlink:href="http://host:8080/wms/NivelNacional2?"
14    xlink:type="simple"/>
15  <ContactInformation>...</ContactInformation>
16  <Fees>none</Fees>
17  <AccessConstraints>none</AccessConstraints>
18 </Service>
19 <!-- A partir de aqui comienza la descripcion de las capacidades del servicio-->
20 <Capability>
21 <!-- Detalles para la operacion getCapabilities-->
22
23 <Request>
24   <GetCapabilities>
25     <Format>application/vnd.ogc.wms.xml</Format>
26     <DCPType><HTTP>
27       <Get>
28         <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
29           xlink:href="http://host:8080/wms/NivelNacional2?"
30           xlink:type="simple"/>
31       </Get>
32     </HTTP></DCPType>
33 </GetCapabilities>
34
35 <!-- Detalles para la operacion getMap-->
36 <GetMap>
37   <Format>image/png</Format>
```

```
38 <Format>image/jpeg</Format>
39 <!-- Formatos MIME validos -->
40 <DCPType><HTTP>
41   <Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
42     xlink:href="http://host:8080/wms/NivelNacional2?"
43     xlink:type="simple"/>
44   </Get>
45 </HTTP></DCPType>
46 </GetMap>
47
48 <!-- Detalles para la operacion getFeatureInfo-->
49 <GetFeatureInfo>
50   <Format>application/vnd.ogc.wms_xml</Format>
51   <Format>text/xml</Format>
52   <Format>text/html</Format>
53   <Format>text/plain</Format>
54   <!-- Formatos permitidos para el regreso de informacion -->
55   <DCPType><HTTP>
56     <Get>
57       <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
58         xlink:href="http://host:8080/wms/NivelNacional2?"
59         xlink:type="simple"/>
60     </Get>
61   </HTTP> </DCPType>
62 </GetFeatureInfo>
63 </Request>
64
65 <!-- Formato de excepciones que puede devolver el servicio -->
66 <Exception>
67   <Format>application/vnd.ogc.se_xml</Format>
68   <Format>application/vnd.ogc.se_inimage</Format>
69   <Format>application/vnd.ogc.se_blank</Format>
70 </Exception>
71
72 <!-- Definicion de capas que ofrece el servicio -->
73
74 <Layer queryable="0" opaque="0" noSubsets="0">
```

```

75 <Title>NivelNacional2</Title>
76 <SRS>EPSG:4326</SRS>
77 <SRS>EPSG:4267</SRS>
78 <SRS>EPSG:4267</SRS>
79 <SRS>EPSG:4269</SRS>
80 <!-- Proyecciones aceptadas para el servicio. -->
81
82 <LatLonBoundingBox minx="-105.0196534715" miny="19.508737054"
83 maxx="-104.3839028341" maxy="20.0960991769"/>
84 <!-- Extension del mapa en coordenadas geograficas. -->
85 <BoundingBox SRS="EPSG:4267" minx="-105.0196534715" miny="19.508737054"
86 maxx="-104.3839028341" maxy="20.0960991769"/>
87 <!-- Extension del mapa en algun otro sistema de referencia espacial. -->
88
89 <Layer queryable="1">
90 <Name>11</Name>
91 <Title>ESTADOS</Title>
92 <SRS>EPSG:4326</SRS>
93 <LatLonBoundingBox minx="-117.12496" miny="14.53187"
94 maxx="-86.71012" maxy="32.71867"/>
95 </Layer>
96 <!-- Esta es una capa que se puede solicitar para incorporarla
97 dentro de un mapa, esta definida en coordenadas geograficas
98 y permite obtener informacion sobre los rasgos contenidos en el mapa. -->
99 </Capability>
100 </WMT_MS.Capabilities>

```

2. *GetMap*: devuelve la imagen de un mapa cuyos parámetros espaciales están bien definidos. Esta operación se invoca por medio del protocolo *HTTP* y a través del método *GET*. Un ejemplo del *URL* y los parámetros necesarios para invocar la operación es el siguiente:

`http://host/wmsservice?`

`REQUEST=getMap &`

```

SERVICE=WMS &
VERSION=1.1.1
WIDTH=4000 &
HEIGHT=4500 &
BBOX=' '-120.0,20.0,-119.0,21.0' ' &
SRS= EPSG:4326 &
LAYERS= 0,1,2,3 &
FORMAT= JPEG &
TRANSPARENT=false

```

Elemento	Valores	Descripción
URL	http://host/wmsservice?	URL del servicio
VERSION	1.1.1	Versión de la especificación
SERVICE	wms	Servicio solicitado
REQUEST	getMap	Tipo de petición
WIDTH	4000	Ancho de la imagen en pixeles
HEIGHT	4500	Alto de la imagen en pixeles
BBOX	"-120.0,20.0,-119.0,21.0"	Extensión geográfica
SRS	EPSG:4326	Sistema de Referencia Espacial
LAYERS	0,1,2,3	Capas solicitadas separadas por comas
FORMAT	JPEG	Formato de la imagen devuelta
TRANSPARENT	false	Indica la transparencia de la imagen

Cuadro 3.2: Parámetros de una petición *GetMap*

Los detalles de los parámetros empleados en una petición *GetMap* pueden observarse en el cuadro 3.2, cuyo resultado es una imagen con características correspondientes a los parámetros de la solicitud. La imagen puede verse en la Figura 3.1.

3. *GetFeatureInfo*: operación que permite obtener información sobre los rasgos presentes en un punto o área particular del mapa. El punto que se emplea para indicar la

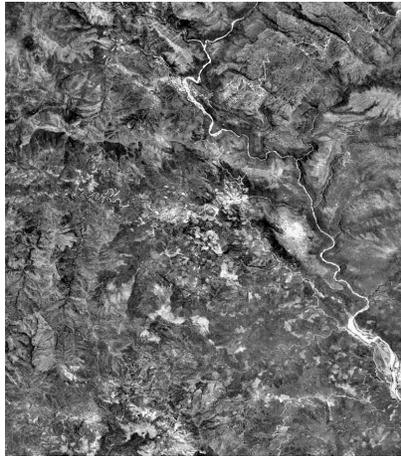


Figura 3.1: Imagen resultado de una petición *getMap*.

operación corresponde a un pixel sobre el mapa y no a una coordenada sobre la tierra. Esta operación es opcional dentro de la especificación *WMS*, se invoca por medio del protocolo *HTTP* y a través del método *GET*. Un ejemplo del *URL* y los parámetros necesarios para invocar la operación es el siguiente:

```
http://host/wmsservice?  
  VERSION=1.1.1 &  
  REQUEST=getFeatureInfo &  
  SERVICE=wms &  
  WIDTH=800 &  
  HEIGHT=900 &  
  BBOX=' -120.0,20.0,-119.0,21.0'  
  SRS= EPSG:4326 &  
  QUERY_LAYERS= 0,1,2,3 &  
  X= 45  
  Y= 100
```

Elemento	Valores	Descripción
http://host/wmsservice?		URL del servicio
VERSION	1.1.1	Versión de la especificación
REQUEST	getMap	Tipo de petición
SERVICE	wms	Servicio solicitado
WIDTH	800	Ancho de la imagen en pixeles
HEIGHT	900	Alto de la imagen en pixeles
BBOX	"-120.0,20.0,-119.0,21.0"	Extensión geográfica
SRS	EPSG:4326	Sistema de Referencia Espacial
QUERY_LAYERS	0,1,2,3	Capas cuya información es solicitada
X	[0...WIDTH)	Renglón de la imagen
Y	[0...HEIGHT)	Columna de la imagen

Cuadro 3.3: Parámetros de una petición GetFeatureInfo

## 3.2. Análisis del sistema

Cualquier aplicación desarrollada para la visualización de mapas en línea requiere considerar muchos detalles relacionados con la funcionalidad general del sistema, pero debe de hacer énfasis en las características asociadas con el despliegue de mapas. La guía que establecen las siguientes preguntas permite obtener buenos resultados en el desarrollo de aplicaciones para la visualización de mapas [Neb04] al mismo tiempo que ofrece una clasificación para ellas:

- ¿Se requiere visualizar en línea el mapa completo?
- ¿Se requiere sobreponer distintas capas de información provenientes de distintas fuentes?
- ¿La información contenida en los mapas es estática o dinámica?

- ¿Se trata de mapas interactivos?

Respecto a las preguntas planteadas, los desarrollos considerados en el presente trabajo corresponden a aplicaciones que sobreponen capas de información provenientes de distintas fuentes, cuyo contenido es estático. Esto ocurre debido a que las aplicaciones poseen un mecanismo de despliegue de mapas basado en mosaicos de imágenes a escalas fijas que se organizan en pirámides.

En términos de percepción remota, un mosaico es la composición de una imagen *raster* a partir de varias imágenes *raster* de tamaño menor. Para los fines de visualización de mapas, un mosaico será tomado como un conjunto de mapas de áreas adyacentes que comparten la misma referencia espacial y escala, en los cuales los bordes se han disuelto [gis06]. Un ejemplo de mosaico puede observarse en la Figura 3.2, la cual se compone de 64 imágenes de menor tamaño

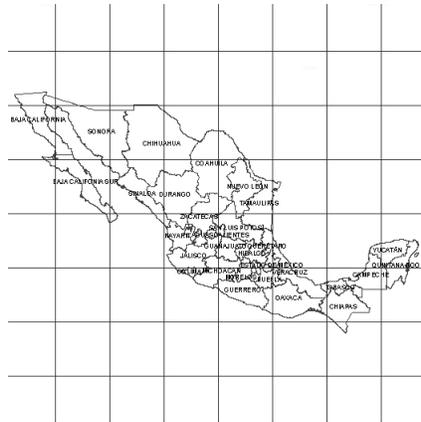


Figura 3.2: Un mosaico de imágenes.

Una *pirámide* [ESR],[Zei99] es una serie de imágenes de resolución reducida, usadas principalmente para mejorar el desempeño de la visualización de las imágenes cuando no se necesita trabajar con ésta en una resolución completa. Usualmente una imagen o *raster*

contiene mayor información de la que se puede mostrar en la pantalla; asimismo trabajar con una imagen completa requiere efectuar múltiples cálculos sobre ésta para decidir el subconjunto de datos necesario para procesar. Las pirámides guardan imágenes de menor resolución en distintos niveles y cuando se requiere una imagen se utiliza el nivel de la pirámide con resolución semejante a la pantalla o mapa. Esto reduce el espacio de búsqueda y por consiguiente el tiempo de utilización del servidor, el tiempo de respuesta y el tiempo necesario para realizar el despliegue de la información.

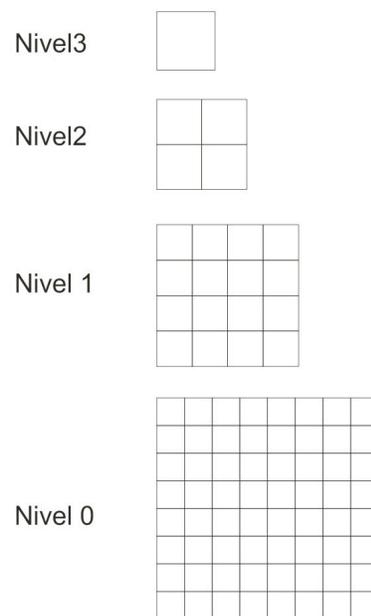


Figura 3.3: Pirámides *raster*.

Las pirámides son un conjunto de imágenes que se agrupan en capas o niveles, cada nivel corresponde a una representación de la imagen original con una menor resolución.

Las imágenes de la pirámide se generan a partir de las imágenes del nivel anterior, obteniendo cada vez imágenes de resolución menor. Si la pirámide comienza en el nivel 0 y contiene los píxeles de la imagen original, en el nivel superior inmediato se deben

integrar los píxeles del nivel anterior, es decir, se toman cuatro píxeles de nivel  $n$  para producir un píxel en el nivel  $n + 1$ . Este proceso continúa por cada nivel de la pirámide que deba generarse.

En términos de aplicaciones para el mapeo en línea, una práctica común es dividir el objeto de estudio, en este caso la Tierra, por medio de una rejilla regular cuyas celdas son cuadradas, formando un mosaico de imágenes. Los detalles concretos de las retículas varían según la proyección empleada. En este trabajo se consideran dos proyecciones a las cuales se ajusta una rejilla.

La primera proyección empleada se conoce como *proyección cilíndrica equidistante*, en la cual el cilindro es secante a la esfera. Esta proyección es equidistante a lo largo de los paralelos estándar y el ecuador. La distorsión de las distancias y las formas se incrementan proporcionalmente a medida que la región se acerca a los polos. Los paralelos se transforman en líneas horizontales a distancia constante uno del otro. Los meridianos se transforman en líneas ortogonales a los paralelos. El espacio entre ellos se incrementa a medida que se alejan del centro. Los polos se convierten en líneas rectas de igual longitud que el ecuador [Gon07]. La Figura 3.4 ilustra un ejemplo de esta proyección..

La siguiente proyección empleada se llama *Mercator Esférica* o *Mercator Simple*, [Gon07] y es una proyección cilíndrica conforme donde el cilindro es secante a la esfera. Se trata de una proyección ortomórfica o conforme, es decir, mantiene los ángulos. La escala es verdadera a lo largo del ecuador y de los paralelos estándar. La distorsión en la distancia y el área aumenta conforme se acerca a los polos, haciéndose infinita en éstos, por lo que los polos y regiones cercanas a éstos no pueden mostrarse. El factor de escala es constante a lo largo de los paralelos. Los meridianos se transforman en líneas rectas verticales y los paralelos se convierten en líneas rectas horizontales. La Figura 3.5 proporciona un ejemplo del mundo bajo esta proyección.

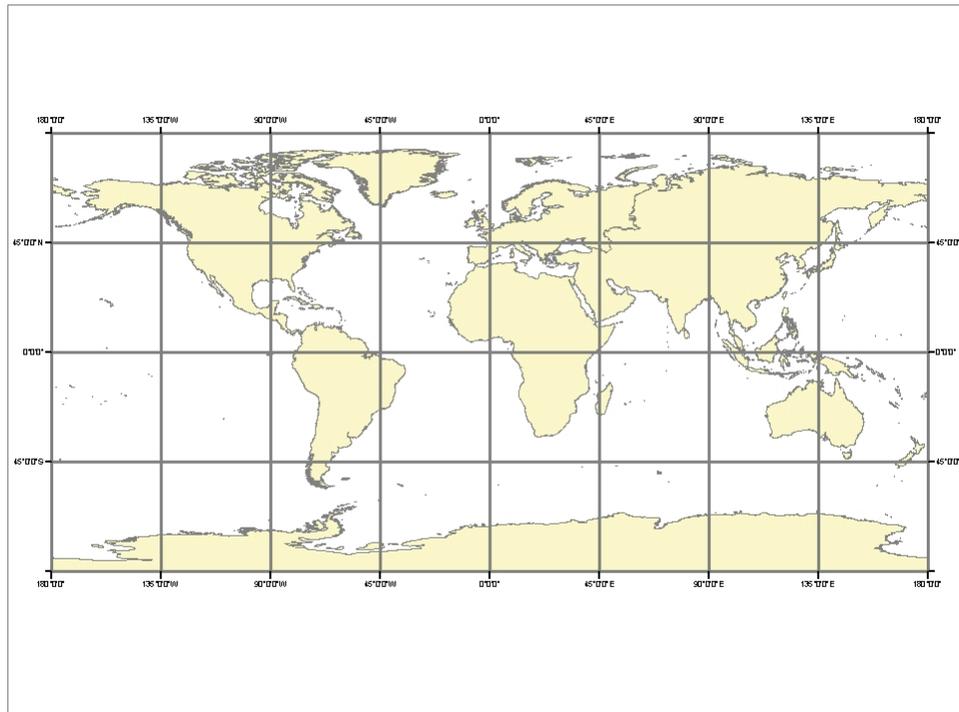


Figura 3.4: Mosaico de la Tierra en una proyección cilíndrica equirectangular que muestra una retícula cada 45 grados.

La generación de pirámides de imágenes es necesaria puesto que los servidores *WMS* representan un punto de acceso importante a la información y cualquier tipo de falla en este apartado produce problemas para brindar acceso a la información geográfica. Ya se ha mostrado de manera práctica[Mil] que los sistemas que proporcionan imágenes a través de la interfaz *WMS* no son capaces de soportar la demanda de trabajo impuesta por las herramientas de visualización de mapas en línea basadas en mosaicos de imágenes, ya que cada imagen solicitada debe de generarse procesando todos sus datos subyacentes, no importando si la misma imagen se vuelve a solicitar.

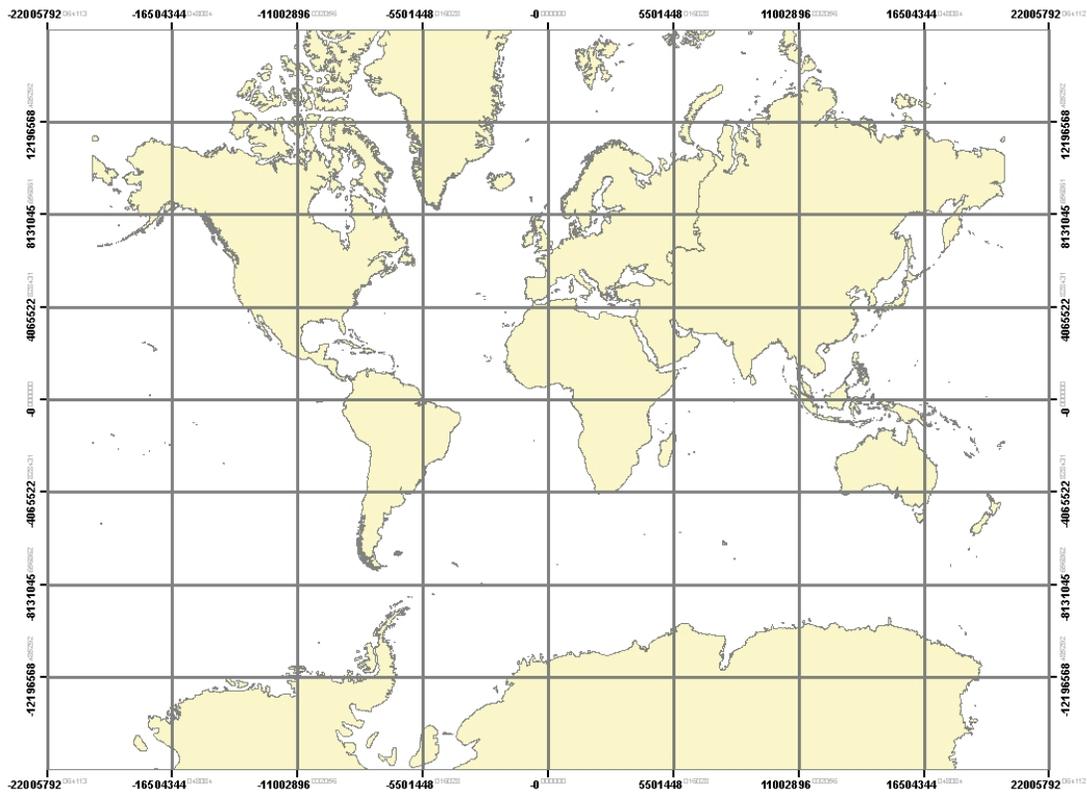


Figura 3.5: Mosaico regular de la Tierra para una proyección Mercator.

La especificación *WMS* versión 1.1.1 acepta un número arbitrario de parámetros para la generación de un mapa, sin embargo no contempla el preprocesamiento ni el almacenamiento de imágenes previamente solicitadas, ya que el número de peticiones posible es demasiado grande, y almacenar todas las imágenes generadas requeriría demasiado espacio, lo cual resulta poco práctico.

Cabe mencionar que aquellas aplicaciones que consumen imágenes basadas en mosaicos establecen restricciones de escalas, extensiones geográficas, así como del tamaño de los mosaicos; restricciones que sirven de guía para el diseño de sistemas que propicien la disminución en la carga de trabajo impuesta a los servidores *WMS*.

Una aplicación que despliegue mapas geográficos usando un esquema de mosaicos y que emplee la especificación *WMS* básica versión 1.1.1, mantendrá constantes todos de los parámetros especificados para una petición *getMap* por cada nivel o escala dentro de la pirámide. Únicamente el parámetro *BBOX* o *BoundingBox* cambia su valor, debido a que cada una de las celdas o mosaicos solicitados corresponde a una de las casillas inducidas por la retícula. Puesto que el parámetro *BBOX* corresponderá a alguna de las retículas especificadas por el mosaico, podemos decir que el número de posibilidades para éste se puede asociar a la retícula empleada, concluyendo que el *BoundingBox* toma valores fijos dentro de una colección dada. Entonces podemos decir que para un nivel de la pirámide, todas las peticiones *WMS* están dadas por un conjunto de parámetros constantes.

### 3.3. Diseño del sistema

Toda esta información puede aprovecharse para diseñar un mecanismo que reciba peticiones *WMS* y regrese las imágenes correspondientes almacenando copias de éstas en el sistema de archivos, previniendo que en caso de recibir nuevamente la misma petición, se evite generar nuevamente la imagen, devolviendo únicamente la copia previamente solicitada.

Un sistema con estas características se puede implementar como un *Servidor de Mapas en Cascada* o *Cascading Map Server (CMS)*, el cual se define como un servicio *WMS* que se comporta como un *WMS* para otros clientes. Un servicio de este tipo emplea otros servicios *WMS* para agregar funcionalidad como reproyección, cambio del formato de imágenes o almacenamiento de capas, entre otras funcionalidades [Inc02].

El desarrollo de un *CMS* produce una solución transparente<sup>4</sup> e involucra modificaciones

---

<sup>4</sup>Se dice que es transparente porque los cambios no son perceptibles por el cliente.

simples a nivel cliente. Los clientes consumen información empleando los mecanismos propuestos por el estandar *WMS* 1.1.1 pero para aprovechar correctamente [OSG06] este desarrollo debemos agregar parámetros adicionales a las peticiones creadas por el cliente al momento de solicitar una imagen.

El conjunto de parámetros adicionales a los parámetros *WMS* básico puede observarse en el Cuadro 3.4; tales extensiones indican si la petición debe obtenerse del almacén de imágenes o si cada imagen debe volver a generarse a partir del servidor *WMS*. También se debe indicar la aplicación que consume las imágenes para poder validar las peticiones acordes al tipo de fuente.

<b>Párametro</b>	<b>Valores</b>	<b>Descripción</b>
TILED	true / false	Indica si la imagen debe provenir del caché
SOURCE	ID	Identificador que indica la aplicación que origina la petición y sirve para determinar los mecanismos para almacenar y obtener la imagen.

Cuadro 3.4: Parámetros adicionales empleados para las peticiones *getMap*.

El conjunto completo del sistema puede verse como un sistema *WMS* convencional con la diferencia de que éste permite aprovechar las ventajas de consumir conjuntos de imágenes previamente generados. Véase Figura 3.6.

En el servidor, tanto el conjunto de parámetros *WMS* como las extensiones propuestas permiten que cada petición realizada dentro de las restricciones expuestas sea identificada correctamente y almacenada en el sistema de archivos.

En resumen, la tupla que identifica cada petición de manera única se forma por los elementos mostrados en el Cuadro 3.5:

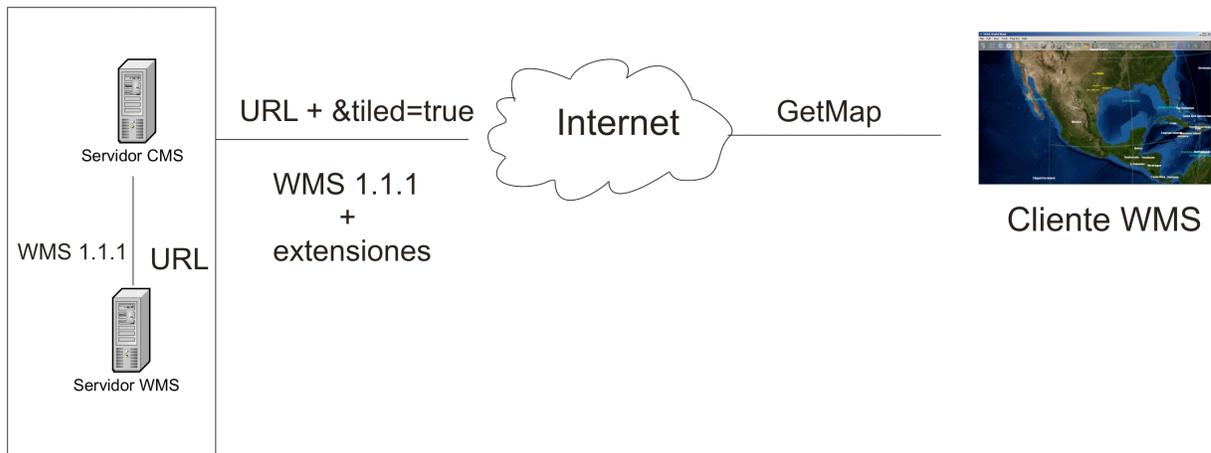


Figura 3.6: Modelo del sistema.

