

---

---

# Visualización en Web de productos del pronóstico numérico meteorológico del CCA-UNAM

---

---

Presenta:

ÁNGEL VALENTÍN BAUTISTA DURÁN



Facultad de Ciencias de la Computación  
BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Tesis presentada para obtener el título de:  
LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

Asesores:

DRA. ERIKA DANAÉ LÓPEZ ESPINOZA  
Centro de Ciencias de la Atmósfera, UNAM

M.C. YALÚ GALICIA HERNÁNDEZ  
Facultad de Ciencias de la Computación  
BUAP

OCTUBRE DE 2017



## Resumen

En los últimos años, los modelos numéricos para el pronóstico meteorológico se han convertido en una herramienta indispensable para proporcionar información cada vez más precisa del estado de la atmósfera, no sólo a los especialistas en la materia, sino al público no experto en el tema de la meteorología. Sin embargo, para este último tipo de usuarios es necesario un sistema en línea interactivo y amigable que facilite la visualización y comprensión de la información meteorológica pronosticada.

En este trabajo se hace uso de tecnologías *web mapping* y estándares web para la visualización y distribución de datos meteorológicos generados por el modelo *Weather Research and Forecasting (WRF)* que se encuentra operativo en el Centro de Ciencias de la Atmósfera (CCA) de la UNAM. Se desarrolló una aplicación *web mapping* que permite interpretar cualitativamente mediante símbolos meteorológicos, el pronóstico a 5 días de las variables de temperatura, precipitación y viento en superficie. La aplicación *web mapping* fue implementada usando la arquitectura Modelo-Vista-Controlador (MVC) y responde a los avances tecnológicos en materia de dispositivos móviles.

Por otro lado, la distribución de datos se realizó mediante la creación y el uso de un servicio web utilizando como referencia el modelo *Digital Weather Markup Language (DWML)* proporcionado por la *National Oceanic and Atmospheric Administration (NOAA)* para la creación de servicios web meteorológicos. Cabe mencionar que toda la implementación del sistema en línea se desarrolló con tecnologías de software libre.





## Dedicatoria

*A Braulio Bautista Herrera, tu llegada a nuestras vidas nos llenó de amor y alegría. Permanecerás siempre en nuestros corazones.*

*A mi amada esposa Dulce, por tu apoyo incondicional, por ser mi amiga y confidente, mi fortaleza y mi refugio.*



## Agradecimientos

La presente investigación fue realizada gracias al programa UNAM-DGPA-PAPIIT IA000915 y es un esfuerzo en conjunto en el cual, directa o indirectamente, participaron varias personas leyendo, opinando, corrigiendo, teniéndome paciencia, dando ánimo, acompañando en los momentos buenos y malos, por lo cual merecen las gracias porque sin su valiosa aportación no hubiera sido posible este trabajo.

Agradezco a la Dra. Erika Danaé López Espinoza, por haber confiado en mí, por la paciencia e invaluable dirección de este trabajo, por ser una gran guía, un gran apoyo en los momentos difíciles, porque nunca escatimó esfuerzo y tiempo para corregir y mejorar este trabajo, por sus valiosos conocimientos siempre al servicio del tesista, por ser firme en los momentos que se requería. También agradezco a la M.C. Yalú Galicia Hernández por sus comentarios en todo el proceso de elaboración y sus correcciones.

A cada uno de mis maestros que participaron en mi desarrollo profesional durante mi carrera, compartiendo sus conocimientos, experiencias y formar parte de lo que ahora soy.

Te agradezco a ti mi amada compañera y amiga. Porque eres una extensión de mi espíritu ayudándome a lo largo de mi vida personal y profesional a superarme en cada paso que damos juntos, gracias por tu ayuda, apoyo y fortaleza.

Y finalmente, quiero agradecer a todos mis amigos cercanos y familiares. Todos ustedes han alentado y creído en mí. Me han ayudado a enfocarme en lo que ha sido un proceso enormemente gratificante y enriquecedor. Especialmente a