Elemento	Descripción
Aplicación cliente	Determina la aplicación que origina la petición.
Nombre del servicio	Identifica el servicio <i>WMS</i> .
Capas solicitadas	Indica las capas solicitadas para el servicio.
Ancho del mosaico	Identifica el nivel de la pirámide
Origen del mosaico en el eje X	Coordenada extrema izquierda
Origen del mosaico en el eje Y	Coordenada extrema inferior

Cuadro 3.5: Elementos de la tupla que identifica cada petición *WMS getMap*.

## Alcance del trabajo

El sistema que se pretende generar está dirigido a toda persona interesada en publicar mapas estáticos accesibles a través de la especificación *WMS*, así como al desarrollo de aplicaciones para la visualización de mapas que empleen mosaicos de imágenes organizados en pirámides de múltiples resoluciones.

La correcta utilización del sistema permite disminuir la carga de trabajo en los servidores *WMS* y, por consiguiente, aumentar la disponibilidad de la información. Una mayor disponibilidad de la información se refleja en una mejor calidad de servicio y, por lo tanto,

en aplicaciones de mayor calidad. Estos factores resultan en un consiguiente incremento en el número de usuarios potenciales y en un mayor acceso a la información.

### 3.4. Requerimientos del sistema

Se requiere desarrollar un sistema *WMS* en cascada que implemente las operaciones descritas en la especificación *WMS* versión 1.1.1, en su versión básica. Adicionalmente, se requiere que las imágenes solicitadas sean almacenadas para su posterior uso, siempre que las peticiones correspondan a las retículas empleadas y sean válidas.

Estas funcionalidades permiten guiar los casos de uso del sistema y se desglosan en la siguiente lista:

- Recibir peticiones *WMS*: *getCapabilities*, *getFeatureInfo* y *getMap*.
- Para cada petición recibida, se debe responder adecuadamente según una lista propia de capas y servicios disponibles, siempre acorde a la especificación *WMS* básica.
- Almacenar una copia de las imágenes solicitadas en el sistema de archivos siempre que la petición generada corresponda a una retícula definida en el servicio.
- En caso de recibir una solicitud cuya imagen ya se haya generado, devolver la imagen desde el sistema de archivos, evitando así la redirección hacia el servidor *WMS* y por tanto la regeneración de la misma.
- Almacenar todas las imágenes solicitadas a partir del *URL* de la petición.
- Manejar adecuadamente los casos de error según la especificación *WMS*.

### 3.5. Casos de uso

A continuación se presentan los casos de uso asociados al sistema. Se distinguen dos actores que no son usuarios convencionales sino sistemas puesto que hablamos del desarrollo de servicios Web y de interacciones entre ellos.

Los casos de uso para el cliente *WMS* son los siguientes:

- El cliente *WMS* solicita una imagen nueva correspondiente a un mosaico.
- El cliente *WMS* solicita una imagen previamente almacenada correspondiente a un mosaico.
- El cliente *WMS* pregunta acerca del servicio.
- El cliente *WMS* solicita información acerca de un rasgo visible en el mapa.

Otro conjunto de casos de uso se presenta cuando el *CMS* debe solicitar imágenes o información al servidor *WMS* fuente. Estos casos de uso son los siguientes:

- El *CMS* solicita una imagen al servidor *WMS*.
- El *CMS* pregunta acerca del servicio *WMS*.
- El *CMS* solicita información acerca de un rasgo visible en el mapa.

Un diagrama con los casos de uso resumidos puede observarse en la Figura 3.7.

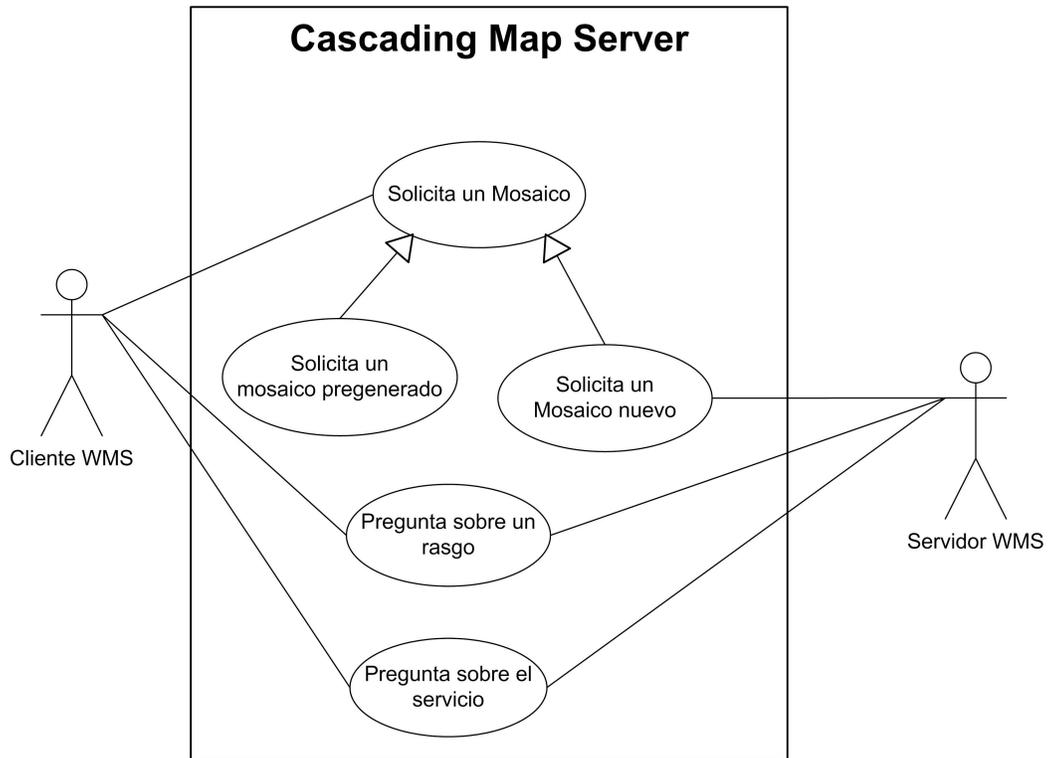


Figura 3.7: Diagrama de *casos de uso* del sistema.



# Capítulo 4

## Desarrollo del sistema

### 4.1. Tecnología

#### Plataforma de desarrollo.

La implementación del *CMS* se realiza utilizando la plataforma Java en su edición estándar. Ésta comprende el uso del lenguaje de programación Java, herramientas y diversas *API*<sup>1</sup>, así como bibliotecas adicionales para el control de la entrada y salida de datos a través de la red, disco y memoria, entre otras [jav08a].

Java es un lenguaje programación de propósito general, orientado a objetos y explícitamente “tipado”. El lenguaje se compila a *bytecode*, el cual es un conjunto de instrucciones binarias que pueden ejecutarse dentro de una máquina virtual para Java.

---

<sup>1</sup>Del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones

## Servidor de Aplicaciones WEB

La tecnología *Java Servlet* [jav08b] proporciona a los desarrolladores un mecanismo simple y consistente, cuyo objetivo es extender la funcionalidad de un servidor Web para acceder al conjunto de sistemas de negocios existente. Se pueden considerar a los componentes desarrollados bajo esta tecnología como aplicaciones Java que se ejecutan del lado del servidor pero sin interfaz gráfica. La tecnología *Java Servlet* posibilita la creación de aplicaciones avanzadas por medio del empleo de componentes reutilizables que permiten la creación de aplicaciones Web, sin las limitantes respecto al número de aplicaciones *CGI* que pueden ejecutarse de manera simultánea.

Esta tecnología puede acceder por completo al *API* de Java y diversas *API* complementarias para realizar llamadas *HTTP* específicas. Los *Servlets* aprovechan toda la madurez del lenguaje de programación *Java*, el cual permite la portabilidad, desempeño, reusabilidad y protección contra errores del código generado, facilitando así la creación de aplicaciones Web interactivas.

La tecnología *Java Servlet* requiere de un contenedor que permita la ejecución y administración de los *Servlets*; el contenedor empleado en el presente trabajo es *Apache Tomcat* [apa08], debido a que es una implementación diseñada para ser rápida, eficiente y rigurosa respecto a la especificación de *Servlets*. Adicionalmente, esta implementación está liberada bajo una licencia abierta que permite el uso y distribución de los productos desarrollados bajo este contenedor.

## Servicios Web geográficos

Los Servicios Web OpenGIS (*OpenGIS Web Services*) son componentes concebidos como bloques de construcción para el desarrollo de aplicaciones geoespaciales, que a su vez

forman parte de un paradigma para la solución de problemas geoespaciales, el cual propone la creación de una “Red Espacial” o *Spatial-Web* que impone restricciones conceptuales y prácticas acerca del funcionamiento de los Servicios Web *OpenGIS*.<sup>[Inc03]</sup>

Las restricciones conceptuales para la creación de los Servicios Web *OpenGIS* se definen como sigue:

- Deben crearse y orientarse como servicios, es decir, adoptar la filosofía Cliente/Servidor.
- Deben de pensarse como componentes interconectables.
- Deben ser auto descriptivos.
- Los servicios deben ser libres de estado, es decir, las operaciones expuestas por el servicio deben ser atómicas<sup>2</sup>.

Las restricciones de implementación están orientadas a lograr la interoperabilidad y se describen en los siguientes puntos:

- Se debe emplear *XML* como formato de intercambio.
- Se utiliza *HTTP* como mecanismo de transporte.
- Las interfaces desarrolladas deben ser estrictamente definidas por una sintaxis.
- Deben de contar con modelos para describir los servicios y metadatos relacionados.

---

<sup>2</sup>Deben realizarse en una sola llamada a un procedimiento y por tanto, no depender de estados anteriores.

## 4.2. Implementación del *Cascading Map Server*

El servicio que se implementa en el presente trabajo es un Servidor de Mapas en Cascada o *CMS*, el cual por definición funciona como un servicio *WMS* para los clientes que lo consultan y, a su vez, contiene otros servicios *WMS*.

El empleo del *CMS* implementado permite reducir el tiempo de respuesta tras el envío de una petición *WMS* solicitando una imagen georeferenciada. Esto tiene por consecuencia un aumento en la velocidad de despliegue de mapas para las aplicaciones que consumen imágenes en forma de mosaicos organizados en pirámides de múltiples resoluciones. Para el servidor *WMS* involucrado, el empleo de este *CMS* permite reducir la carga de trabajo impuesta por solicitud masiva de mapas al almacenar una copia de todas las imágenes generadas, devolviendo una copia cuando la imagen vuelve a ser solicitada.

La implementación del servicio *CMS* propuesto sigue la guía establecida por la especificación *WMS* versión 1.1.1 [Inc02], la cual define una sintaxis para invocar las operaciones disponibles por medio de los *URL*. Esta implementación requiere básicamente extensiones a nivel servidor dentro de la operación *getMap*.

Las operaciones *getCapabilities* y *getFeatureInfo* requieren modificaciones menores pero de consideración dentro del presente trabajo, ya que sólo se trata de redirigir peticiones *WMS*. El conjunto de extensiones y modificaciones se exponen a continuación.

### 4.2.1. Documento *Capabilities*

Es importante que el *CMS* indique dentro del documento *Capabilities* toda funcionalidad adicional respecto al servidor *WMS*, que en el presente caso, permite emplear un esquema de caché para almacenar mosaicos de imágenes. Otro elemento que se debe indicar dentro del documento *Capabilities* es el conjunto de parámetros adicionales, los cuales

permiten aprovechar el esquema de caché cuando se realiza la solicitud de un mapa. A estos parámetros adicionales se les conoce como parámetros específicos del proveedor o *VSP*<sup>3</sup>

La especificación *WMS* implementada [Inc02] permite el uso de parámetros adicionales o *VSP* que permitan mejorar los resultados de una petición, además de indicar que cualquier cliente estándar no está obligado a hacer uso de éstos parámetros para poder utilizar el servicio de forma normal.

La implementación de un Servicio Web *OGC* debe producir resultados válidos incluso si los *VSP* no están presentes o se encuentran mal formados. En estos casos, el servicio implementado debe de proveer estos parámetros y asignarles valores predeterminados.

Un Servicio Web *OGC* puede no indicar todos o algunos de estos parámetros. En caso de incluir los parámetros dentro del documento *Capabilities*, deben definirse por medio de un *DTD* interno dentro del documento *XML*.

A continuación se presentan los detalles relevantes de la implementación del *CMS*, referentes a las tres operaciones de la especificación *WMS*, haciendo énfasis en los detalles asociados a las extensiones para acelerar el despliegue de imágenes.

## Catálogo de servicios

Un *Cascading Map Server* es un servicio *WMS* que retransmite otros servicios *WMS* provenientes de diferentes fuentes. Esta retransmisión utiliza un catálogo simple para mantener la asociación entre servicios y fuentes.

La estructura del catálogo propuesto se conforma de la siguiente manera:

- Nombre del servicio *CMS*.
- *URL* del servicio *WMS* retransmitido.

---

<sup>3</sup>Siglas de Vendor-Specific Parameters .

- URL de la operación *getCapabilities* del servicio *WMS* retransmitido.
- URL de la operación *getMap* del servicio *WMS* retransmitido.
- URL de la operación *getFeatureInfo* del servicio *WMS* retransmitido.
- Ruta local del documento *Capabilities* correspondiente al *CMS*.
- Ruta local del documento *Capabilities* correspondiente al *WMS*.

Un ejemplo del catálogo asociado al *CMS* se puede ver en la Figura 4.1. En ésta se pueden ver dos servicios declarados: *NivelNacional* y *NivelRegional*. En cada caso, se encuentran los *URL* necesarios para redirigir cada una de las peticiones *WMS* dependiendo de la operación solicitada.

```
1 <?xml version="1.0"?>
2 <services >
3 <service name="NivelNacional"
4 URL="http://localhost/wmsconnector/com.esri.wms.Esrimap/NivelNacional?"
5 getMapURL = "http://host/wms/com.esri.wms.Esrimap/NivelNacional?"
6 getCapabilitiesURL = "http://host/wms/com.esri.wms.Esrimap/NivelNacional?"
7 getFeatureInfoURL = "http://host/wms/com.esri.wms.Esrimap/NivelNacional?"
8 />
9 <service name="NivelRegional"
10 URL="http://localhost/wmsconnector/com.esri.wms.Esrimap/NivelRegional?"
11 getMapURL = "http://host/wms/com.esri.wms.Esrimap/NivelRegional?"
12 getCapabilitiesURL = "http://host/wms/com.esri.wms.Esrimap/NivelRegional?"
13 getFeatureInfoURL = "http://host/wms/com.esri.wms.Esrimap/NivelRegional?"
14 />
15 </services >
```

Figura 4.1: Ejemplo de un catálogo de servicios.

Existen restricciones acerca del contenido del catálogo y se enlistan a continuación:

- El nombre del servicio *CMS* es único y sirve como identificador o llave dentro del catálogo

- La ruta del documento *Capabilities* para el servicio *CMS* debe ser única.
- La ruta del documento *Capabilities* para el servicio *WMS* debe ser única.

La ventaja de emplear un catálogo radica en que pueden existir dos servicios con nombres distintos que compartan la misma fuente. Esto posibilita que en futuras implementaciones se puedan usar estilos diferentes y parámetros adicionales. El *CMS* emplea la tecnología disponible en el servidor Web de aplicaciones para deducir, a partir del *URL* de la petición, el nombre del servicio solicitado dentro del catálogo. En caso de no existir el servicio dentro del catálogo, el *CMS* indica que dicho recurso no existe, devolviendo un código de error *HTTP*.<sup>4</sup>

#### 4.2.2. *GetCapabilities*: obtener información del servicio.

Los servicios *WMS* son autodescriptivos y cuentan con mecanismos para describir sus capacidades. Esta descripción muestra el contenido o colección de datos geográficos disponibles, así como también el conjunto de parámetros que acepta.

El proceso para obtener la información asociada a un *WMS* se realiza en la ejecución de la operación *GetCapabilities*, ya que una llamada a esta operación regresa una descripción detallada sobre el contenido a nivel de servicio sobre el conjunto de datos disponibles y acerca del conjunto de parámetros aceptables para llamar a las distintas operaciones disponibles.

### Parámetros de la petición

La forma de una petición *GetCapabilities* está definida por un *URL* y un conjunto básico de parámetros definidos a continuación:

---

<sup>4</sup>Ver sección 4.2.5 acerca del manejo de errores.

Parámetro	Valor	Requerido/ Opcional	Descripción
VERSION	1.1.1	O	Versión Solicitada
SERVICE	WMS	R	Tipo de servicio
REQUEST	getCapabilities	R	Nombre de la operación

Cuadro 4.1: Descripción de los parámetros de la operación *GetCapabilities*

## Implementación

La implementación que procesa esta operación requiere verificar la existencia de dos de tres parámetros requeridos: *SERVICE* y *REQUEST*. El comportamiento ante el parámetro opcional *VERSION* está definido por las reglas para la negociación de versiones.<sup>5</sup> El valor predeterminado del parámetro *VERSION* es “1.1.1” y el documento *Capabilities* devuelto será uno correspondiente a dicha versión.

Esta implementación no requiere parámetros adicionales y cualquier otro parámetro es innecesario para obtener una respuesta satisfactoria.

## Respuesta

La respuesta a una petición válida de solicitud de información sobre el servicio es un documento *XML*, el cual debe ser válido de acuerdo con la definición *XML Document Type Definition* (DTD) correspondiente. Este documento especifica el contenido de la respuesta, sus elementos obligatorios, así como también cada uno de sus elementos opcionales. El documento *XML DTD* debe estar ubicado en un *URL* calificado o válido. Al documento *XML* se le nombra informalmente como documento *Capabilities*, el cual contiene múltiples elementos que describen el servicio y la información contenida.

<sup>5</sup>Los detalles respecto a la negociación de versiones se encuentran en la especificación para la implementación de WMS [Inc02] dentro de la sección de “Elementos Básicos del Servicio”, en la Sección 6.1 de dicho documento.

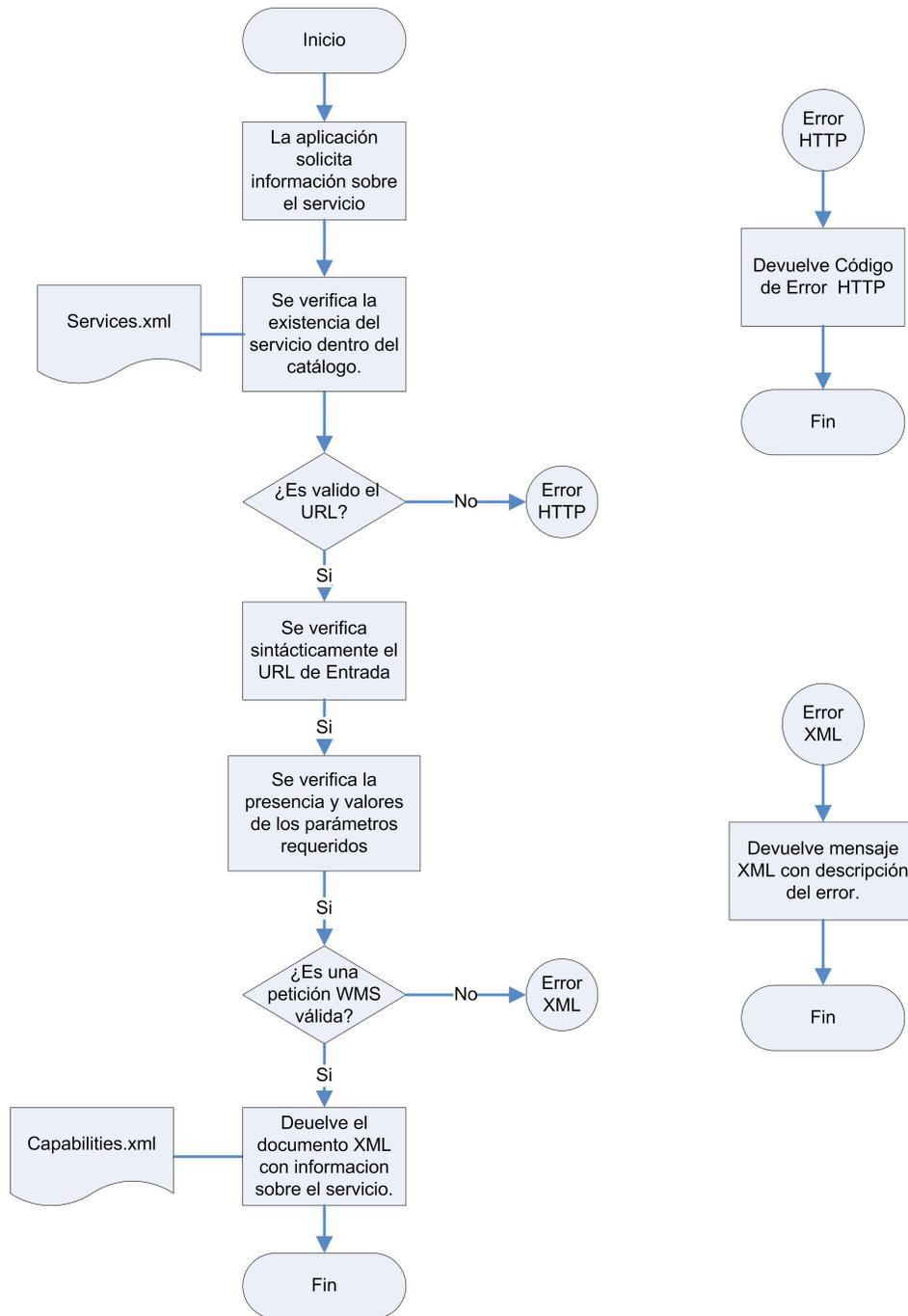


Figura 4.2: Diagrama de Flujo de la operación getCapabilities

## Documento *Capabilities*

Cada servicio *WMS* posee un documento *Capabilities* que describe su contenido y los parámetros aceptados. Para el desarrollo de este sistema *CMS*, se crea una versión específica a partir del documento *Capabilities* proveniente del servidor *WMS* original. Este acercamiento implica realizar modificaciones tanto en partes opcionales como en partes obligatorias, indicando que se trata de un *CMS*, el cual proporciona funcionalidad adicional al servidor original, permitiendo el empleo de imágenes previamente generadas basadas en mosaicos. No se pretende completar elementos opcionales no presentes, evitando así ofrecer funcionalidades adicionales.

El documento *Capabilities* se compone de dos nodos o secciones, la primera sección con etiqueta `< Service >`, proporciona información general acerca del servicio, mientras que la segunda, con etiqueta `< Capabilities >`, está encargada de publicar las capacidades del servicio *WMS*.

### Sección *Service*

Los cambios en la sección con etiqueta `< Service >` se realizan en dos de sus elementos:

- El primer cambio ocurre en el elemento `< Abstract >`, donde al texto que proporciona un resumen acerca del servicio, se le añade un texto explicativo adicional para indicar que se trata de un *CMS* que despacha imágenes almacenadas en caché para aplicaciones que consumen imágenes como mosaicos.
- El segundo cambio ocurre en el elemento `< OnlineResource >`, que reemplaza el valor del atributo `xlink:href`, por uno nuevo que indica el *URL* del servidor *CMS*.

En la Figura 4.3 se presenta un ejemplo de un documento *Capabilities* que muestra los cambios realizados. El primero de ellos aparece en la línea 4, mientras que el segundo

cambio ocurre en la línea 6.

```
1 <Service >
2   <Name>OGC:WMS</Name>
3   <Title>Web Map Service NivelNacional2 </Title >
4   <Abstract>ArcIMS 9.1.0 NivelNacional2 Web Map Service </Abstract >
5   <KeywordList><Keyword>ArcIMS </Keyword></KeywordList>
6   <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
   xlink:href="http://host:8080/wmsconnector/NivelNacional2?"
   xlink:type="simple"/>
7   <ContactInformation >... </ContactInformation >
8   <Fees>none </Fees >
9   <AccessConstraints >none </AccessConstraints >
10 </Service >
```

Figura 4.3: Sección *Service* de un documento *Capabilities* modificado.

### Sección *Capabilities*

La sección *< Capabilities >* se compone de tres elementos:

- *< Request >*,
- *< Layer >*,
- *< Exception >* .

El elemento *< Request >* indica el conjunto de operaciones disponibles y la plataforma de cómputo distribuido empleada, que en este caso es la *World Wide Web*, o más específicamente, un conjunto de *anfitriones* de Internet que implementan el Protocolo de Transferencia de Hipertexto o *HTTP* [Inc02]. Esta plataforma es la única reconocida explícitamente por los servicios Web *OGC*.

En la Figura 4.4 se observan las modificaciones al elemento *< Request >*, que ocurren en las tres operaciones disponibles: *< GetCapabilities >*, *< GetMap >* y *< GetFeatureInfo >*. Cada

operación posee un elemento `< OnlineResource >`, que debe modificarse para reflejar el *URL* del servidor *CMS*. Las modificaciones se presentan en las líneas 17, 27 y 40, para cada una de las operaciones disponibles respectivamente.

```

11 <Capabilities >
12 <Request>
13   <GetCapabilities >
14     <Format>application/vnd.ogc.wms_xml</Format>
15     <DCPType><HTTP><Get>
16       <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
17         xlink:href="http://CMSHost/CMS/NivelNacional2?" xlink:type="simple"/>
18     </Get> </HTTP></DCPType>
19   </GetCapabilities >
20
21 <GetMap>
22   <Format>image/png</Format>
23   <Format>image/jpeg</Format>
24   <DCPType><HTTP>
25     <Get>
26       <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
27         xlink:href="http://CMSHost/CMS/NivelNacional2?" xlink:type="simple"/>
28     </Get>
29   </HTTP></DCPType>
30 </GetMap>
31
32 <GetFeatureInfo >
33   <Format>application/vnd.ogc.wms_xml</Format>
34   <Format>text/xml</Format>
35   <Format>text/html</Format>
36   <Format>text/plain</Format>
37   <DCPType><HTTP>
38     <Get>
39       <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
40         xlink:href="http://CMSHost/CMS/NivelNacional2?"
41         xlink:type="simple"/>
42     </Get>
43   </HTTP></DCPType>
44 </GetFeatureInfo >
45 </Request >

```

Figura 4.4: Modificaciones al elemento *Request* dentro de la sección *Capabilities*.

Otro de los elementos pertenecientes a la sección `< Capabilities >` que presenta modificaciones es el nodo `< Layer >`, que posee un atributo llamado *cascaded* que indica el número de veces que

la capa expuesta por dicho nodo en ha sido retransmitida a través de un *CMS*. Dentro del *CMS*, el valor para este atributo debe actualizarse para indicar el número de retransmisiones previas. La aparición de este atributo es opcional cuando su valor es igual a cero, pero en el caso de poseer un valor distinto a cero, su aparición es obligatoria. En el caso del presente trabajo, el valor del atributo siempre es mayor o igual a uno para todas las capas y servicios disponibles debido a que siempre ocurre una retransmisión.

En caso de que un *WMS* retransmita una capa de información y dicha capa no posea el atributo *cascaded*, el *CMS* debe de agregar el atributo y asignarle el valor de uno.

En la Figura 4.5, se observa en la línea 57 la descripción de una capa con título “ESTADOS”, la cual originalmente no poseía el atributo *cascaded*, pero que se agrega y le es asignado el valor de uno.

```
46 <Layer queryable="0" opaque="0" noSubsets="0">
47   <Title >NivelNacional2 </Title >
48   <SRS>EPSG:4326 </SRS>
49   <SRS>EPSG:4267 </SRS>
50   <SRS>EPSG:4267 </SRS>
51   <SRS>EPSG:4269 </SRS>
52   <LatLonBoundingBox minx="-105.0196534715" miny="19.508737054"
53     maxx="-104.3839028341" maxy="20.0960991769"/>
54   <BoundingBox SRS="EPSG:4267" minx="-105.0196534715" miny="19.508737054"
55     maxx="-104.3839028341" maxy="20.0960991769"/>
56
57 <Layer queryable="1" cascaded='1' >
58   <Name>11</Name>
59   <Title >ESTADOS</Title >
60   <SRS>EPSG:4326 </SRS>
61   <LatLonBoundingBox minx="-117.12496" miny="14.53187"
62     maxx="-86.71012" maxy="32.71867"/>
63 </Layer>
64
65 </Capability >
66 </WMT_MS_Capabilities >
```

Figura 4.5: Modificaciones al elemento *< Layer >* dentro de la seccion *Capabilities*.

### 4.2.3. *GetMap*: obtener una imagen georeferenciada.

La operación *getMap* está diseñada para producir mapas que pueden estar formados por imágenes o conjuntos de elementos gráficos. Las características del mapa devuelto deben de corresponder al conjunto de parámetros *CGI* y estar comprendidos en el documento *Capabilities* correspondiente.

## Parámetros de entrada

La llamada a la operación *GetCapabilities* requiere de un URL y de un conjunto de parámetros *CGI*. Esta implementación considera únicamente el uso de servicios *WMS* básicos, que son todos aquellos que no emplean “estilos nombrados” ni documentos *SLD* como mecanismos de despliegue personalizado.

Los parámetros empleados por un servicio *WMS* básico se presentan en el Cuadro 4.2.

## Implementación

La operación para solicitar mapas *getMap* contempla dos casos básicos de operación. El primer caso es cuando la imagen solicitada debe ser “fresca”, al no provenir de un caché de imágenes, mientras que el otro caso es cuando la petición requiere una imagen proveniente del caché.

## Imagen fresca

Cuando se requiere una imagen fresca, ésta se especifica a través de los parámetros *CGI* de alguna de las siguientes dos formas: la primera es cuando el parámetro *tiled* se encuentra ausente y debe tomar su valor predeterminado, es decir: *tiled = false*. La segunda es cuando el parámetro *tiled* explícitamente tiene el valor *false*. La ausencia del parámetro *tiled* o el uso de valores inapropiados no debe de afectar el comportamiento ordinario de la operación *getMap*, ya que este parámetro es del tipo *VSP* y la especificación implementada indica que se responda de esta manera. En esta situación, el parámetro toma el valor predeterminado *false*.

Parámetro CGI	Valor predeterminado	Requerido/Opcional	Descripción
VERSION	1.1.1	R	Versión solicitada del servicio.
SERVICE	N/A	R	Tipo de servicio solicitado.
REQUEST	N/A	R	Tipo de operación a realizar.
STYLES	N/A	R	Lista separada por comas de los “estilos nombrados” empleados para dibujar cada capa.
WIDTH	N/A	R	Ancho de la imagen en pixeles
HEIGHT	N/A	R	Alto de la imagen en pixeles
BBOX		R	Extensión geográfica de la petición
SRS		R	Sistema de Referencia Espacial
LAYERS	N/A	R	Capas solicitadas
FORMAT	N/A	R	Formato de salida del mapa.
TRANSPARENT	false	O	Indica la transparencia en la imagen
BGCOLOR	0xFFFFFFFF	O	Indica qué color será transparente.
EXCEPTIONS	SE_XML	O	Tipo de excepción devuelta en caso de error.
TIME	false	O	Indica el tiempo de la imagen deseada
ELEVATION	false	O	Indica la elevación de la imagen
TILED	false	O	VSP que especifica el uso del caché al CMS.
SOURCE	false	O	VSP que especifica la aplicación que origina la petición.

Cuadro 4.2: Descripción de los parámetros en la operación *GetMap*

La realización de la operación *getMap* solicitando una imagen fresca se sintetiza en los siguientes pasos:

1. Recibir la petición *WMS* del cliente.
2. Obtener el nombre del servicio *WMS* solicitado a partir del *URL* de la petición.
3. Verificar la existencia del servicio dentro del catálogo.
4. Obtener los parámetros del *CGI* y sus valores.
5. Se verifica la operación *getMap*.
6. Verificar la existencia de los parámetros obligatorios a la operación.
7. Verificar los valores para ciertos parámetros obligatorios.
8. Confirmar que la imagen solicitada sea fresca.
9. Transformar la petición *WMS* recibida por el *CMS* en una petición de acuerdo al servidor *WMS* original, para enviarla posteriormente al cliente.
10. Obtener la respuesta del *WMS* original, la cual es una imagen correspondiente a la petición enviada.
11. Retransmitir la imagen recibida al cliente.

En el paso 3, la verificación del servicio dentro del catálogo se auxilia de una estructura de datos conocida como *Hash Table*<sup>6</sup> para guardar información acerca de los servicios disponibles. La implementación emplea una tabla compartida y permite realizar búsquedas en tiempo  $O(1)$  cuando los datos del catálogo poseen una distribución uniforme.

---

<sup>6</sup>Estructura de datos que emplea una función Hash para identificar la posición de un objeto dentro de una tabla a partir de una llave única.

En el paso 4, que es donde se obtienen los parámetros del *CGI* junto con sus valores y se emplea una tabla Hash para normalizar las llaves o parámetros *CGI*. La normalización se debe realizar puesto que los parámetros *CGI* pueden escribirse indistintamente en mayúsculas, minúsculas o una combinación de ellas.

El paso 5 solamente confirma que el valor del parámetro *request* corresponda a una operación *getMap* y que el servicio solicitado sea del tipo *WMS*.

En el paso 6 se verifica la existencia de los parámetros obligatorios para la operación. La verificación revisa solamente si los parámetros necesarios existen y si tienen algún valor. No se realiza verificación semántica alguna puesto que solamente se retransmite información.

En el paso 7 se revisa el valor de los parámetros *VERSION* y *SRS*. La validez del resto de los parámetros se deja al servidor *WMS* original.

El paso 8 confirma el valor del parámetro *tiled*.

En el paso 9 el URL recibido se convierte en otro que involucra al *URL* del servidor retransmitido. En el paso 10 se recibe la respuesta en el servidor *CMS*.

El paso 11 devuelve la respuesta al cliente y en este caso, una imagen.

## Imagen del caché

Cuando la solicitud indica que la imagen debe de provenir de un caché, pueden ocurrir dos casos: el primero es cuando la imagen no existe en el caché y debe de generarse a partir de la fuente original; el segundo caso ocurre cuando la imagen existe en el caché y basta con enviarla al cliente.

En caso de que no exista la imagen, se sigue un proceso similar al empleado para solicitar una imagen fresca, con la única diferencia de que antes de retransmitir la imagen al cliente, la imagen se guarda en disco para su uso posterior. Por otra parte, cuando la imagen solicitada existe en disco, solamente se requiere transmitir los datos al cliente.

En ambos casos, es necesario contar con un mecanismo para poder identificar y almacenar cada petición de manera única.

## Organización del Caché de imágenes

La estructura adoptada por la pirámide de mosaicos tiene un impacto considerable en el diseño del sistema, ya que debe permitir el acceso a cada uno de los elementos contenidos, al mismo tiempo que debe reducir el espacio de búsqueda.

La pirámide considerada se forma con mosaicos de distinto tamaño o nivel, donde cada mosaico contiene diferentes imágenes que se colocan dentro de la retícula inducida por el mosaico. Además, cada celda permite colocar imágenes diferentes, que son producto de las variaciones en la petición.

A partir de esta organización se derivan los siguientes requisitos para almacenar una pirámide de mosaicos:

- Debe existir un mecanismo para identificar cada uno de los mosaicos o niveles de la pirámide.
- Debe existir un mecanismo para identificar cada una de las imágenes pertenecientes a un mosaico.
- Debe existir un mecanismo para identificar las variaciones de una imagen del mosaico.

El desarrollo de un sistema que permita disminuir la carga de trabajo impuesta a los servidores WMS, atendiendo solamente la petición de imágenes estáticas a escalas fijas y correspondientes a una retícula, requiere atender correctamente los tres puntos antes expuestos.

Cada mosaico o nivel dentro de la pirámide puede ser identificado a partir de la extensión geográfica de las imágenes que contiene, la cual se calcula a partir del *Bounding Box* de cada petición, empleando las unidades del Sistema de Referencia Espacial o SRS. Por ejemplo, el primer nivel de la pirámide tiene una extensión geográfica de 180 grados, mientras que el segundo tiene una extensión de 90 grados. Para cada uno de los niveles sucesivos, la extensión geográfica de las imágenes posee una extensión de la mitad de tamaño.

Por otra parte, cada mosaico es una retícula rectangular que divide a la tierra en celdas cuadradas, y cada una de éstas puede identificarse por alguna de sus coordenadas extremas. En este caso se emplea la coordenada inferior de la extrema izquierda que se deriva del parámetro *BBOX*.

Por último, una imagen correspondiente a un encuadre geográfico y asociada a un sistema de referencia espacial, cuyo ancho y alto sean fijos, puede generarse de distintas formas dependiendo de los parámetros empleados para la petición. Estas variaciones pueden producirse por el empleo de un estilo distinto, un cambio en el orden de las capas, al formato de la imagen, al color seleccionado como color de fondo transparente, etcétera. Los parámetros que afectan la generación de una imagen se exponen en el Cuadro 4.3.

Parámetro CGI	Descripción
SOURCE	Indica la aplicación que origina la petición.
LAYERS	Lista separada por comas de las capas solicitadas.
STYLES	Cada capa puede ser dibujada con un estilo distinto. El orden de esta lista corresponde al empleado en la lista de capas.
FORMAT	Los posibles formatos para una imagen.
TRANSPARENT	Una imagen puede emplear un fondo transparente.
BGCOLOR	Alguno de los 256 <sup>3</sup> colores empleados como fondo transparente.
TIME	Permite obtener una escena correspondiente a cierta fecha
ELEVATION	Parámetro mutidimensional.

Cuadro 4.3: Parámetros CGI que producen variaciones en una imagen.

En el presente trabajo sólo se consideró el almacenamiento de imágenes estáticas, lo que implica que ciertos parámetros deben restringir sus valores, reduciendo así las posibilidades para generar una imagen. Las restricciones contempladas obedecen a las necesidades de los clientes *WMS* empleados para la visualización de imágenes, los cuales deben sobreponer capas de información provenientes de distintos servidores. Las restricciones de cada parámetro se resumen en el Cuadro 4.4.

Este conjunto de restricciones afecta de manera directa el contenido del caché y la operación del sistema implementado, ya que cuando se recibe una petición solicitando una imagen, pero ésta incluye parámetros *CGI* no permitidos, significa que la imagen (1) no puede existir en el almacén; (2) debe proceder directamente de la fuente original y (3) no debe guardarse para su uso posterior.

Establecidas las restricciones para almacenar una imagen en caché, se deriva un conjunto de elementos empleados para identificar cada imagen. Tales elementos sirven para generar una cadena que identifica al archivo correspondiente dentro del sistema de archivos. A esta cadena se anexa

Parámetro CGI	Restricción
SOURCE	N/A
LAYERS	N/A
STYLES	Se omite el uso de estilos
FORMAT	N/A
TRANSPARENT	N/A
BGCOLOR	Se restringe al valor 0xFFFFFFFF
TIME	Se omite el uso de este parámetro
ELEVATION	Se omite el uso de este parámetro

Cuadro 4.4: Parámetros CGI restringidos

información para identificar el nivel y la celda del mosaico. El conjunto completo de parámetros derivados se observa en el Cuadro 4.5.

Parámetros Derivados	Descripción
SOURCE	Indica la aplicación que origina la petición
LAYERS	Lista separada por guiones de las capas solicitadas
Ancho del mosaico	Identifica el nivel dentro de la pirámide y se obtiene a partir del <i>Bounding Box</i>
Origen del mosaico en el eje X	Longitud mínima en X
Origen del mosaico en el eje Y	Latitud mínima en Y
FORMAT	Formato de la imagen
TRANSPARENT	Indica el color empleado como transparente.

Cuadro 4.5: Parámetros empleados para identificar una petición

Los parámetros obtenidos se usan para crear una estructura jerárquica de directorios, que sirve para organizar y almacenar las imágenes del caché. Esta estructura se forma con los elementos citados en el Cuadro 4.6 y se conforma tanto de nombres de directorios, como de partes del nombre de archivo.

En la Figura 4.6, podemos observar el *URL* junto con sus parámetros *CGI*, solicitando una imagen al *CMS*. Esta petición solicita un mapa con las capas de “estados” y “municipios”, para la aplicación *OpenLayers* y desea que el mapa provenga del caché de imágenes.

Elemento	Tipo	Observaciones
< <i>Source</i> >	Directorio	Aplicación que origina la petición
< <i>NombreDelServicio</i> >	Directorio	
< <i>ListaDeCapas</i> >	Directorio	Se reemplazan las comas por guiones
< <i>AnchoDelMosaico</i> >	Directorio	
< <i>OrigenY</i> >	Directorio	
< <i>OrigenX</i> >	Cadena	
< <i>Transparente</i> >	Cadena	T es transparente, O de opaco
< <i>Formato</i> >	Cadena	Extension que denota el formato de archivo

Cuadro 4.6: Elementos actuales para identificar una petición

```
http://host/CMS/NivelNacional?request=getMap & service=wms & version=1.1.1
& BBOX=-100.0,20.0,-90.0,30.0 & layers=estados,municipios & format=PNG &
& TRANSPARENT=true & width=256 & height=256
& source=OpenLayers & tiled=true
```

Figura 4.6: Petición HTTP Get solicitando una imagen proveniente del caché al CMS.

La ruta correspondiente dentro del sistema de archivos que identifica a la imagen solicitada tiene la siguiente forma:

```
NivelNacional\estados_municipios\10.0\100.0\20.0_T.PNG
```

la cual se compone de:

- Nombre del servicio: “NivelNacional”.

- La lista de capas separadas entre sí por guiones bajos: “estados\_municipios”.
- El ancho del mosaico: “10.0” grados.
- La coordenada extrema izquierda: “100.0” grados.
- La coordenada extrema inferior: “20.0” grados.
- La transparencia de la imagen: “T”.
- La extensión correspondiente al formato de imagen “PNG”.

## Diagrama de flujo

La implementación de la operación *getMap* puede resumirse en el diagrama de la Figura 4.7. Dicho diagrama contempla el caso cuando se solicita una imagen fresca, así como el caso cuando la imagen debe provenir del caché en disco.

Se observa que el sistema devuelve dos tipos de error<sup>7</sup> distintos, dependiendo de la situación. El primero de ellos ocurre si el *URL* solicita algún recurso inexistente o posee elementos no permitidos por el *URL*. Esto se indica en el diagrama como *Error HTTP*. El segundo tipo de error aparece en el dominio del servicio Web y se notifica como un documento *XML*, correspondiendo a alguno de los tipos solicitados y permitidos por la especificación *WMS*. Dentro del diagrama, los errores notificados como *XML* aparecen como *Error XML*.

## Consideraciones sobre el sistema de archivos

El mecanismo de organización y almacenamiento de las imágenes en el presente trabajo implica guardar una gran cantidad de información, empleando nombres largos de archivo y directorios. Para este fin, es necesario que el sistema de archivos empleado cumpla con los siguientes requisitos:

- Permitir el empleo de nombres de directorio largos.

---

<sup>7</sup>Véase sección 4.2.5, acerca del manejo de errores.

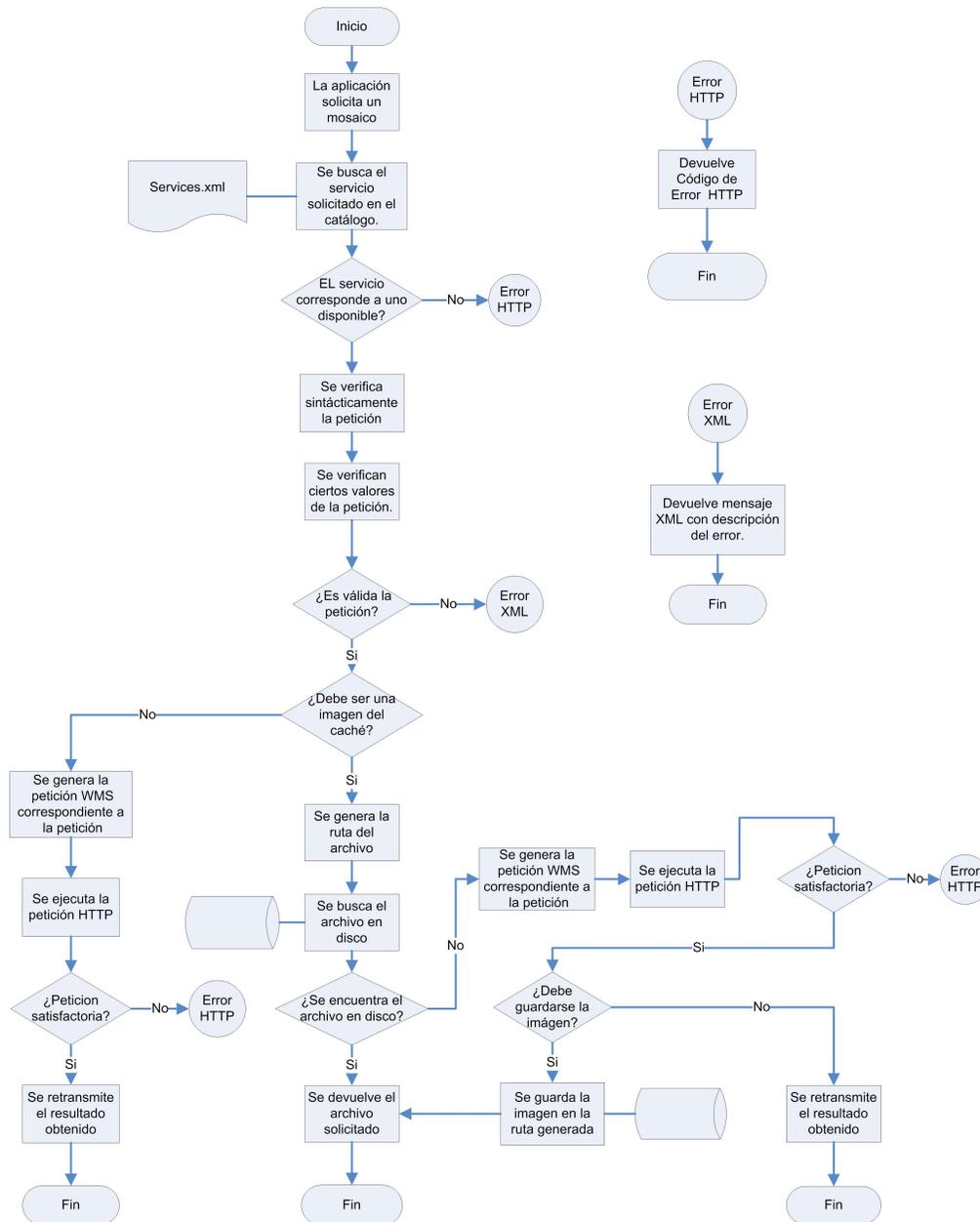


Figura 4.7: Diagrama de flujo de la operación *getMap*

- Permitir el empleo de nombres de archivo largos.
- Permitir el empleo de rutas largas.
- Permitir el uso arbitrario de caracteres, es decir, permitir el uso de números, guiones, etcétera.
- Permitir guardar un número elevado de archivos.

La unidad empleada para almacenar la información dentro de los sistemas de archivos considerados es el archivo<sup>8</sup> y a la fecha de creación del presente trabajo, los sistemas de archivos más empleados son *NTFS*<sup>9</sup> y *EXT2*<sup>10</sup>. Estos sistemas de archivos difieren en su implementación y capacidades, siendo aptos para el presente desarrollo, debido a que permiten el uso de nombres de archivos y directorios con un máximo de 255 caracteres. Adicionalmente, permiten el uso de cualquier carácter<sup>11</sup> para escribir los nombres de directorios o archivos.

En la Tabla 4.7 se presenta un resumen del caso de *NTFS* con las de las características importantes para poder implementar el sistema de caché en dicho sistema de archivos[Mic03].

Descripción	Límite
Tamaño máximo del archivo	16 terabytes menos 64KB
Tamaño máximo del volumen	256 terabytes menos 64KB
Archivos por volumen	4,294,967,295
Longitud máxima para nombre de archivo	256 caracteres
Longitud máxima para nombre de directorio	256 caracteres
Longitud máxima de ruta de archivo	32,768 caracteres,

Cuadro 4.7: Resumen de características básicas de almacenamiento para NTFS.

El sistema de archivos *EXT2* permite el uso de archivos comunes, directorios, archivos especiales y ligas simbólicas. Las características importantes para una correcta implementación del sistema

<sup>8</sup>Dentro de los sistemas de archivos empleados, un directorio también se considera un archivo.

<sup>9</sup>Siglas de *New Technolog File System*, el cual es un sistema de archivos empleado específicamente para Windows NT

<sup>10</sup>Siglas de *Second Extended FileSystem*, el cual es un sistema de archivos para *GNU LINUX*

<sup>11</sup>En cada caso existen restricciones para algunos caracteres, pero no son de importancia al esquema utilizado para nombrar las cosas.

de caché de imágenes se presentan en el Cuadro 4.8.

Descripción	Límite
Tamaño máximo del archivo	2 terabytes
Tamaño máximo del volumen	16 terabytes
Archivos por volumen	4,294,967,295
Longitud máxima para nombre de archivo	255 caracteres
Longitud máxima para nombre de directorio	255 caracteres
Longitud máxima de ruta de archivo	Limitado por el numero máximo de i-nodos
Número máximo de archivos	10 <sup>18</sup>

Cuadro 4.8: Resumen de características básicas de almacenamiento para EXT2.

#### 4.2.4. *GetFeatureInfo*: Obtener información sobre rasgos presentes en el mapa

La operación *getFeatureInfo* está diseñada para proporcionar a los clientes *WMS* información acerca de los rasgos que aparecen dentro de los mapas previamente devueltos. Dichos rasgos se encuentran representados como pixeles dentro del mapa y la operación está permitida para todas aquellas capas que tengan especificado dentro del documento *Capabilities* el atributo *queryable="1"*.

La operación *getFeatureInfo* requiere de los parámetros espaciales empleados para solicitar un mapa por medio de la operación *GetMap*. Esta dinámica plantea la necesidad de un contexto previo a la solicitud de información y aunque es común que ésta sea precedida por la solicitud de un mapa, no es completamente necesario.

#### Parámetros de entrada

Los parámetros espaciales de esta operación son semejantes a aquellos empleados para la operación *getMap*. En el caso de los demás parámetros, cambia el el tipo de petición, el formato de imagen está ausente, la lista de capas se modifica y se requiere la coordenada del pixel en cuestión. Los parámetros de la operación y su descripción se encuentran en el Cuadro 4.9.

<b>Elemento</b>	<b>Descripción</b>
<i>VERSION</i>	Versión solicitada del servicio.
<i>REQUEST</i>	Tipo de operación, en este caso: <i>getFeatureInfo</i> .
<i>QUERY_LAYERS</i>	Lista de capas a las cuales se solicita información.
<i>INFO_FORMAT</i>	Formato de la información obtenida.
<i>FEATURE_COUNT</i>	Límite de registros obtenidos.
<i>X</i>	Coordenada X del pixel.
<i>Y</i>	Coordenada Y del pixel.

Cuadro 4.9: Elementos de una petición *getFeatureInfo*.

## Implementación

La implementación de esta operación dentro del servidor *CMS* es semejante a las operaciones anteriores, es decir, se trata de una redirección de una petición *WMS* desde el cliente al servidor *WMS* original.

Como parte de la implementación del sistema, se revisa que la petición *WMS* contenga los parámetros obligatorios, obteniendo una respuesta negativa en caso de ausencia de algún parámetro obligatorio. El resto de la verificación sintáctica y semántica es responsabilidad del servidor *WMS* original. En caso de recibir una petición bien formada, ésta se retransmite al servidor *WMS* empleando el catálogo de servicios para encontrar el servicio correspondiente a la petición.

La implementación de este *CMS* no guarda información para esta operación puesto que el objetivo es almacenar imágenes estáticas, cuyo contexto está bien definido y delimitado.

## Conversión de peticiones.

En la creación del catálogo de servicios se contempla una entrada para la operación *GetFeatureInfo*, la cual indica el servicio hacia el cual hay que redirigir la petición entrante.

La conversión se realiza reemplazando únicamente el *URL* del servidor *CMS* por el *URL* especificado dentro del documento *Capabilities* para la operación *getFeatureInfo* correspondiente.

## Salida

El parámetro opcional *INFO\_FORMAT* especifica el formato para obtener la información acerca del rasgo preguntado de regreso. Los formatos posibles para solicitar información están listados en el documento *Capabilities* y la respuesta *HTTP* correspondiente debe de coincidir con el tipo de contenido, es decir, el tipo *MIME* debe ser igual al formato de regreso.

La versión implementada de la especificación no indica un comportamiento concreto ni un formato predeterminado [Inc02] de regreso para la operación *getFeatureInfo*. La decisión sobre el tipo de formato de regreso se deja a consideración de los servidores *WMS* correspondientes.

Esta operación se encuentra en fase de pruebas para obtener experiencia, con la finalidad de establecer mayor rigor en versiones posteriores

## Diagrama de flujo.

La implementación a grandes rasgos de la operación *getFeatureInfo* puede verse en la Figura 4.8. En éste se observa que sólo se trata de una redirección precedida de la revisión de parámetros y de un servicio *WMS* asociado. El tipo de errores considerado es semejante al de las operaciones anteriores, es decir, errores *HTTP* y *XML*.

### 4.2.5. Manejo de errores

Los errores dentro del *CMS* pertenecen a alguno de dos entornos posibles: *HTTP* y *Web Services*. En el primer caso, cuando una petición *HTTP* es inválida, el servicio notifica por medio de mensajes de error *HTTP*.

Los errores *HTTP* considerados se observan en el Cuadro 4.10 y contemplan errores de sintaxis, *URL* inexistentes y errores internos en el servidor.

Adicionalmente, el objeto devuelto por una petición válida al servicio siempre debe de corresponder al tipo solicitado por el parámetro *FORMAT*. El tipo de regreso se especifica en el parámetro

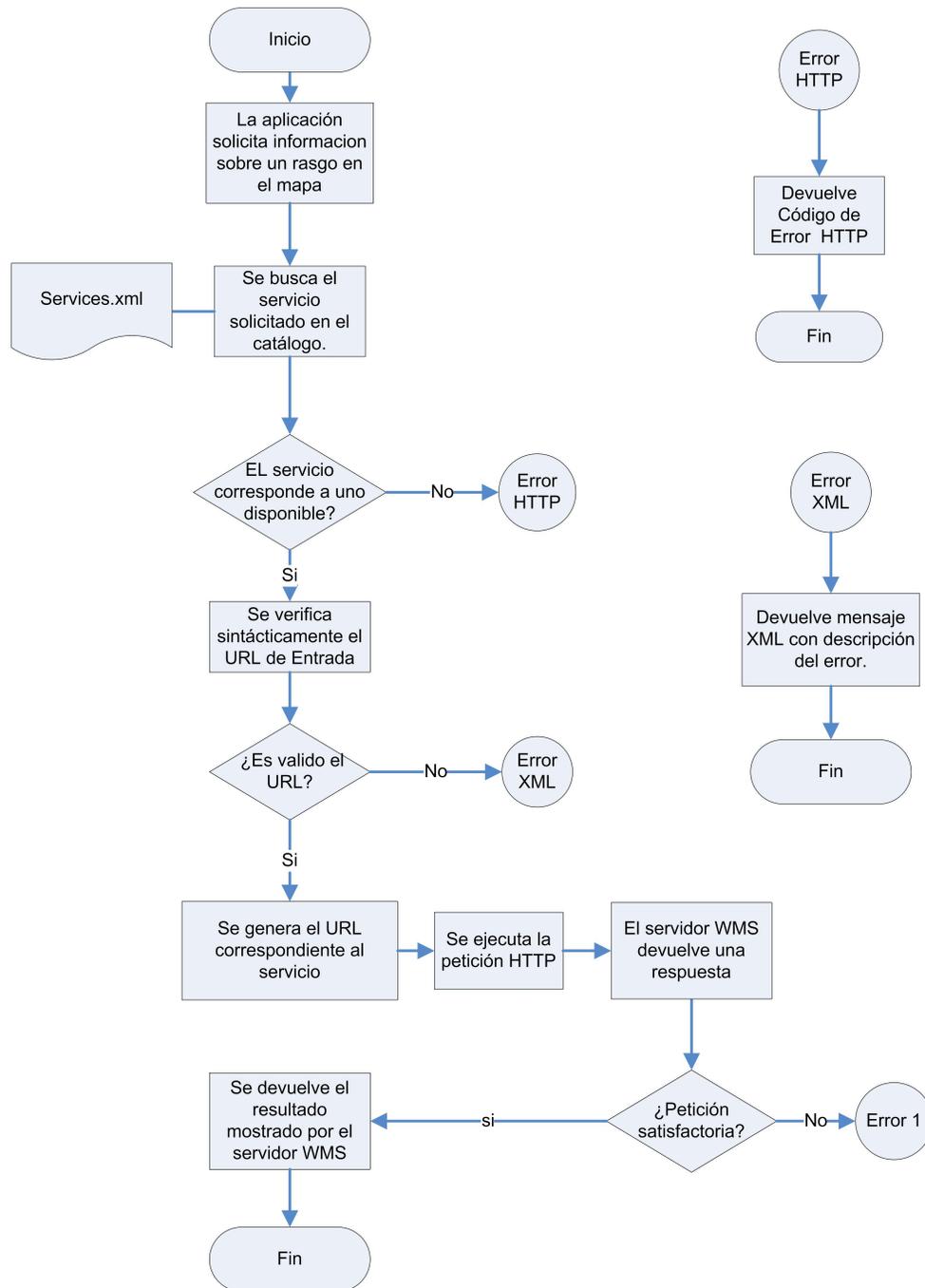


Figura 4.8: Diagrama de flujo de la operación *getFeatureInfo*

Código de error	Descripción
400	URL mal formado.
404	Recurso no existente
500	Error interno del servidor

Cuadro 4.10: Códigos de error HTTP empleados.

*HTTP Content-Type* y debe corresponder al objeto obtenido.

Cuando los errores corresponden a peticiones inválidas dentro de las reglas de los Servicios Web *OGC*, el servicio debe notificar el error empleando un Reporte de Excepción al Servicio (*Service Exception Report*). Este tipo de reportes sirve para notificar a la aplicación cliente, o en su caso al usuario, la razón por la cual la petición es inválida. Los tipos de excepciones deseadas por el cliente se especifican dentro del parámetro *EXCEPTIONS* y el único valor común a todos los servicios Web *OGC* es *application/vnd.ogc.se\_xml*, el cual es un documento *XML* diseñado para reportar excepciones del servicio.

Los reportes de excepción del servicio deben ser válidos según el *DTD* definido en la especificación de *WMS 1.1.1* [Inc02]. Los mensajes de error individuales deben de aparecer dentro del elemento *<ServiceException>* , anidados dentro del nodo *<ServiceExceptionReport>*.

# Capítulo 5

## Evaluación del sistema

Al implementar un sistema Web, surgen preguntas respecto a las características de desempeño del mismo, es decir, la confiabilidad, escalabilidad, capacidad y velocidad que tiene éste asociada a una carga de trabajo. La implementación propuesta del Servidor de Mapas en Cascada busca proporcionar beneficios tanto para servidores *WMS* involucrados, como para las aplicaciones cliente que consumen mosaicos de imágenes organizados en pirámides.

La evaluación del desempeño de un sistema usualmente se realiza para: [MFB<sup>+</sup>07]

- Descubrir la disposición del sistema para un ambiente de producción.
- Conocer si cumple o no con los requerimientos que llevaron a su desarrollo.
- Evaluar las características del sistema en diferentes equipos y configuraciones.
- Encontrar la fuente de los problemas de desempeño o cuellos de botella.
- Afinar el desempeño del sistema.
- Obtener el desempeño real del sistema.

En conjunto, estos elementos permiten conocer la calidad del sistema desarrollado, así como los requerimientos que deben cubrirse para su utilización en el mundo real.

A continuación se definen cada uno de los términos usados en las siguientes páginas:

- **Métricas:** son las mediciones obtenidas durante la ejecución de las pruebas de desempeño expresadas en alguna escala previamente conocida.
- **Capacidad de un sistema:** se refiere a la carga de trabajo total que un sistema puede soportar sin violar sus criterios básicos de desempeño.
- **Latencia:** es la medida de responsividad, la cual representa el tiempo necesario para completar la ejecución de una petición. Usualmente representa la suma de las diversas subtareas realizadas para responder a una petición.
- **Desempeño:** se refiere a la información obtenida acerca del tiempo de respuesta, tasa de procesamiento , número de transacciones por segundo y a la utilización de recursos.
- **Prueba de desempeño:** es una investigación realizada para determinar o validar la velocidad, la escalabilidad y la estabilidad de un determinado producto bajo ciertas condiciones de prueba.
- **Metas de desempeño:** es el conjunto de directivas o estándares que se pretenden alcanzar antes de liberar un producto.
- **Tasa de procesamiento:** es la velocidad con la que cierto trabajo se realiza. Dentro del contexto del *WMS*, la medida de desempeño se estima por el número de peticiones por segundo que se atienden.
- **Objetivos de desempeño:** son los objetivos especificados en término del tiempo de respuesta, la tasa de procesamiento y la utilización de recursos.
- **Umbral de desempeño:** es el conjunto de valores máximos para las métricas obtenidas en las pruebas. Normalmente se expresan en tiempo de respuesta, tasa de procesamiento y utilización de recursos del sistema.

- **Utilización de recursos:** es el costo del sistema en términos de utilización de recursos. Normalmente los recursos son el procesador, la memoria, la actividad en disco y el uso de la red.
- **Utilización:** indica la fracción de tiempo que un dispositivo está ocupado atendiendo a la petición y usualmente se reporta como un porcentaje de la unidad de tiempo.
- **Tiempo de servicio:** es el tiempo requerido por un dispositivo (o servidor) para atender una petición.
- **Tiempo de espera:** es la cantidad de tiempo que un proceso espera para ser atendido por el servicio. Usualmente se asocia con la cola de peticiones a nivel aplicación, a nivel procesador, disco, etcétera.
- **Tiempo de respuesta:** es el tiempo acumulado, asociado al procesamiento de la petición (tiempo de servicio) y con la espera por el servicio (tiempo de espera).
- **Saturación:** es el punto en el cual un sistema alcanza su máxima utilización.
- **Escalabilidad:** es la capacidad que tiene una aplicación para manejar una carga de trabajo adicional sin afectar adversamente el desempeño. Esto puede realizarse aumentando la capacidad de cada uno de los recursos como el procesador, la memoria, la capacidad de almacenamiento o la conectividad en red.
- **Carga de trabajo:** es el estímulo aplicado a un sistema, aplicación o componente para simular el patrón de uso con respecto a la información de entrada. La carga de trabajo incluye el número total de usuarios concurrentes, el volumen de datos, el volumen de transacciones, junto con la mezcla de éstas. Para modelar el desempeño se asocia una carga de trabajo con un escenario individual.

## 5.1. Contexto del proyecto

La posibilidad de poder distribuir información geográfica eficientemente, trae muchos beneficios a los actores involucrados. Para quienes albergan la información, significa destinar una cantidad menor de recursos para brindar acceso a la información geográfica, al tiempo que aumenta el valor de ésta. Para los usuarios finales representa la posibilidad de tener acceso inmediato, oportuno y eficiente a la información necesaria para alcanzar sus objetivos.

El objetivo particular del desarrollo del *CMS* es disminuir la carga de trabajo impuesta al servidor de mapas al evitar la ejecución de los cálculos involucrados para la generación de imágenes, debido al almacenamiento de las peticiones previas. Tal disminución en la carga de trabajo trae consigo una disminución en el tiempo de servicio, una mejora en la disponibilidad del sistema y, por consiguiente, un incremento en el número de transacciones realizadas.

Los beneficios directos para un cliente *WMS* que consuma imágenes en mosaicos y aproveche el *CMS*, se traducen en un menor tiempo de espera por cada petición realizada y, por lo tanto, en un menor tiempo de espera para desplegar el mapa final al usuario.

## 5.2. Ambiente de evaluación

El ambiente de prueba considerado coincide en gran parte con el ambiente de producción, tanto en su parte física como en su parte lógica; sin embargo, para realizar las pruebas se emplean herramientas de prueba y monitoreo que introducen diferencias en ambos ambientes.

### Herramientas

Para la evaluación del sistema se emplean dos tipos de herramientas:

- Herramientas de generación de carga.
- Herramientas de monitoreo.

Para simular la carga de trabajo es necesario recrear el comportamiento de los clientes empleados, lo cual se consigue al almacenar y reproducir el conjunto de peticiones recibidas por el *CMS*. Dado que estas peticiones se realizan bajo condiciones de trabajo reales, la generación depende del tipo de actividad de los usuarios.

Para probar el sistema, las peticiones se envían en el orden en el cual las recibe el servidor, descargando las imágenes en memoria para evitar el cuello de botella impuesto por el sistema de almacenamiento.<sup>1</sup>

Para simular la carga de trabajo y monitorear el desempeño del servidor *CMS* se emplea una herramienta que llamada *Apache JMeter*[jme08], la cual es una aplicación escrita completamente en Java, la cual está diseñada para realizar pruebas y medir el desempeño de recursos Web estáticos y dinámicos (archivos, servlets, guiones de Perl, objetos de Java, conexiones a bases de datos, consultas y servidores *FTP*, entre otros).

## Factores externos

Debido a que el ambiente de pruebas considerado coincide con el de producción, existe una carga de trabajo adicional que corresponde a las peticiones generadas por usuarios reales, así como por aplicaciones ajenas a este ambiente.

## Consideraciones sobre el ambiente de pruebas

Es necesario resaltar que parte del desempeño de las aplicaciones Web depende de diversos factores. Uno de ellos se relaciona con las características de la conectividad en red, que en caso de tratarse de redes locales pueden poseer una velocidad de transferencia elevada, que varía entre 100Mbps y 1Gbps. Cuando ejecutamos el sistema a través de Internet para probar el desempeño del mismo, la velocidad general de la red es considerablemente menor y está limitada por diversos

---

<sup>1</sup>El número de usuarios concurrentes es la única variable considerada y se hace variar para simular diferentes cargas de trabajo, calculando así el desempeño del sistema en función del número de usuarios concurrentes.

factores independientes del sistema, los cuales son aleatorios y dificultan la evaluación, derivando en cuellos de botella para la transferencia de imágenes y el envío de peticiones.

El entorno actual de red implementa una red local Ethernet<sup>2</sup> cuyo ancho de banda es de 1Gbps, y emplea *TCP/IP*<sup>3</sup> como protocolo de transmisión y comunicación. Realizando cálculos simples es posible estimar, al menos teóricamente, una transmisión máxima de 1200 archivos de 100KB por segundo, aunque en la práctica, la cifra es considerablemente menor.

Si se realizan pruebas a través de Internet, lo cual implica un ancho de banda menor, es necesario considerar factores adicionales como el número de redes que debe atravesar la información, el retraso involucrado por el cambio de redes y la saturación de las mismas, ya que Internet es una red pública que se emplea por millones de usuarios por lo que sus condiciones son variables.

### 5.3. Criterios de evaluación

El empleo de Internet para visualizar mapas en línea impone la necesidad de distribuir grandes volúmenes de información de manera rápida y oportuna. La implementación de un *WMS* es una solución estándar para la distribución de mapas digitales en línea debido a su especificación abierta y precisa. Sin embargo, la generación de imágenes georeferenciadas tiene un costo computacional elevado, ya que las operaciones geométricas empleadas son costosas. En el costo de esta operación también debe considerarse la carga de trabajo impuesta a los sistemas de almacenamiento, debido al gran volumen de datos que deben procesarse para generar una imagen.

La distribución de imágenes georeferenciadas es una necesidad inamovible que debe atenderse sin comprometer los sistemas empleados y es cuando el empleo de tecnologías para el despliegue de mosaicos de imágenes organizadas en pirámides resulta en una solución viable para aliviar la utilización del procesador.

---

<sup>2</sup>Familia de tecnologías de redes de área local y comprende la capa física. Es mejor conocido como el estándar *IEEE 802.3*

<sup>3</sup>Siglas de *Transmission Control Protocol over Internet Protocol*.

Con este antecedente, las soluciones desarrolladas que permitan disminuir el tiempo de respuesta, aumentar la tasa de procesamiento y disminuir el consumo de recursos de los sistemas involucrados resultan en soluciones satisfactorias.

En el ámbito del presente trabajo, el sistema desarrollado cumple con dichos requerimientos, siendo necesario cuantificar la mejora obtenida sobre el empleo exclusivo de un *WMS*. Para cuantificar la mejora en desempeño, se toma la línea de referencia proporcionada por el empleo exclusivo de servidores *WMS* y se plantean las siguientes metas:

- Cuantificar la tasa de procesamiento del sistema *CMS* y compararla con el rendimiento del *WMS*.
- Cuantificar el tiempo de respuesta empleado por el *CMS* y compararlo con el obtenido del *WMS*.
- Cuantificar el consumo de recursos del *CMS* y compararlo con el requerido por un *WMS*.
- Comparar el desempeño del sistema *CMS* con desarrollos semejantes.

Se considera que los resultados de la evaluación serán exitosos cuando:

- El número de transacciones que atiende el sistema sea mayor que el número de transacciones que entrega el *WMS*.
- El tiempo de respuesta obtenido sea menor o equivalente a aquél del *WMS*.
- El nivel de utilización de recursos de procesador y memoria sean menores.

### **Tiempo de respuesta**

Empleando el ambiente de producción, se ha encontrado que los tiempo de respuesta varían dependiendo del tipo de mapa empleado y la cantidad de información involucrada. De manera general, se encuentra que la media en el tiempo de respuesta oscila entre 500ms y 2000ms en casos tipificados [Bri05] empleando únicamente el servidor *WMS*.

Se espera que al evitar la ejecución de las operaciones geométricas, el tiempo de respuesta se reduzca en la mitad, oscilando entre 250ms y 1000ms. Esta suposición ignora la velocidad de transferencia de la red y considera que cualquier aumento en el tiempo de respuesta, supone una degradación en el desempeño del sistema.

## Tasa de procesamiento

El número de peticiones que puede atender un sistema que genera mapas depende del número de recursos disponibles. Se estima que a partir de evaluaciones realizadas en condiciones similares el conjunto de equipo disponible para pruebas y producción puede generar mapas a una tasa máxima de 27000 transacciones por hora (450 imágenes por minuto o 7.5 imágenes por segundo) sin degradar el desempeño[ESR07],[Bri05]. En general, a una mayor cantidad de recursos disponibles, es posible atender un mayor número de peticiones.

Se espera que la tasa de procesamiento aumente de forma considerable al emplear el *CMS*, ya que al suprimir el uso de algoritmos geométricos para producir imágenes se evita el uso de algoritmos cuya complejidad computacional oscila entre  $O(n \log n)$  y  $O(n^3)$ . El conjunto de estructuras de datos y algoritmos que emplea el *CMS*, tienen complejidades en tiempo que oscilan entre  $O(1)$ ,  $O(\log n)$  y  $O(n)$  [CLRS01][Mic03][ext]. Esto permite pensar que el aumento en desempeño debe incrementarse aproximadamente en forma cuadrática.

Existe una herramienta llamada *TileCache* [Sch] semejante a la que se desarrolla en el presente trabajo que comparte la filosofía del mismo y plantea que puede atender al menos un promedio de 300 peticiones por segundo. A pesar de no detallar las condiciones sobre las cuáles se obtiene dicho desempeño, esta cifra sirve de referencia para comparar el desempeño del *CMS* atendiendo el mismo problema.

## Utilización de recursos

La cantidad de recursos computacionales necesarios para generar imágenes geográficas referenciadas es elevada y puede observarse en varios subsistemas, es decir, en la utilización del procesador, en la memoria, en la actividad de disco y la actividad de red. El tipo de algoritmos geométricos y geospaciales empleados por los servidores *WMS* usualmente tienen costos elevados, ya sea en tiempo, espacio o acceso a dispositivos de entrada y salida.

El empleo del *CMS* supone un menor uso del procesador y la memoria, a cambio de una mayor utilización del sistema de archivos y del sistema de almacenamiento.

Si relacionamos la tasa de procesamiento esperada con el tamaño esperado de los archivos<sup>4</sup>, podemos determinar el consumo potencial de memoria que tendría el sistema. Para atender 300 peticiones de aplicación *OpenLayers* se requiere en el caso extremo de 75MB de memoria *RAM*, mientras que para atender 300 peticiones de la aplicación *WorldWind* se requerirían 300MB de memoria *RAM*. Estas cifras suponen que no se emplean algoritmos de compresión de imágenes, aunque en la práctica, las imágenes tienen un tamaño considerablemente menor.

## 5.4. Diseño y planeación de las pruebas

Las aplicaciones Web que solicitan información al sistema poseen un flujo de trabajo específico donde el usuario despliega un mapa de tamaño fijo compuesto de diversas capas, de las cuales al menos una de ellas siempre está activa o visible. Dicha capa funciona como fondo del mapa y se emplea como referencia para el resto de las capas.

Para desplegar un mapa completo es necesario obtener las imágenes que forman cada capa, es decir, se debe realizar un número determinado de peticiones *WMS* para obtener cada una de las imágenes del mosaico.

---

<sup>4</sup>Los archivos de 256 píxeles de ancho y alto tendrían un tamaño máximo de 256KB. Los archivos de 512 píxeles de ancho y alto tendrían un tamaño máximo de 1MB.

## Escenario típico

Las actividad típica de un usuario de un visualizador de mapas comienza por desplegar un mapa inicial, seguido de alguna de las siguientes acciones:

- Realizar un desplazamiento dentro del mapa.
- Realizar un acercamiento o retroceso a un área determinada del mapa.
- Activar o desactivar alguna de las capas presentes en el mapa.

En cualquiera de estos casos, la operación común es solicitar imágenes nuevas, ya sea para mostrar una parte del mapa de la cual no se han descargado imágenes, o bien para mostrar la capa que no se encontraba visible.

Hasta cierto punto es posible determinar el número de imágenes de 256 por 256 píxeles que forman una capa, por medio de la siguiente fórmula:

$$f(\text{width}, \text{height}) = (\lfloor \text{width} \div 256 \rfloor + 2)(\lfloor \text{height} \div 256 \rfloor + 2)$$

En el caso de un mapa cuyas dimensiones son de 800 píxeles de ancho por 600 de alto, se requiere un total de 20 imágenes para desplegar cada capa. Si el mapa se compone de cinco capas, de las cuales tres están activas, para desplegar dicho mapa se requiere descargar aproximadamente 60 imágenes por medio de peticiones *WMS*.

## Estrategia de pruebas

La simulación de los escenarios de usos típicos es una parte importante de las pruebas, ya que la obtención de resultados confiables esta condicionada a la calidad de las simulaciones. Con resultados consistentes, se pueden tomar decisiones acerca del uso del sistema y el desarrollo futuro de éste.

En las pruebas a desarrollar, el ambiente de producción coincide con el ambiente de pruebas, lo cual permite obtener resultados sobre el desempeño del sistema en condiciones reales. En el

ambiente actual, los recursos disponibles en los servidores se comparten con otras aplicaciones necesarias para la propia generación de mapas, con servidores de aplicaciones y en servicios adicionales.

Para poder generar cargas de trabajo reales, en el servidor *CMS* se almacena cada uno de los *URL* que se recibe. Es un requisito para ejecutar las pruebas y evitar variaciones en los resultados que todas las imágenes solicitadas se encuentren almacenadas en el servidor, pero como la carga de trabajo se deriva del conjunto de *URL* recibidos, se considera que esta situación es inexistente.

Para reproducir diversas condiciones de trabajo, la actividad de los usuarios se genera a diferente velocidad para determinar el desempeño del sistema tanto en condiciones esperadas, como en condiciones de estrés<sup>5</sup>.

### 5.4.1. Obtención de métricas

Para conocer el desempeño que otorga el sistema bajo distintas cargas de trabajo, es necesario obtener diversas métricas y controlar las variables involucradas como el número de usuarios concurrentes y el retraso entre cada petición.

El costo computacional de solicitar dos imágenes diferentes es el mismo, siempre que ya se hayan almacenado en caché. Esta igualdad de costos indica que el factor a considerar en la evaluación es el número de imágenes que pueden ser devueltas en una unidad de tiempo, para así poder determinar el número de usuarios potenciales. Este factor, junto con el tiempo de respuesta, determinan en gran medida la capacidad de respuesta del sistema.

## Tiempo de Respuesta

Para conocer el tiempo de respuesta promedio, se emplea la herramienta *JMeter* para simular una carga de trabajo y calcular dicho valor. Esta herramienta funciona del lado del cliente y para cada

---

<sup>5</sup>Se considera una condición de estrés cuando el sistema debe funcionar por encima de las expectativas de producción.

petición que se ejecuta, se obtiene el tiempo correspondiente, promediando el total de peticiones.

El tiempo de respuesta de una petición corresponde a una serie de actividades que debe realizar el servidor y puede clasificarse dentro de alguno de los tres siguientes casos:

1. El primer caso corresponde al tiempo necesario para obtener una imagen a partir de un servidor *WMS* ordinario y sin intervención del *CMS*.

El tiempo de respuesta resultante corresponde al tiempo de respuesta obtenido desde que se genera una petición *WMS* y hasta que dicha petición la resuelve exitosamente el servidor *WMS*. Este resultado que se define como *TR1* es la suma del tiempo empleado para interpretar la petición *WMS*, generar la imagen correspondiente a la petición y transferir el resultado al cliente.

Expresado en notación matemática, tenemos que:

$$TR1 = ParseTimeWMS + ImageGenerationTimeWMS + TransferTimeWMS$$

donde

*ParseTimeWMS* es el tiempo empleado por el *WMS* para interpretar la petición y encaminar la generación de la imagen correspondiente.

*ImageGenerationTimeWMS* es el tiempo empleado por el *WMS* para generar la imagen descrita por los parámetros empleados.

*TransferTimeWMS* es el tiempo requerido para transferir la imagen solicitada al cliente.

2. El segundo caso corresponde al tiempo requerido para solicitar las mismas imágenes que en el caso anterior, pero anteponiendo el *CMS* y solicitándolas por primera vez. Este resultado se define como *TR2* y corresponde al tiempo procesamiento del servidor *WMS* original junto con el tiempo requerido por el *CMS* para interpretar, redirigir la petición y retransmitir la respuesta. Esta relación puede observarse claramente en la siguiente expresión matemática:

$$TR2 = ParseTimeCMS + TR1 + TransferTimeCMS$$

donde *ParseTimeCMS* es el tiempo empleado por el *CMS* para interpretar la petición y encaminar la generación de la imagen correspondiente; *TransferTime CMS* es el tiempo requerido por el sistema para localizar y leer el archivo solicitado en el sistema de archivos. Se observa que el tiempo de respuesta  $TR2 > TR1$  debido al uso del *CMS*.

3. El tercer caso es semejante al anterior pero supone que las imágenes ya han sido almacenadas debido a peticiones previas *WMS* y pueden ser devueltas inmediatamente tras su solicitud. El tiempo de respuesta *TR3* corresponde al tiempo involucrado en procesar los parámetros de la petición y buscar la imagen almacenada en el sistema de archivos, es decir:

$$TR3 = ParseTimeCMS + SearchTimeCMS + TransferTimeCMS$$

donde *SearchTimeCMS* es el tiempo necesario para encontrar el archivo en el sistema de archivos y leer su contenido, justo antes de devolver la imagen al cliente.

## Tasa de procesamiento

La tasa de procesamiento del sistema es una métrica que se calcula del lado del cliente empleando la herramienta de generación de carga *JMeter*. Esta cifra se obtiene de dividir el número total de peticiones realizadas entre la duración de la prueba, obteniendo así el número de peticiones atendidas por unidad de tiempo.

Debido a que las operaciones realizadas por el *CMS* tienen un costo constante, el número de peticiones que el sistema puede atender está condicionado a factores externos como las estructuras de datos empleadas y algoritmos implementados tanto en el sistema de archivos como por los sistemas de almacenamiento.

El tiempo de búsqueda necesario para buscar un archivo dentro del sistema de archivos *NTFS* es  $O(\log n)$  [Mic03]. En el caso de *EXT2*, la implementación no define el tipo de estructura de datos utilizada, pero se considera que el peor caso emplea una estructura de tipo lineal.

Por otra parte, el sistema de almacenamiento empleado determina el desempeño en tiempo real del sistema, es decir, existe un tiempo de búsqueda y una tasa de transferencia asociada a cada sistema de almacenamiento. En este caso, se considera importante conocer la tasa de transferencia cuando ocurren lecturas aleatorias y archivos de tamaño variable.

En conjunto, toda esta información acerca del sistema de archivos y del sistema de almacenamiento es necesaria para estimar el desempeño esperado del *CMS* y en su caso, atribuir la baja del desempeño a alguno de estos factores.

## Utilización de recursos

Para supervisar el consumo de recursos empleados por el *CMS*, se emplea *JMeter* en un modo especial para monitorear sistemas remotos, empleando las funcionalidades del contenedor *Tomcat*. Esta herramienta permite conocer de manera visual y generar un reporte de la cantidad de memoria empleada, la carga de trabajo y el número de peticiones en espera cada cierto intervalo de tiempo.

En este caso es necesario considerar la carga de trabajo, así como el consumo inicial de recursos del *CMS* para poder determinar en términos de porcentaje el nivel de incidencias que produce el *CMS*.

## Pruebas

Para la realización de las pruebas se cuenta con varios conjuntos de datos que contienen una serie de peticiones *WMS* realizadas al sistema. Para cada aplicación considerada, se crea un archivo con una cantidad significativa<sup>6</sup> de peticiones que permita simular la carga de trabajo.

Para cada una de las aplicaciones, es necesario obtener la línea base de desempeño empleando únicamente servidores *WMS*, estableciendo tiempos de respuesta, rendimiento y carga de trabajo. A partir de una línea base de desempeño, es posible saber si las hipótesis de desempeño acerca del *WMS* son acertadas o erróneas.

---

<sup>6</sup>Se emplean archivos con más de veinte mil peticiones *WMS*.

Cada prueba se realiza simulando cierta cantidad de usuarios que generan peticiones *WMS*, los cuales realizan una sola petición a la vez, es decir, un usuario no envía una nueva petición hasta que recibe la anterior. De este modo, para simular suficientes peticiones concurrentes es necesario crear un mayor número de usuarios que generen suficientes peticiones de manera independiente.

El conjunto total de pruebas se describe a continuación:

1. La primera prueba busca obtener una línea base de desempeño al realizar peticiones únicamente a los servidores *WMS* con la aplicación que utiliza el *API OpenLayers*. La prueba se realiza simulando varios usuarios, comenzando con dos usuarios e incrementando su número en 4, 8, 10, 12, 16 y 20 usuarios. Para cada incremento en el número de usuarios se realiza la misma prueba durante dos minutos. En cada prueba se obtienen los tiempos de respuesta, la tasa de procesamiento, el consumo de memoria y la carga de trabajo del servidor.
2. La segunda prueba busca obtener una línea base de desempeño al realizar peticiones a los servidores *WMS* con la aplicación *NASA WorldWind*. La prueba simula varios usuarios, comenzando con dos e incrementando su número en 4, 8, 10, 12, 16 y 20. Para cada incremento en el número de usuarios se realiza la misma prueba durante dos minutos. En cada prueba se obtienen los tiempos de respuesta, la tasa de procesamiento, el consumo de memoria y la carga de trabajo del servidor.
3. La tercera prueba busca obtener el desempeño al utilizar el servidor *CMS* en sustitución del *WMS*, al realizar peticiones con la aplicación que utiliza el *API OpenLayers*. Esta prueba busca alcanzar el doble de la tasa de procesamiento esperada de 300 peticiones por segundo. La prueba se realiza simulando varios usuarios que generan peticiones *CMS*, comenzando con diez usuarios e incrementando su número hasta alcanzar los 60. Cada nueva prueba incrementa en diez la cantidad de usuarios de la simulación y cada prueba tiene una duración de dos minutos. En cada prueba se obtienen los tiempos de respuesta, la tasa de procesamiento, el consumo de memoria y la carga de trabajo del servidor.

4. La cuarta prueba busca obtener el desempeño al utilizar el servidor *CMS* en sustitución del *WMS*, al realizar peticiones con la aplicación *NASA WorldWind*. Esta prueba busca alcanzar el doble de la tasa de procesamiento esperada de 300 peticiones por segundo.

La prueba se realiza simulando varios usuarios que generan peticiones *CMS*, comenzando con diez usuarios e incrementando su número hasta alcanzar los 60. Cada nueva prueba incrementa en diez el número de usuarios de la simulación y cada una tiene una duración de dos minutos. En cada prueba se obtienen los tiempos de respuesta, la tasa de procesamiento, el consumo de memoria y la carga de trabajo del servidor.

## 5.5. Configuración del ambiente de pruebas

### Servidor de mapas

La generación de mapas requiere de herramientas especializadas para obtener imágenes georeferenciadas a partir de información geoespacial.

La herramienta utilizada para generar mapas se llama *ArcIMS* (*Arc Internet Map Server*), la cual permite crear servicios web, así como publicar mapas web dinámicos. *ArcIMS* es conocido por su capacidad para proporcionar soluciones escalables y de alto desempeño para desplegar mapas a través de Internet. Actualmente es la herramienta de mayor uso y aceptación para la visualización y despliegue de cartografía en línea.

*ArcIMS* es compatible con los estándares propuestos por la *OGC* (*WMS* y *WFS* entre otros), además permite que aplicaciones que operen con dichos estándares, puedan consumir los mapas creados con esta herramienta. Los clientes potenciales de *ArcIMS* pueden ser aplicaciones de escritorio, aplicaciones de servidor, servicios web y en general, todo programa que pueda comunicarse a través del estándar *OGC WMS*.

## Mapas

Para la evaluación del sistema se crean tres mapas con características diferentes que abarcan gran cantidad de casos posibles. Estos mapas van desde casos simples donde la cantidad de información y las opciones de visualización son limitadas, hasta casos complejos donde la cantidad de información es grande y las opciones de visualización complejas.

Los mapas elaborados para la evaluación del sistema se describen a continuación:

1. El primer mapa representa el conjunto de información vectorial a nivel nacional con una escala de 1:250,000 correspondiente a México. Tal subconjunto se compone por capas de áreas urbanas, cuerpos de agua, hidrología, vías de comunicación, división estatal, división municipal, áreas naturales protegidas, islas, áreas geoestadísticas básicas de tipo urbano y capas de etiquetas.

En cada uno de los casos, los datos se clasifican y despliegan acorde a su categoría. En caso de ser rasgos con nombre, se muestra la etiqueta correspondiente dicho mapa. En la Figura 5.1 se puede observar una imagen del mapa.

2. El segundo mapa contiene dos conjuntos de imágenes raster. El primero es un mosaico de imágenes de satélite con cobertura nacional cuya resolución es de 150 metros por pixel. Un ejemplo se muestra en la Figura 5.2. El segundo conjunto se forma por el total de ortofotos digitales que cubren la totalidad del territorio de México y consta de más de trece mil imágenes georeferenciadas en formato *JPEG*<sup>7</sup> Las imágenes cuentan con una calidad del 95 por ciento y cada una posee una resolución de 4000 pixeles de ancho por 4500 pixeles de alto. Su tamaño en disco oscila entre 4 y 5 MB. El espacio requerido para almacenar todas las imágenes es aproximadamente de 70 GB. En la Figura 5.3 se presenta el detalle de una ortofoto.

---

<sup>7</sup>Joint Photographic Expert Group es un formato de imagen que emplea compresión con pérdidas y almacena tres bandas dentro del espectro visible.

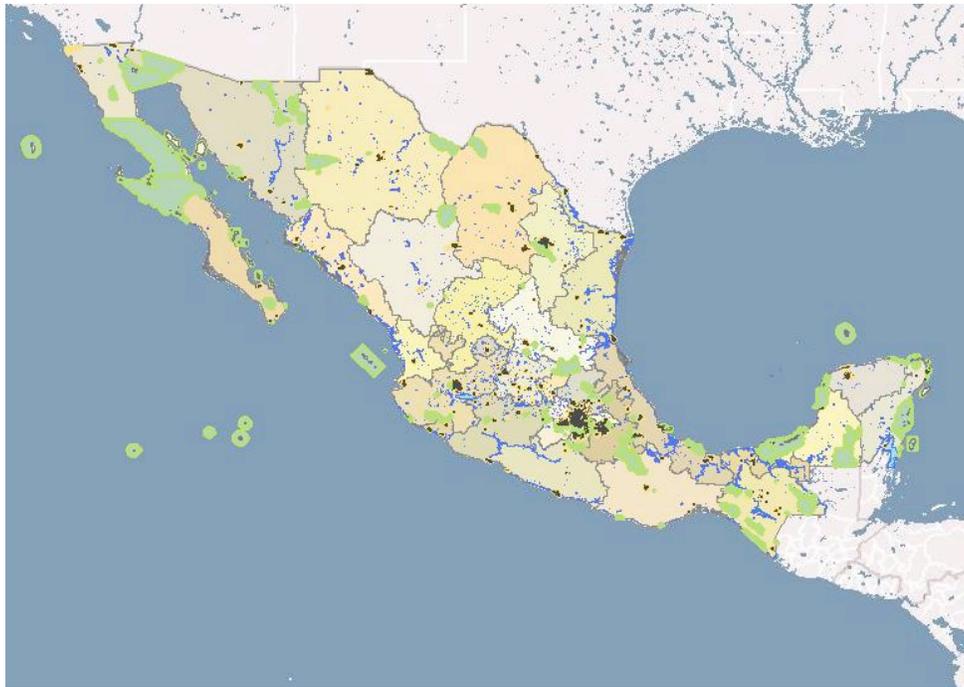


Figura 5.1: Mapa a nivel nacional escala 1:250,000

3. El tercer mapa representa un conjunto de información en formato vectorial con una cobertura a nivel nacional y tiene una escala 1:1,000,000 correspondiente a México. Tal subconjunto se compone por capas de información temática y topográfica. El conjunto de mapas temáticos comprende provincias y subprovincias fisiográficas, uso de suelo y vegetación. Las capas adicionales representan áreas urbanas, cuerpos de agua, corrientes de agua, carreteras, curvas de nivel, áreas de vegetación densa, áreas naturales protegidas, islas, agebs urbanos, entradas a grutas, líneas de transmisión, rutas de embarcación, retículas de un grado y gradículas, acueductos, etcétera. En la Figura 5.4 se puede observar una imagen del mapa.

## Clientes WMS

Las aplicaciones cliente empleadas para evaluar el desempeño del *CMS* tienen un comportamiento activo, ya que solicitan una gran cantidad de imágenes de forma continua y por



Figura 5.2: Mapa a Nivel Nacional de Imágenes Raster



Figura 5.3: Detalle de ortofoto de Ciudad Universitaria.

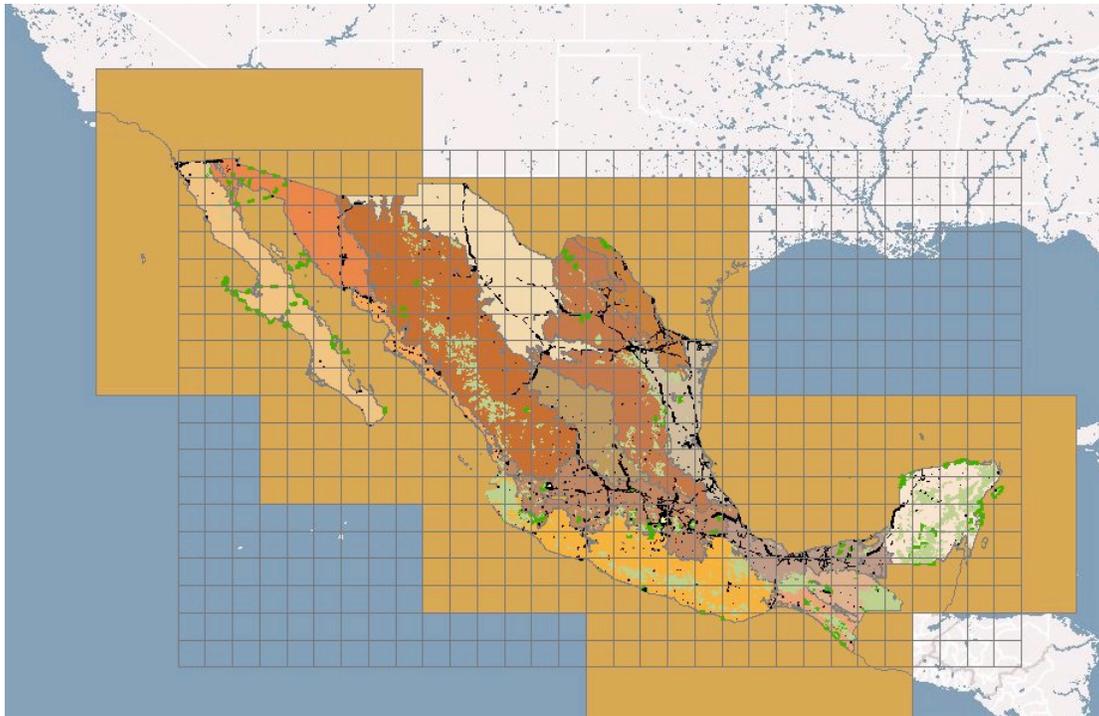


Figura 5.4: Mapa de México a nivel nacional escala 1:1,000,000

consiguiente consumen un gran volumen de datos para desplegar mapas con información acerca del planeta Tierra. Las aplicaciones cliente empleadas son las siguientes:

- *NASA WorldWind* es una aplicación de código abierto que permite visualizar en tres dimensiones cualquier región de la tierra, mostrando imágenes satelitales Landsat y datos de relieve proveniente del *SRTM*.<sup>8</sup> Esta aplicación puede consumir información proveniente de servidores *WMS* y desplegarla sobre el planeta, siempre que los datos empleen la proyección *WGS84*.<sup>9</sup>
- *OpenLayers API* es una biblioteca que permite desplegar mapas dentro de navegadores Web sin emplear código del lado del servidor. El *API* implementa métodos estandarizados para acceder a la información geográfica, entre los que se encuentran los estándares

<sup>8</sup>*Shuttle Radar Topography Mission*

<sup>9</sup>*World Geodetic System 1984*

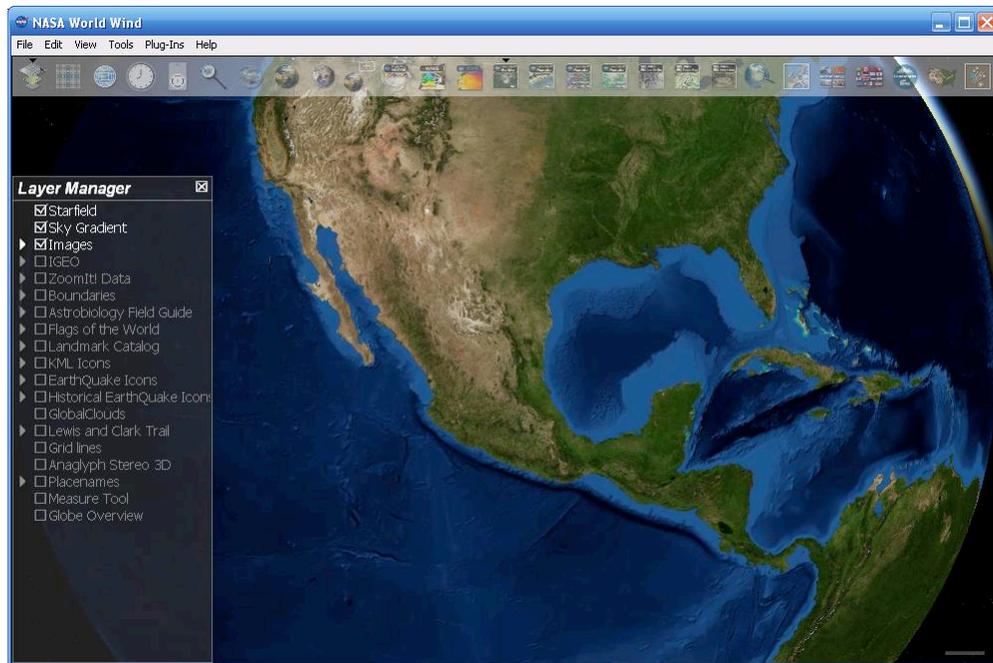


Figura 5.5: Aplicación *NASA WorldWind*.

*OGC WMS* y *WFS*. El *API Open Layers* es desarrollado por completo usando el lenguaje *JavaScript* con orientación a objetos.

Cada una de estas aplicaciones emplea parámetros específicos que se exponen en la Tabla 5.2 y 5.1 y para *NASA WorldWind* y en la Tabla 5.3 para las aplicaciones desarrolladas con el *API OpenLayers*.

### 5.5.1. Implementación de las pruebas

#### Línea base de desempeño empleando WMS

Para obtener el desempeño base de los servidores *WMS*, se configura la herramienta *JMeter* para simular la carga de trabajo de la siguiente forma:

Parámetro	Valor	Descripción
FORMAT	image/png	Cadena <i>MIME</i> para el formato <i>PNG</i>
WIDTH	512	Ancho en pixeles
HEIGHT	512	Alto en pixeles
SRS	EPSG:4326	Coordenadas geográficas
TRANSPARENT	true	Emplea fondo transparente
BGCOLOR	0xFFFFFFFF	Color blanco definido como transparente
STYLES	default	Emplea el estilo predeterminado

Cuadro 5.1: Descripción de los parámetros para *NASA WorldWind* en caso de imágenes en formato *PNG* (Portable Network Graphics).

Parámetro	Valor	Descripción
FORMAT	image/jpeg	Cadena <i>MIME</i> para formato <i>JPEG</i>
WIDTH	512	Ancho en pixeles
HEIGHT	512	Alto en pixeles
SRS	EPSG:4326	Coordenadas geográficas
TRANSPARENT	true	Emplea fondo transparente
BGCOLOR	0xFFFFFFFF	Color blanco definido como transparente
STYLES	default	Emplea el estilo predeterminado

Cuadro 5.2: Descripción de los parámetros para *WorldWind* en caso de imágenes en formato *JPG*.

Parámetro	Valor	Descripción
FORMAT	PNG o JPEG	Formato de imagen empleado.
WIDTH	256	Ancho en pixeles
HEIGHT	256	Alto en pixeles
SRS	EPSG:4326	Coordenadas Geográficas
TRANSPARENT	true	Emplea fondo transparente
BGCOLOR	0xFFFFFFFF	Color blanco definido como transparente
STYLES	default	Emplea el estilo predeterminado

Cuadro 5.3: Descripción de los parámetros para *OpenLayers*

- Se crea un hilo de ejecución por usuario<sup>10</sup>, dentro del cual solamente se puede producir una petición a la vez.
- Se introduce un retraso entre las peticiones de 90 ms por usuario.
- Los parámetros de cada petición *WMS* se obtienen de un archivo que todos los usuarios comparten y consumen, sin embargo cada usuario lee parámetros diferentes.
- Las pruebas se realizan incrementando el número de usuarios, comenzando con dos y posteriormente aumenta a 4, 8, 12, 16 y 20 usuarios en cada prueba.
- En cada prueba se emplea un componente de la herramienta llamado Informe Agregado, que sirve para mostrar de forma sintética el número total de peticiones, la media y la desviación estándar del tiempo de respuesta, así como el tiempo mínimo y máximo de respuesta, el porcentaje de errores, el rendimiento en peticiones por segundo y la tasa de transferencia promedio en bytes.
- Se monitorea visualmente el consumo de recursos de los servidores empleados.
- La duración de cada prueba es de 120 segundos.

## Pruebas de desempeño empleando CMS

Para las pruebas que emplean el *CMS* se configura la herramienta *JMeter* para simular la carga de trabajo de la siguiente forma:

- Se crea un hilo de ejecución por usuario, dentro del cual solamente se puede producir una petición a la vez.
- Se introduce un retraso entre peticiones de 90 ms por usuario.

---

<sup>10</sup>En el contexto de *JMeter*, se simula un usuario como un hilo de ejecución.

- Los parámetros de cada petición *CMS* se obtienen un archivo y todos los usuarios comparten el archivo de peticiones, sin embargo pero cada usuario lee parámetros diferentes.
- Las pruebas se realizan incrementando el número de usuarios, comenzando con 10, aumentando a 20, 30, 40, 50 y 60 usuarios en cada prueba.
- En cada prueba se emplea un componente de la herramienta llamado Informe Agregado, que sirve para mostrar de forma sintética el número total de peticiones, la media y la desviación estándar del tiempo de respuesta, así como el tiempo mínimo y máximo de respuesta, el porcentaje de errores, el rendimiento en peticiones por segundo y la tasa de transferencia promedio en bytes.
- Se monitorea visualmente el consumo de recursos de los servidores empleados.
- La duración de cada prueba es de 120 segundos.

## 5.6. Análisis de resultados

Cada una de las pruebas realizadas produce un conjunto de resultados y métricas, que se presentan en cinco partes:

- a) La primera parte presenta una gráfica con información estadística acerca del tiempo de respuesta: la media, el máximo y la desviación estándar. La media y el máximo se presentan como líneas suavizadas con la finalidad de obtener una tendencia a partir de datos anteriores. Por otra parte, la desviación estándar se muestra como un rango centrado alrededor de la media, dando idea del conjunto de valores para el tiempo de respuesta. Esta información sirve para estimar el tiempo que un cliente *CMS* debe esperar para recibir una imagen dependiendo de la carga del sistema, la cual se expresa en términos del número de usuarios concurrentes.

- b) La segunda gráfica sobrepone la tasa de procesamiento de imágenes con el promedio en Kbytes transferidos por unidad de tiempo. Estos resultados sirven como guía para estimar los requisitos de ancho de banda de la red, la velocidad de acceso y lectura del sistema de archivos, así como las capacidades mínimas del subsistema de almacenamiento para atender un determinado número de usuarios sin degradar la tasa de procesamiento.
- c) La tercera gráfica es importante porque relaciona el tiempo de respuesta y el número de imágenes devueltas por segundo con el número de usuarios concurrentes que atiende el servidor. Estos resultados muestran como el aumento en el número de usuarios concurrentes incrementa el tiempo de respuesta y produce una penalización en la tasa de procesamiento. Esta información en conjunto sirve para establecer la carga que puede soportar el sistema sin degradar su desempeño, tomando como parámetro el número de usuarios concurrentes.
- d) La cuarta parte presenta la información sobre la utilización de memoria o uso del procesador, según sea el caso.
- e) La última parte es complementaria y presenta el conjunto de resultados empleados para la creación de las gráficas.

### **Prueba 1: Línea base de desempeño WMS para *OpenLayers***

Debido a que se emplean dos servidores WMS para la generación de mapas, la prueba se divide en dos para modelar independientemente el comportamiento de cada servidor.

La gráfica de la Figura 5.6 muestra los resultados obtenidos para cada servidor WMS involucrado<sup>11</sup>, en los que puede observarse que conforme el número de usuarios se incrementa, el

---

<sup>11</sup>Denotamos con la letra *A* al servidor PowerEdge 2800, mientras que la letra *B* corresponde al servidor PowerEdge 2900.

tiempo de respuesta de cada petición aumenta debido a que debe emplearse la misma cantidad de recursos para atender un número mayor de peticiones.

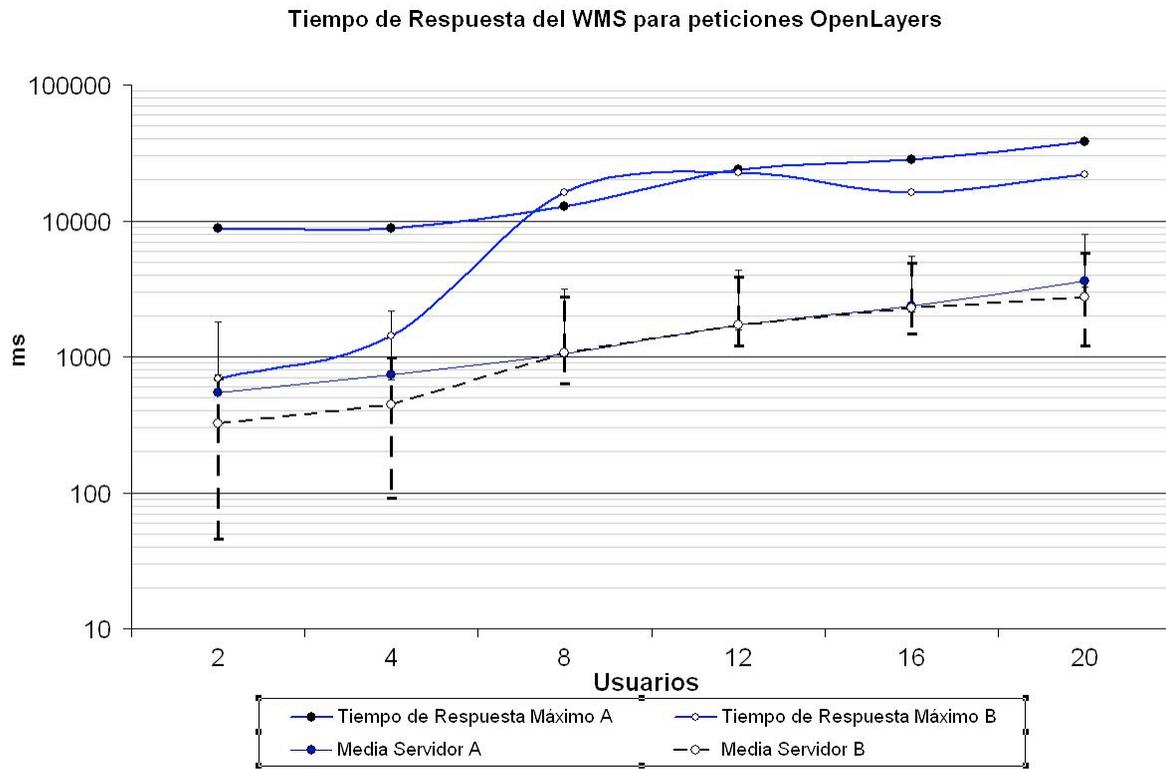


Figura 5.6: Tiempos de respuesta obtenidos con servidores *WMS* y el *API OpenLayers* en escala logarítmica

El ritmo al cual aumenta el tiempo de respuesta depende de la capacidad de cada servidor. En este caso, el servidor A posee aproximadamente un tercio de la capacidad de cómputo y recursos disponibles para B, lo cual se refleja en tiempos de respuesta significativamente mayores que los obtenidos con B. Cabe mencionar que los tiempos de respuesta se presentan en una escala logarítmica y forman una línea recta, lo cual se traduce en una curva al presentarse en una escala lineal.

En la Figura 5.7 se muestra la relación actual entre el número de peticiones y la cantidad de bytes transferidos. En ella se observa una baja en la tasa de transferencia y por consiguiente

un bajo índice de transferencia de bytes al atender a más de ocho usuarios concurrentes, con una tendencia a la baja.

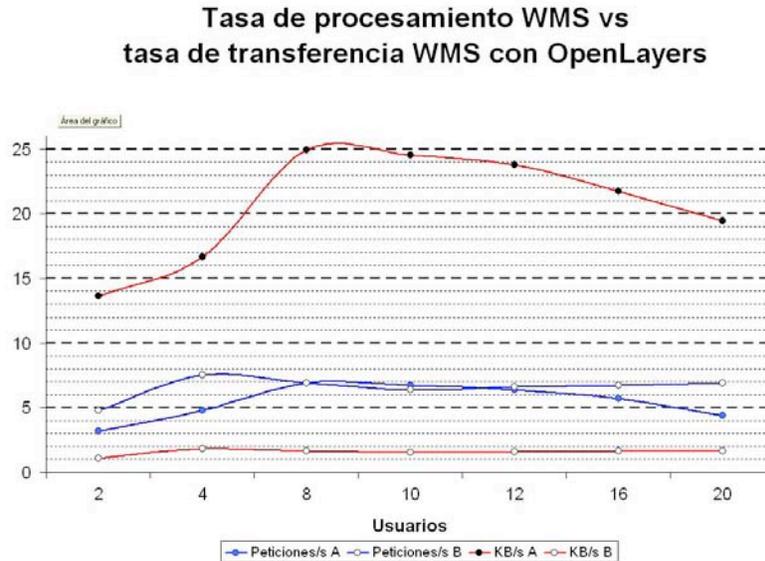


Figura 5.7: Peticiones por segundo vs KB por segundo empleando servidores WMS con el API OpenLayers

En la Figura 5.8 al sobreponer el tiempo de respuesta con el número de peticiones, se observa que los servidores WMS tienen problemas para atender más de ocho usuarios, debido a que el tiempo de respuesta aumenta mientras que la tasa de procesamiento disminuye significativamente. Es posible determinar a partir de esta información que el equipo y los servidores WMS no poseen la capacidad de atender a más de 20 usuarios concurrentes y mucho menos generar la cantidad de imágenes necesarias para las aplicaciones que despliegan mapas como mosaicos de imágenes.

En el Cuadro 5.4 se observa la utilización del procesador al atender peticiones WMS, y se confirma que la generación de imágenes georeferenciadas emplea algoritmos computacionalmente intensivos. En el caso de esta prueba, al atender más de 12 usuarios, la utilización del procesador es completa, por lo que se considera un cuello de botella la capacidad de

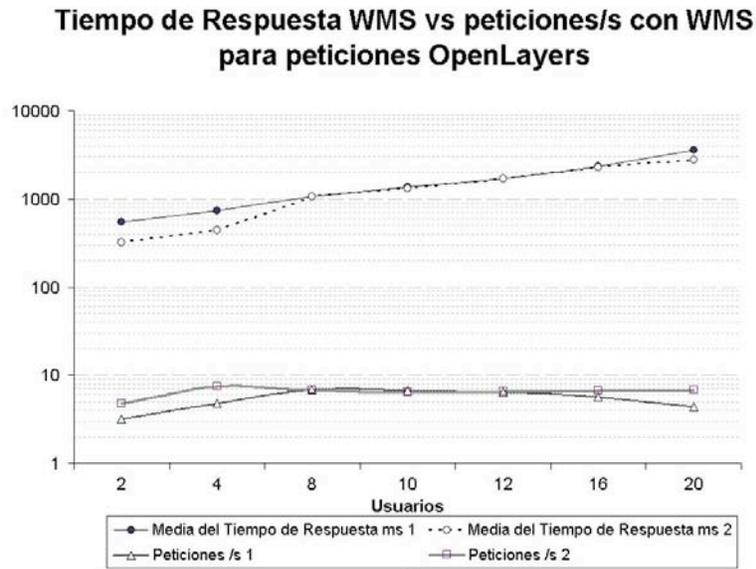


Figura 5.8: Peticiones por segundo vs Tiempo de respuesta empleando servidores *WMS* con el *API OpenLayers*

cómputo del equipo, y por tanto, un límite de operación.

Usuarios	2	4	8	10	12	16	20
Servidor A	50	75	75	75	75	100	100
Servidor B	50	50	75	90	100	100	100

Cuadro 5.4: Porcentaje de uso del procesador atendiendo peticiones *WMS* del *API OpenLayers*.

Los datos obtenidos de la realización de la prueba se presentan en el Cuadro 5.5 y el Cuadro 5.6. La media, el mínimo, el máximo y la desviación estándar se miden en milisegundos. El rendimiento se muestra como peticiones realizadas por segundo.

## Prueba 2: Línea base de desempeño *WMS* para *NASA WorldWind*

En esta prueba se divide el conjunto de datos en dos para modelar independientemente el comportamiento de cada servidor.

Usuarios	Muestras	Media	Min	Max	Desviación Estándar	Rendimiento	KB / s
2	379	541	31	8813	1474	3.2	13.62
4	585	739	46	8812	1366.75	4.8	16.64
8	856	1059	46	12826	2093	6.9	24.89
10	849	1371	46	15531	2485.25	6.7	24.51
12	838	1703	46	23906	2965.53	6.4	23.75
16	807	2345	46	28078	3985.11	5.7	21.71
20	690	3583	46	38422	5382.39	4.4	19.43

Cuadro 5.5: Datos obtenidos para pruebas WMS con *OpenLayers* en el servidor A

Usuarios	Muestras	Media	Min	Max	Desviación Estándar	Rendimiento	KB / s
2	582	323	234	688	90.52	4.8	1.1
4	898	442	235	1422	182.94	7.5	1.84
8	832	1063	250	16110	1268.05	6.9	1.67
10	849	1325	250	22453	1632.04	6.4	1.54
12	798	1715	250	22656	1825.4	6.6	1.62
16	621	2270	250	16188	1423.34	6.7	1.63
20	854	2763	250	21921	1885.92	6.9	1.68

Cuadro 5.6: Datos obtenidos para pruebas WMS con *OpenLayers* en el servidor B

La Figura 5.9 muestra la información obtenida sobre los tiempos de respuesta y se observa que a partir de 16 usuarios concurrentes, hay un crecimiento acelerado en el tiempo de respuesta fuera de los valores tolerados.

Este límite está asociado con el tamaño de la cola de peticiones, que limita a 20 el número de peticiones en espera.

En la Figura 5.10 se observa que conforme el número de usuarios aumenta, la tasa de procesamiento llega al tope de 11 peticiones, y comienza a descender hasta colapsar. Con los resultados obtenidos se estima que empleando únicamente servidores WMS para generar imágenes georeferenciadas, sólo se puede atender a un máximo de 16 usuarios concurrentes sin aumentar el tiempo de respuesta ni degradar la tasa de procesamiento, que no obstante es baja.

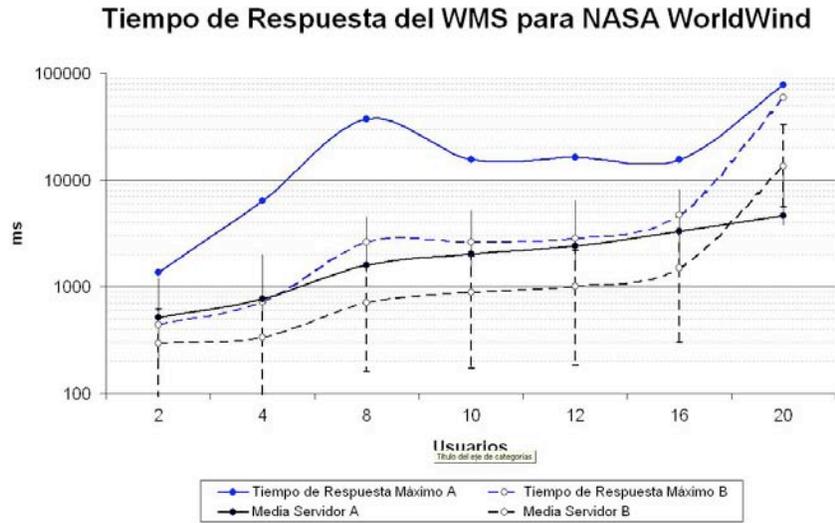


Figura 5.9: Tiempos de respuesta obtenidos con servidores WMS y el API OpenLayers en escala logarítmica

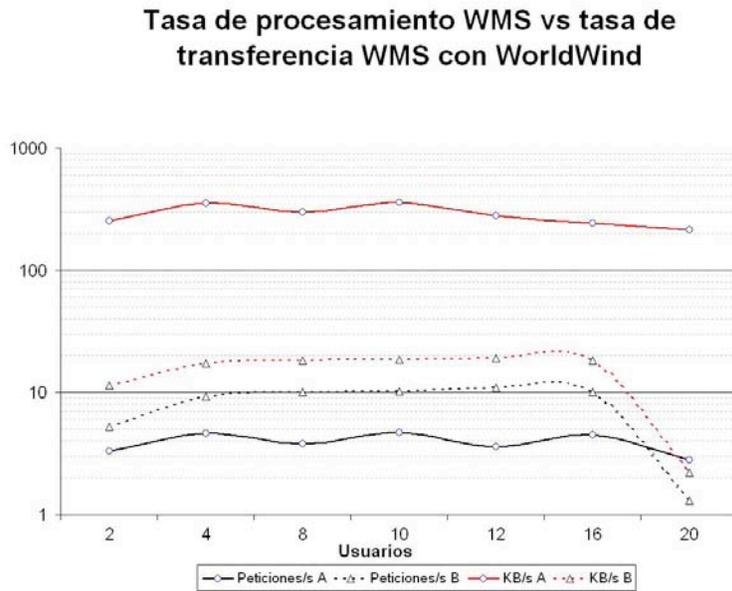


Figura 5.10: Peticiones por segundo vs tiempo de respuesta empleando servidores WMS con NASA WorldWind

En la Figura 5.11 se observa, que al sobreponer el tiempo de respuesta con la tasa de procesamiento del sistema, a partir de 16 usuarios concurrentes el tiempo de respuesta crece aceleradamente, mientras el número de peticiones que el sistema puede atender disminuye. Estos resultados coinciden con los resultados de la Prueba 1 y con aquellos empleados como referencia [ESR07], indicando un bajo rendimiento de los servidores WMS en las pruebas.

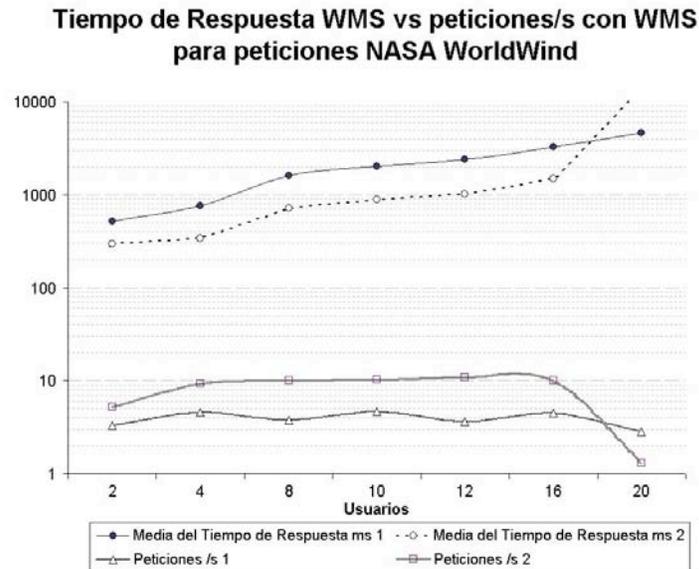


Figura 5.11: Peticiones por segundo vs tiempo de respuesta empleando servidores WMS con *NASA WorldWind*

El Cuadro 5.7 muestra el porcentaje de utilización del procesador para la generación de imágenes. Al igual que en la Prueba 1, al atender más de 12 usuarios la utilización del procesador es completa, por lo que la capacidad de cómputo del equipo se considera un cuello de botella, y por lo tanto, un límite de operación.

Los datos obtenidos de la realización de la prueba se presentan en los Cuadros 5.8 y 5.9. La media, el mínimo, el máximo y la desviación estandar se miden en milisegundos. El rendimiento se muestra como peticiones realizadas por segundo.

Usuarios	2	4	8	10	12	16	20
<b>Servidor A</b>	50	75	100	100	100	100	100
<b>Servidor B</b>	50	50	75	90	100	100	100

Cuadro 5.7: Porcentaje de uso de procesador procesando las peticiones *WMS* de la aplicación *NASA WorldWind*.

Usuarios	Muestras	Media	Min	Max	Desviación Estándar	Rendimiento	KB / s
2	398	513	46	1359	401.19	3.3	254.16
4	567	763	46	6391	992.39	4.6	352.02
8	600	1600	46	37547	2666.9	3.8	298
10	581	2012	47	15625	2421	4.7	358
12	597	2405	47	16422	3240.31	3.6	277
16	598	3296	62	15516	3249.1	4.5	243.75
20	535	4603	46	77657	7511.84	2.8	213

Cuadro 5.8: Datos obtenidos para pruebas *WMS* con *NASA WorldWind* en el servidor A

Usuarios	Muestras	Media	Min	Max	Desviación Estándar	Rendimiento	KB / s
2	624	294	250	438	42.43	5.2	11.34
4	1126	336	250	719	77.97	9.3	17.32
8	1201	711	250	2594	316	10	18.26
10	1233	886	886	2610	343	10.2	18.52
12	1316	1006	1006	2813	364.4	10.9	19.08
16	1210	1501	4672	4672	604.36	10	18.21
20	193	13505	344	59516	11183.46	1.3	2.2

Cuadro 5.9: Datos obtenidos para pruebas *WMS* con *NASA WorldWind* en el servidor B

### Prueba 3: Desempeño Esperado del *CMS* para *OpenLayers*

En esta prueba no se divide el conjunto de datos, ya que el *CMS* agrega servidores *WMS* de forma automática; sin embargo se reportan dos ejecuciones que, a diferencia de las primeras pruebas, alcanzan 60 usuarios concurrentes.

En la gráfica de la Figura 5.12 se observan los resultados obtenidos y se encuentra que la desviación estándar toma valores mayores a la de la mitad de la media, pero semejantes a ésta,

indicando una distribución normal de los datos alrededor de la media. En general el tiempo de respuesta de las peticiones se encuentra por debajo de los 250 ms, sin alcanzar valores mayores a 3000 ms. Estos resultados contrastan con los valores obtenidos de la Prueba 1 que oscilan entre 290 ms y 140,000 ms. Respecto a los criterios de evaluación, se considera que se disminuye el tiempo de respuesta consistentemente a lo largo de toda la prueba.

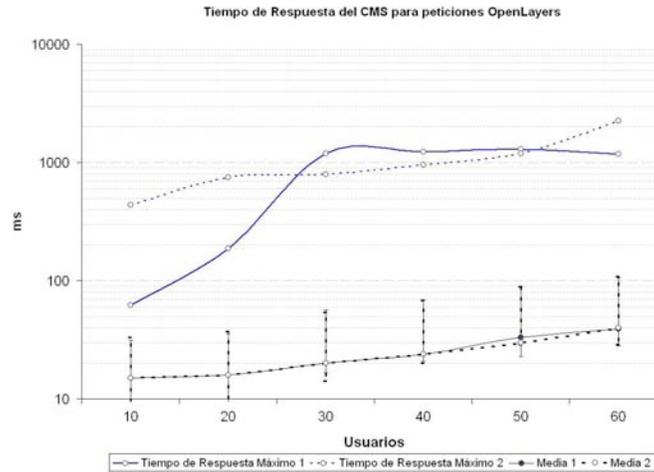


Figura 5.12: Tiempos de respuesta obtenidos para el *API OpenLayers* en escala logarítmica

La Figura 5.13 muestra la relación entre el número de peticiones atendidas por el *CMS* y se observa que puede atender a 60 usuarios y entregar una cantidad considerable de imágenes por segundo. Debe notarse que el número de bytes aumenta conforme más peticiones se procesan y por consiguiente proporciona una idea de las capacidades que debe tener el sistema de almacenamiento para atender cargas de trabajo mayores, ya que al atender 450 peticiones por segundo requiere transmitir más de un megabyte de información por segundo. Aunque esta cantidad puede parecer pequeña, debe considerarse el tráfico adicional de las redes involucradas y la carga impuesta al sistema de almacenamiento.

Cuando se combinan los tiempos de respuesta con la tasa de procesamiento de la prueba, es fácil observar que el sistema escala adecuadamente conforme el número de usuarios aumenta. Esto se observa en la Figura 5.14 cuando al aumentar el número de usuarios, el tiempo de

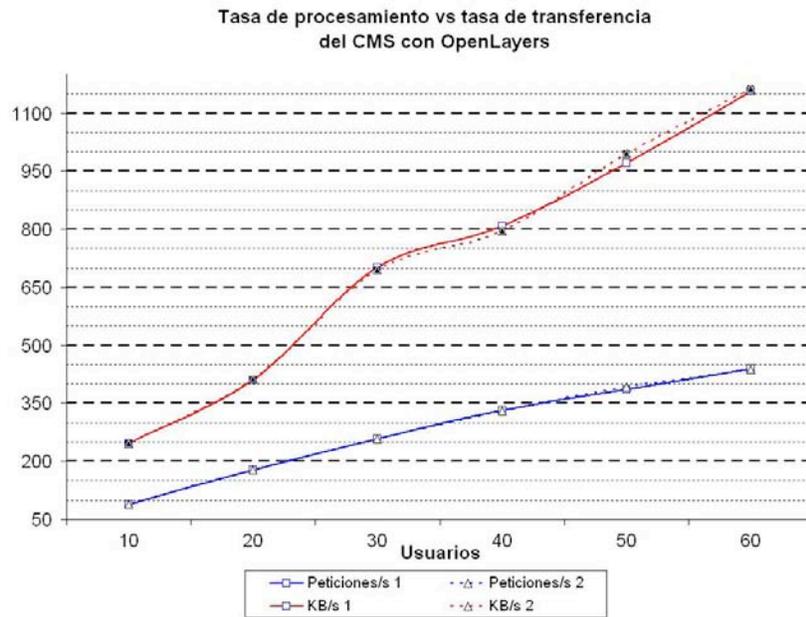


Figura 5.13: Peticiones por segundo vs KB por segundo para el API *OpenLayers*

respuesta aumenta proporcionalmente al número de peticiones atendidas.

En la Figura 5.15 se presentan los resultados que otorga el contenedor *Apache Tomcat* sobre el uso de memoria, la carga del sistema, el número de peticiones en espera y la estabilidad general del sistema. La información que presenta la gráfica abarca las tres últimas ejecuciones de la prueba, es decir, con 40, 50 y 60 usuarios concurrentes. Aunque no se presentan los resultados para el comienzo de la prueba, la diferencia encontrada es ínfima y se omite por brevedad. El Cuadro 5.10 muestra el porcentaje de uso del procesador al atender distinto número de usuarios.

Usuarios	10	20	30	40	50	60
Porcentaje	5	7	7	10	10	12

Cuadro 5.10: Uso de procesador atendiendo peticiones CMS del API *OpenLayers* .

Los datos obtenidos de la realización de la prueba se presentan en los Cuadros 5.11 y 5.12.

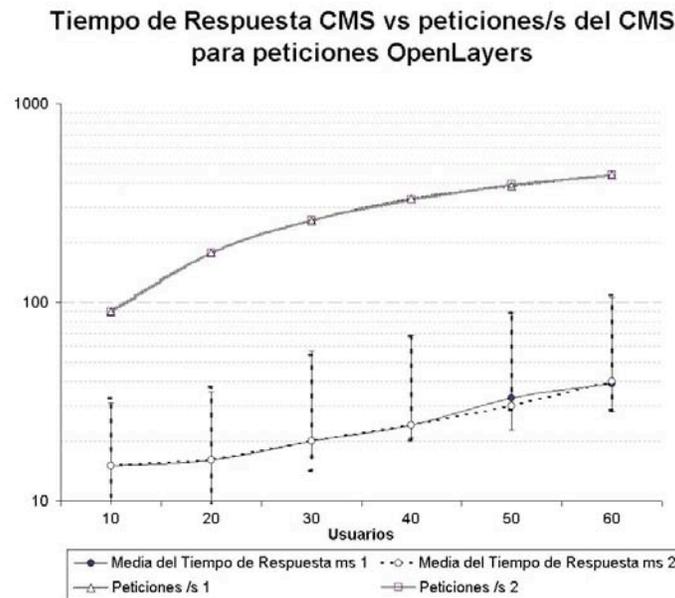


Figura 5.14: Peticiones por segundo vs tiempo de respuesta para el *API OpenLayers*

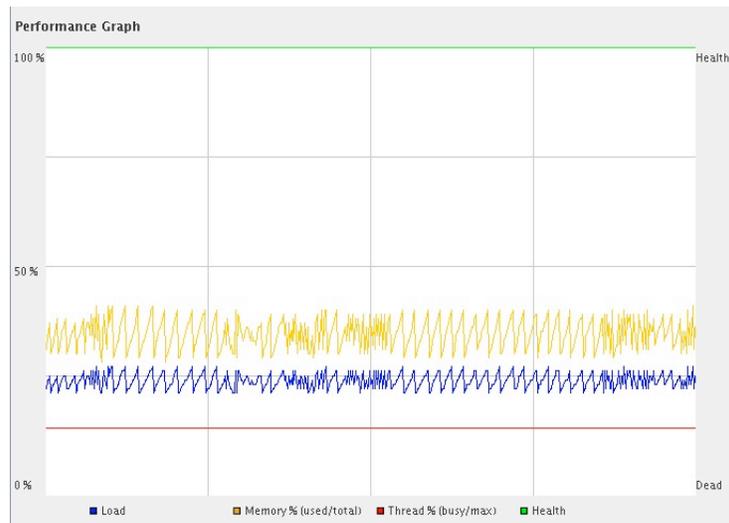


Figura 5.15: Carga del sistema, uso de memoria y peticiones en espera

La media, el mínimo, el máximo y la desviación estandar se miden en milisegundos. El rendimiento se muestra como peticiones realizadas por segundo.

Usuarios	Muestras	Media	Min	Max	Desviación Estándar	Rendimiento	KB / s
10	10784	15	0	62	2.17	89.7	245.5
20	21311	16	0	188	6.73	178.1	411.65
30	30997	20	0	1188	33.41	257.9	700.9
40	40298	24	0	1235	41.52	331.8	807.58
50	48366	33	0	1297	45.6	385.2	971
60	53595	39	0	1172	56.2	438.5	1155.85

Cuadro 5.11: Datos obtenidos para prueba 1 del CMS con *Open Layers*

Usuarios	Muestras	Media	Min	Max	Desviación Estándar	Rendimiento	KB / s
10	10710	15	0	438	6.18	89.5	245.74
20	21237	16	0	750	10.72	177.7	413.39
30	31026	20	0	797	28.35	258.6	695.74
40	39531	24	0	953	40.25	330.5	795.67
50	48041	30	0	1187	57.39	393	995
60	53261	40	0	2266	57.49	438.2	1163.25

Cuadro 5.12: Datos obtenidos para la prueba 2 del CMS con *OpenLayers*

#### Prueba 4: Desempeño esperado del CMS para NASA WorldWind

En la última prueba se presentan los resultados de dos ejecuciones similares y los resultados para el tiempo de respuesta están en de la Figura 5.16. Se observa que la desviación estándar toma valores mayores a la de la mitad de la media, aunque para ser precisos, es un valor semejante a ésta. Esto nos indica que la distribución de los datos es normal alrededor de la media empleando el CMS. En general todas las peticiones se encuentran por debajo de los 250 ms estimados para el CMS. Por otra parte, los tiempos de respuesta máximos no alcanzan valores mayores a 1000 ms, contrastando con los valores obtenidos con los servidores WMS que oscilan entre 40,000 ms y 100,000 ms.

El número de peticiones atendidas por el CMS para la aplicación NASA WorldWind se observa en la Figura 5.17 y se encuentra que, conforme el número de peticiones aumenta, la cantidad de bytes transferidos aumenta proporcionalmente, con lo que es posible calcular la velocidad

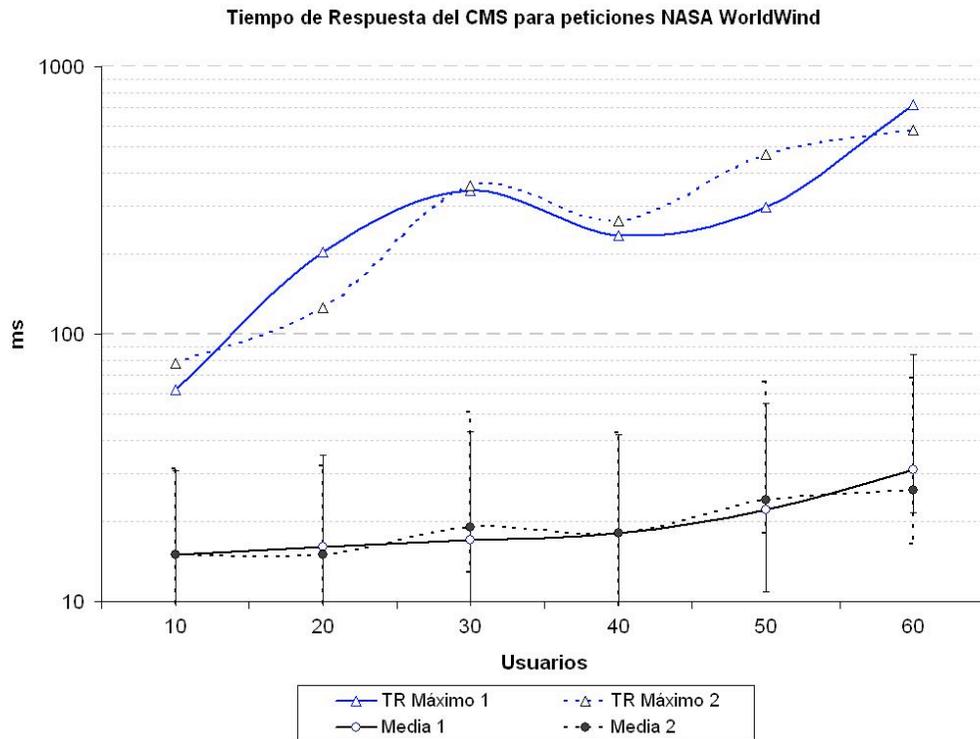


Figura 5.16: Tiempos de respuesta obtenidos para *NASA WorldWind* en escala logarítmica

de transferencia necesaria del sistema de almacenamiento para atender cargas mayores de trabajo, así como el ancho de banda necesario para entregar tal volumen de información.

De la Figura 5.18 se concluye que el tiempo de respuesta es proporcional al número de peticiones atendidas y aumenta conforme se procesa un mayor número de peticiones. En la gráfica se observa que el sistema puede procesar más de 300 peticiones por segundo sin problema aparente y en las pruebas alcanza un total de 450 al atender a 60 usuarios concurrentes. Con la finalidad de no comprometer la estabilidad del servidor y con ello, al resto de servicios existentes, se establece que el sistema puede atender un máximo de 1200 peticiones antes de devolver un mensaje de error, aunque esto no es indicador del número máximo de usuarios que el sistema puede atender.

En la Figura 5.19 se presentan los resultados que otorga el contenedor *Apache Tomcat* sobre

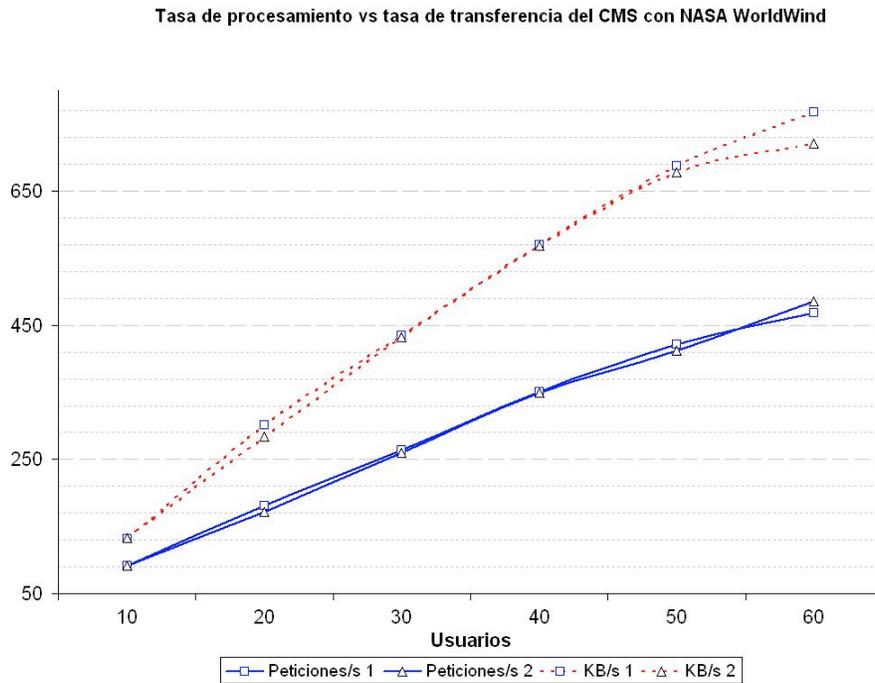


Figura 5.17: Peticiónes por segundo vs KB por segundo para *NASA WorldWind*

el uso de memoria, la carga del sistema, el número de peticiones en espera y la estabilidad general del sistema. La información que presenta la gráfica abarca las tres últimas ejecuciones de la prueba, es decir, 40, 50 y 60 usuarios concurrentes. Aunque no se presentan los resultados al comienzo de la prueba, la diferencia es despreciable y se omite por brevedad. El Cuadro 5.13 muestra el porcentaje de uso del procesador al atender distinto número de usuarios.

Usuarios	10	20	30	40	50	60
Porcentaje	5	7	7	10	10	12

Cuadro 5.13: Uso de procesador atendiendo peticiones *CMS* de *NASA WorldWind*

En el Cuadro 5.14 y el Cuadro 5.15 se presentan los resultados obtenidos de la realización de la prueba. La media, el mínimo, el máximo y la desviación estandar se miden en milisegun-

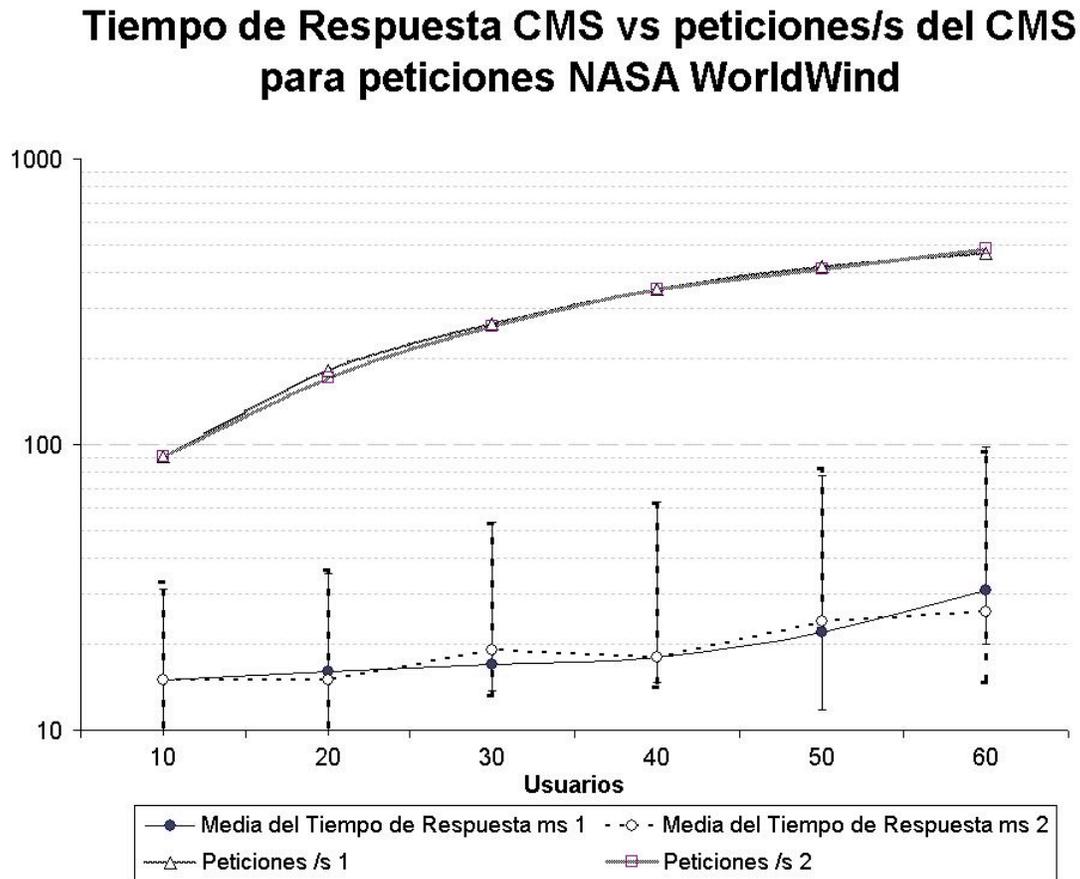


Figura 5.18: Tiempo de respuesta vs peticiones por segundo para *NASA WorldWind*

dos. El rendimiento se muestra como el número peticiones realizadas por segundo.

Usuarios	Muestras	Media	Min	Max	Desviación Estándar	Rendimiento	KB / s
10	11221	15	0	62	1.7	90.8	132.09
20	22038	16	0	203	6.54	181.2	301.68
30	32127	17	0	344	18.57	263.5	434.51
40	42183	18	0	234	12.42	350.7	570
50	50813	22	0	297	21.76	421.4	688.25
60	56887	31	0	719	42.95	468.4	768

Cuadro 5.14: Datos obtenidos para prueba 1 del *CMS* con *NASA WorldWind*

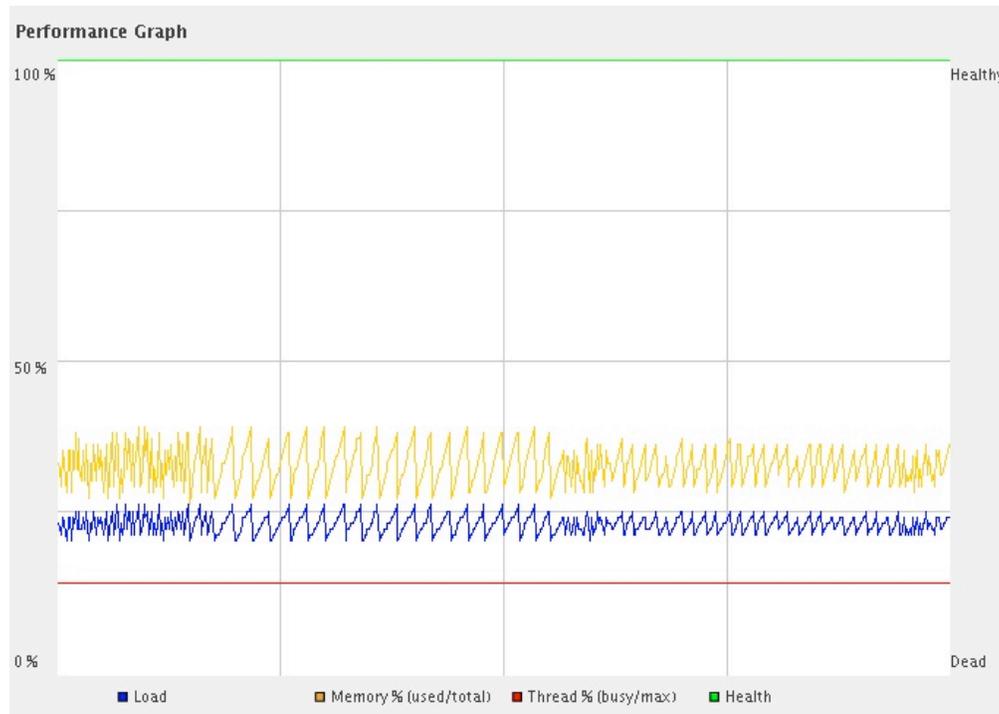


Figura 5.19: Carga del sistema, Uso de memoria y peticiones en espera.

Usuarios	Muestras	Media	Min	Max	Desviación Estándar	Rendimiento	KB / s
10	10898	15	0	78	2.84	91	133
20	20852	15	0	125	4.11	171.7	283.5
30	31655	19	0	359	25.83	260.3	432.7
40	42039	18	0	266	13.88	349	568.64
50	50289	24	0	469	35.99	412.7	678.23
60	59019	26	0	578	32.76	486	721.2

Cuadro 5.15: Datos obtenidos para prueba 2 del CMS con NASA WorldWind

## 5.7. Conclusiones de la evaluación

A partir del conjunto de pruebas se concluye lo siguiente para los servidores WMS:

- Se confirma que el hardware disponible solamente tiene capacidad para atender 8 peticiones WMS sin una penalización significativa del rendimiento.

- El empleo de servidores *WMS* para la generación masiva de imágenes a una tasa mayor de 8 por segundo no es viable a menos que se disponga de suficiente hardware y software adecuado para la distribución de carga.
- El uso del procesador para la generación de imágenes georeferenciadas es elevada, alcanzando el 100 % de uso para atender pocas peticiones, generando inestabilidad y comprometiendo el servicio de generación de imágenes.
- El empleo exclusivo de servidores *WMS* para atender aplicaciones que solicitan imágenes para formar mosaicos no es viable en general, ya sea por la inestabilidad del sistema en situaciones extremas, como por los elevados tiempos de respuesta de cada petición.

De los resultados obtenidos para las pruebas al *CMS* se concluye lo siguiente:

- Los criterios de evaluación planteados se cumplen satisfactoriamente, superando las 300 peticiones por segundo y produciendo tiempos de respuesta menores a 250 ms.
- El uso del procesador al emplear el *CMS* es bajo comparado con las cifras obtenidas al emplear servidores *WMS*, rondando por debajo del 25 % en un sistema de cuatro procesadores.
- El uso de memoria para atender las peticiones es reducido, no superando el 50 % de la memoria asignada a la máquina virtual de Java, que es de 1GB de memoria.
- La cifra propuesta de 600 peticiones por segundo no se alcanza en las pruebas, debido principalmente a los límites de una sola máquina para simular un elevado número de peticiones en un tiempo relativamente corto, ya que aunque no se reportan formalmente los resultados de uso del procesador en las máquinas cliente, aquél alcanza el 100 % al simular 60 peticiones por segundo.
- El empleo de *CMS* en reemplazo del *WMS* para servir imágenes referenciadas de mosaicos es una solución viable debido a la alta tasa de procesamiento obtenida y a los bajos tiempos de respuesta, despreciando el ancho de banda de la red.

# Capítulo 6

## Conclusiones

Las herramientas especializadas en la generación de imágenes georeferenciadas han alcanzado un avance considerable, al grado de ser útiles para atender la mayoría de las necesidades cartográficas.

Los sistemas de cómputo dedicados a la distribución de información geoespacial en forma de mapas deben garantizar ser interoperables entre sistemas heterogéneos. El estándar *OGC WMS* es una propuesta con mucha aceptación al respecto, ya que su uso promueve y facilita tanto la distribución de imágenes geoespaciales, así como la creación de aplicaciones geográficas basadas en interfaces comunes.

El uso de servidores *WMS* como fuente de datos para aplicaciones que consumen imágenes en forma de mosaico es una solución simple pero de bajo rendimiento, ocasionado por el alto costo computacional de generar imágenes georeferenciadas. Debido a esto, la búsqueda de soluciones computacionalmente eficientes resulta indispensable para consolidar la distribución de mapas y la creación de aplicaciones geográficas de alto rendimiento con características avanzadas de visualización.

El desarrollo de un *CMS* para almacenar las imágenes previamente solicitadas propone una

solución transparente y compatible con las aplicaciones que consumen imágenes en mosaico a partir de servidores *WMS*, ya que permite emplear los clientes existentes sin realizar modificaciones fuera del estándar, facilitando así, su desarrollo e implementación.

El presente trabajo revisa el problema detalladamente, realiza un análisis del mismo y propone una solución acorde con las necesidades particulares de las aplicaciones consideradas. Esta propuesta lleva al diseño e implementación de un *CMS* que permite obtener resultados satisfactorios de desempeño.

En resumen, el análisis, diseño, implementación y evaluación del desempeño del sistema lleva a las siguientes conclusiones:

- El empleo de un *CMS* para almacenar las imágenes generadas por un *WMS* y devolverlas al recibir una petición previa reduce la carga de los servidores *WMS* y disminuye el tiempo de respuesta. Este beneficio se traduce en un intercambio del tiempo de procesamiento por espacio en memoria, que es más fácil incrementar al momento del presente trabajo, además de que tiene un costo menor.
- La disminución de la carga en los sistemas permite el empleo de equipo menos especializado y costoso para implementar soluciones de alto rendimiento. Estos cambios se traducen en equipo con menos memoria y procesadores de menor velocidad, a cambio de una mayor capacidad de almacenamiento.
- El uso del *CMS* hace posible la creación de aplicaciones con un alto nivel de interacción e información geográfica, aumentando la satisfacción general del usuario y la utilidad de las aplicaciones.
- La implementación del *CMS* promueve el empleo de interfaces basadas en estándares, con énfasis en la interoperabilidad.
- La posibilidad de distribuir información a una velocidad mayor, incrementa el valor de la información e introduce un valor agregado a los sistemas computacionales.

- El empleo de herramientas de desarrollo de alto nivel, permite la implementación, mantenimiento, evolución y mejora del sistema para extender sus capacidades, aumentando su utilidad y por consiguiente su valor.

De manera complementaria a investigación realizada en el presente trabajo, se obtienen resultados relacionados, relevantes al tema de estudio que compete a la tesis, los cuales se exponen a continuación:

- La carencia de herramientas y metodologías para la evaluación del rendimiento de servicios Web geográficos dificulta la definición de una línea base de desempeño, que deberá ser alcanzada o superada por todo desarrollo dentro del área. Al respecto, el presente trabajo propone una metodología para dicha evaluación, la cual es repetible y fundamentada, proporcionando así una línea base de desempeño y herramientas acordadas.
- Durante la realización de las pruebas, se encontró que las diversas aplicaciones que funcionan como clientes de servicios *WMS* no siguen adecuadamente las especificaciones propuestas por la OGC, complicando así la implementación de soluciones geoespaciales y dificultando la interoperabilidad entre distintos productos.
- El presente trabajo es una contribución significativa a la Geomática, ya que aporta los conocimientos de las Ciencias de la Computación a la solución de problemas de sistematización de información geoespacial, distribución, visualización, desarrollo de aplicaciones y creación de herramientas auxiliares para el procesamiento de datos.

## **Trabajo a futuro**

La creación de mosaicos de imágenes está fuertemente asociada con la elección de parámetros geográficos, como las distintas proyecciones geográficas disponibles, el conjunto de

códigos *EPSG*<sup>1</sup> que definen las proyecciones y la precisión numérica de los sistemas de cómputo. La introducción de variaciones en alguno de estos elementos produce inconsistencias que derivan en incompatibilidades entre sistemas y por lo tanto en un mal aprovechamiento del sistema de caché.

La generación de estándares para la creación de mosaicos permitiría generar mecanismos de validación y emplear sistemas de almacenamiento alternos, reduciendo problemas de almacenamiento duplicado y controlando los problemas de precisión que pueden ocurrir debido a las diferentes implementaciones numéricas. Por estas razones, es importante trabajar en la estandarización de los parámetros empleados para generar mosaicos de imágenes, para facilitar el empleo de diferentes proyecciones acordes a las necesidades de quien distribuya información geográfica.

---

<sup>1</sup>Siglas de *European Petroleum Survey Group*

# Apéndice A

## Ambiente físico y lógico.

Ambiente físico y lógico empleado tanto para pruebas como en producción.

### Hardware y Software

a) Servidor Dell PowerEdge 2800 con 2 procesadores Xeon corriendo a 3.2Ghz y 2MB Cache L2. Hyperthreading activo. 3GB memoria RAM.

Dos discos duros SCSI 320 con capacidad de 300GB y girando a 10k RPM. Los discos se encuentran dentro de un arreglo RAID nivel 0.

Sistema operativo Windows 2003 Server.

Servidor de mapas ESRI ArcIMS 9.1 Service Pack 4 y WMSConnector 9.1.

Software para edición de mapas ESRI ArcView 9.1.

Contenedor de Servlets Java Apache Tomcat versión 5.0.24.

Servidor Web Apache 2.0.24.

Lenguaje de programación Java JDK 1.4.2

b) Servidor Dell PowerEdge 2900. 2 Procesadores Doble Núcleo Xeon 3Ghz. 4MB Ca-

che. 4GB memoria RAM. Cuatro discos duros SAS de 300GB girando a 10k revoluciones. Los discos duros se encuentran dentro de un arreglo RAID nivel 1. Sistema operativo Windows 2003 Server R2.

Servidor de mapas ESRI ArcIMS 9.2 SP4 y WMS Connector 9.2.

Software para edición de mapas ArcView 9.2.

Contenedor de Servlets Apache Tomcat 5.5

Servidor Web Apache 2.0.

Lenguaje de programación Java JDK 1.5.0

- c) Computadora Personal Dell Optiplex GX620, 1 Procesador Pentium D840 3.2Ghz, 2GB memoria RAM. Disco Duro SATA 250GB. Sistema Operativo Windows XP Professional con Service Pack 2.

## Redes

El equipo de red empleado en producción se compone de los siguientes elementos:

- Switch Dell Power Connect 5324. 24 puertos con una velocidad de 1Gbs, Velocidad de switcheo de 3Gb/s.
- 2x Tarjetas de red Intel Pro 1000/MT PCI-E con TOE<sup>1</sup>.
- 2x Tarjetas de red Broadcom PCI-E con TOE

La velocidad de conexión entre servidores es de 2Gbps, empleando enlaces asociados redundantes y un balanceo estático de carga entre tarjetas.

---

<sup>1</sup>Siglas de TCP Offload Engine

# Bibliografía

- [apa08] Apache Tomcat, Febrero 2008, Fecha de Consulta: Febrero 2008.  
<http://tomcat.apache.org/>
- [Bri05] Web Application Stress Test Methodology. ESRI Systems Integration Technical Brief. Reporte Técnico, ESRI, Junio 2005.
- [CLRS01] *Introduction to Algorithms*, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. MIT Press, segunda edición, 2001.
- [Com06] Geospatial Metadata, Noviembre 2006, Fecha de Consulta: Marzo 12, 2008.  
<http://www.fgdc.gov/metadata>
- [ESR] ArcSDE Developer Help, Fecha de Consulta: Abril 2008.  
<http://edndoc.esri.com/arcsde/9.2/concepts/concepts.htm>
- [ESR07] System Design Strategies. ESRI. Technical Reference Document, Environmental Systems Research Institute, Inc, Julio 2007.
- [ext] Design and Implementation of the Second Extended Filesystem, Fecha de Consulta: 18 de Marzo 2008.  
<http://e2fsprogs.sourceforge.net/ext2intro.html#section:ext2fs>
- [fil05] Open GIS Filter Encoding Specification. Reporte Técnico, Open Geospatial Consortium Inc., Mayo 2005.

- [gis06] ESRI GIS Dictionary, Octubre 2006, Fecha de Consulta: 14 de Septiembre 2007.  
<http://support.esri.com/index.cfm?fa=knowledgebase.gisDictionary.gateway>
- [Gon07] Diseño y elaboración de una herramienta didáctica para el trazado de mapas geográficos. Oscar Escamilla González. Tesis de Licenciatura, Facultad de Ciencias, Universidad Nacional Autónoma de México, 2007.
- [Gro98] Content Standard for Digital Geospatial Metadata. Metadata Ad Hoc Working Group. Reporte Técnico, Federal Geographic Data Committee, Junio 1998.
- [Gro04] Web Services Architecture. W3c Working Group. Reporte Técnico, W3c, Febrero 2004.
- [Inc02] Web Map Service Implementation Specification. Open GIS Consortium Inc. Open GIS Publicly available Standard, Open GIS Consortium Inc., Enero 2002.
- [Inc03] OpenGIS Web Services Architecture. Open GIS Consortium Inc. Open GIS Discussion Paper, Open GIS Consortium Inc., Enero 2003.
- [inf05a] Architecture Description Version 2.0 - The Canadian Geospatial Data Infrastructure. The Canadian Geospatial Data infrastructure. Reporte Técnico, GeoConnections, 2005.
- [Inf05b] Vision -Better Knowledge for better decisions. The Canadian Geospatial Data Infrastructure. Reporte Técnico, GeoConnections, 2005.
- [jav08a] Java SE Technologies at glance, Febrero 2008, Fecha de Consulta: Febrero 2008.  
<http://java.sun.com/javase/technologies/index.jspoverview>
- [jav08b] Java Servlet Technology, Febrero 2008, Fecha de Consulta: Febrero 2008.  
<http://java.sun.com/products/servlet/index.jsp>
- [jme08] Apache JMeter, 2008, Fecha de Consulta: 26 de mayo 2008.  
<http://jakarta.apache.org/jmeter/>

- [Mas05] *GIS Worlds, Creating spatial data infrastructures*, Ian Masser. ESRI Press, primera edición, 2005.
- [MFB<sup>+</sup>07] *Performance Testing Guidance for Web Applications, patterns practices*. J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, and Dennis Rea. Reporte Técnico, Microsoft Corporation, 2007.
- [Mic03] How NTFS works, Marzo 2003, Fecha de Consulta: 17 de Marzo 2008.  
<http://technet2.microsoft.com/windowsserver/en/library/8cc5891d-bf8e-4164-862d-dac5418c59481033.msp?mfr=true>
- [Mil] NASA World Wind Tile Structure, Technical Documentation, Fecha de Consulta: 14 de Septiembre 2007.  
<http://ceteranet.com/nww-tile-struct.pdf>
- [Mor01] RM-ODP: El Modelo de Referencia de ISO para el Procesamiento Abierto y Distribuido. Antonio Vallecillo Moreno. (3):69–99, 2001.
- [Neb04] The SDI Cookbook, Douglas D. Nebert, January 2004, Fecha de Consulta: 17 de agosto 2007.  
<http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf>
- [opea] OpenGIS Specifications (Standards), Fecha de Consulta: 2 de Octubre 2007.  
[www.opengeospatial.org/standards](http://www.opengeospatial.org/standards)
- [opeb] OpenLayers, Fecha de Consulta: Marzo 2008.  
<http://openlayers.org/>
- [OSG06] WMS Tiling client Recommendation OSGEO, Noviembre 2006, Fecha de Consulta: 14 de Marzo 2008.  
[http://wiki.osgeo.org/index.php/WMS\\_Tiling\\_Client\\_Recommendation](http://wiki.osgeo.org/index.php/WMS_Tiling_Client_Recommendation)
- [Por07] Geographic Markup Language. Clemens Portele. Open GIS Standard, Open Geospatial Consortium Inc., 2007.

- [Sch] TileCache, from MetaCarta Labs, Fecha de Consulta: 20 Mayo 2008.  
<http://tilecache.org/>
- [sld07] Styled Layer Descriptor profile of the Web Map Service Implementation Specification. Implementation Specification, Open Geospatial Consortium Inc., 2007.
- [Tu06] Web Services for Geographic Information Systems. Sherung Tu. *IEEE Internet Computing*, 10(5):13–15, September/October 2006.
- [wfs02] Web Feature Service Implementation Specification. OGC Implementation Specification, Open Geospatial Consortium Inc., 2002.
- [wha07] ArcGIS 9, What is ArcGIS 9.2, 2007, Fecha de Consulta: 11 de Septiembre 2007.  
[http://webhelp.esri.com/arcgisdesktop/9.2/pdf/what\\_is\\_arcgis.pdf](http://webhelp.esri.com/arcgisdesktop/9.2/pdf/what_is_arcgis.pdf)
- [wmc05] Web Map Context Documents. OGC Implementation Specification, Open Geospatial Consortium Inc., 2005.
- [wor] NASA WorldWind, Fecha de Consulta: 19 de Marzo 2008.  
<http://worldwind.arc.nasa.gov/>
- [Zei99] *Modeling our World, The ESRI Guide to Geotatabase Design*, Michael Zeiler. ESRI Press, 1999.

# Apéndice A

## Ambiente físico y lógico.

Ambiente físico y lógico empleado tanto para pruebas como en producción.

### Hardware y Software

a) Servidor Dell PowerEdge 2800 con 2 procesadores Xeon corriendo a 3.2Ghz y 2MB Cache L2. Hyperthreading activo. 3GB memoria RAM.

Dos discos duros SCSI 320 con capacidad de 300GB y girando a 10k RPM. Los discos se encuentran dentro de un arreglo RAID nivel 0.

Sistema operativo Windows 2003 Server.

Servidor de mapas ESRI ArcIMS 9.1 Service Pack 4 y WMSConnector 9.1.

Software para edición de mapas ESRI ArcView 9.1.

Contenedor de Servlets Java Apache Tomcat versión 5.0.24.

Servidor Web Apache 2.0.24.

Lenguaje de programación Java JDK 1.4.2

b) Servidor Dell PowerEdge 2900. 2 Procesadores Doble Núcleo Xeon 3Ghz. 4MB Ca-

che. 4GB memoria RAM. Cuatro discos duros SAS de 300GB girando a 10k revoluciones. Los discos duros se encuentran dentro de un arreglo RAID nivel 1. Sistema operativo Windows 2003 Server R2.

Servidor de mapas ESRI ArcIMS 9.2 SP4 y WMS Connector 9.2.

Software para edición de mapas ArcView 9.2.

Contenedor de Servlets Apache Tomcat 5.5

Servidor Web Apache 2.0.

Lenguaje de programación Java JDK 1.5.0

- c) Computadora Personal Dell Optiplex GX620, 1 Procesador Pentium D840 3.2Ghz, 2GB memoria RAM. Disco Duro SATA 250GB. Sistema Operativo Windows XP Professional con Service Pack 2.

## Redes

El equipo de red empleado en producción se compone de los siguientes elementos:

- Switch Dell Power Connect 5324. 24 puertos con una velocidad de 1Gbs, Velocidad de switcheo de 3Gb/s.
- 2x Tarjetas de red Intel Pro 1000/MT PCI-E con TOE<sup>1</sup>.
- 2x Tarjetas de red Broadcom PCI-E con TOE

La velocidad de conexión entre servidores es de 2Gbps, empleando enlaces asociados redundantes y un balanceo estático de carga entre tarjetas.

---

<sup>1</sup>Siglas de TCP Offload Engine

# Bibliografía

- [apa08] Apache Tomcat, Febrero 2008, Fecha de Consulta: Febrero 2008.  
<http://tomcat.apache.org/>
- [Bri05] Web Application Stress Test Methodology. ESRI Systems Integration Technical Brief. Reporte Técnico, ESRI, Junio 2005.
- [CLRS01] *Introduction to Algorithms*, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. MIT Press, segunda edición, 2001.
- [Com06] Geospatial Metadata, Noviembre 2006, Fecha de Consulta: Marzo 12, 2008.  
<http://www.fgdc.gov/metadata>
- [ESR] ArcSDE Developer Help, Fecha de Consulta: Abril 2008.  
<http://edndoc.esri.com/arcsde/9.2/concepts/concepts.htm>
- [ESR07] System Design Strategies. ESRI. Technical Reference Document, Environmental Systems Research Institute, Inc, Julio 2007.
- [ext] Design and Implementation of the Second Extended Filesystem, Fecha de Consulta: 18 de Marzo 2008.  
<http://e2fsprogs.sourceforge.net/ext2intro.html#section:ext2fs>
- [fil05] Open GIS Filter Encoding Specification. Reporte Técnico, Open Geospatial Consortium Inc., Mayo 2005.

- [gis06] ESRI GIS Dictionary, Octubre 2006, Fecha de Consulta: 14 de Septiembre 2007.  
<http://support.esri.com/index.cfm?fa=knowledgebase.gisDictionary.gateway>
- [Gon07] Diseño y elaboración de una herramienta didáctica para el trazado de mapas geográficos. Oscar Escamilla González. Tesis de Licenciatura, Facultad de Ciencias, Universidad Nacional Autónoma de México, 2007.
- [Gro98] Content Standard for Digital Geospatial Metadata. Metadata Ad Hoc Working Group. Reporte Técnico, Federal Geographic Data Committee, Junio 1998.
- [Gro04] Web Services Architecture. W3c Working Group. Reporte Técnico, W3c, Febrero 2004.
- [Inc02] Web Map Service Implementation Specification. Open GIS Consortium Inc. Open GIS Publicly available Standard, Open GIS Consortium Inc., Enero 2002.
- [Inc03] OpenGIS Web Services Architecture. Open GIS Consortium Inc. Open GIS Discussion Paper, Open GIS Consortium Inc., Enero 2003.
- [inf05a] Architecture Description Version 2.0 - The Canadian Geospatial Data Infrastructure. The Canadian Geospatial Data infrastructure. Reporte Técnico, GeoConnections, 2005.
- [Inf05b] Vision -Better Knowledge for better decisions. The Canadian Geospatial Data Infrastructure. Reporte Técnico, GeoConnections, 2005.
- [jav08a] Java SE Technologies at glance, Febrero 2008, Fecha de Consulta: Febrero 2008.  
<http://java.sun.com/javase/technologies/index.jspoverview>
- [jav08b] Java Servlet Technology, Febrero 2008, Fecha de Consulta: Febrero 2008.  
<http://java.sun.com/products/servlet/index.jsp>
- [jme08] Apache JMeter, 2008, Fecha de Consulta: 26 de mayo 2008.  
<http://jakarta.apache.org/jmeter/>

- [Mas05] *GIS Worlds, Creating spatial data infrastructures*, Ian Masser. ESRI Press, primera edición, 2005.
- [MFB<sup>+</sup>07] *Performance Testing Guidance for Web Applications, patterns practices*. J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, and Dennis Rea. Reporte Técnico, Microsoft Corporation, 2007.
- [Mic03] How NTFS works, Marzo 2003, Fecha de Consulta: 17 de Marzo 2008.  
<http://technet2.microsoft.com/windowsserver/en/library/8cc5891d-bf8e-4164-862d-dac5418c59481033.msp?mfr=true>
- [Mil] NASA World Wind Tile Structure, Technical Documentation, Fecha de Consulta: 14 de Septiembre 2007.  
<http://ceteranet.com/nww-tile-struct.pdf>
- [Mor01] RM-ODP: El Modelo de Referencia de ISO para el Procesamiento Abierto y Distribuido. Antonio Vallecillo Moreno. (3):69–99, 2001.
- [Neb04] The SDI Cookbook, Douglas D. Nebert, January 2004, Fecha de Consulta: 17 de agosto 2007.  
<http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf>
- [opea] OpenGIS Specifications (Standards), Fecha de Consulta: 2 de Octubre 2007.  
[www.opengeospatial.org/standards](http://www.opengeospatial.org/standards)
- [opeb] OpenLayers, Fecha de Consulta: Marzo 2008.  
<http://openlayers.org/>
- [OSG06] WMS Tiling client Recommendation OSSEO, Noviembre 2006, Fecha de Consulta: 14 de Marzo 2008.  
[http://wiki.osgeo.org/index.php/WMS\\_Tiling\\_Client\\_Recommendation](http://wiki.osgeo.org/index.php/WMS_Tiling_Client_Recommendation)
- [Por07] Geographic Markup Language. Clemens Portele. Open GIS Standard, Open Geospatial Consortium Inc., 2007.

- [Sch] TileCache, from MetaCarta Labs, Fecha de Consulta: 20 Mayo 2008.  
<http://tilecache.org/>
- [sld07] Styled Layer Descriptor profile of the Web Map Service Implementation Specification. Implementation Specification, Open Geospatial Consortium Inc., 2007.
- [Tu06] Web Services for Geographic Information Systems. Sherung Tu. *IEEE Internet Computing*, 10(5):13–15, September/October 2006.
- [wfs02] Web Feature Service Implementation Specification. OGC Implementation Specification, Open Geospatial Consortium Inc., 2002.
- [wha07] ArcGIS 9, What is ArcGIS 9.2, 2007, Fecha de Consulta: 11 de Septiembre 2007.  
[http://webhelp.esri.com/arcgisdesktop/9.2/pdf/what\\_is\\_arcgis.pdf](http://webhelp.esri.com/arcgisdesktop/9.2/pdf/what_is_arcgis.pdf)
- [wmc05] Web Map Context Documents. OGC Implementation Specification, Open Geospatial Consortium Inc., 2005.
- [wor] NASA WorldWind, Fecha de Consulta: 19 de Marzo 2008.  
<http://worldwind.arc.nasa.gov/>
- [Zei99] *Modeling our World, The ESRI Guide to Geotababase Design*, Michael Zeiler. ESRI Press, 1999.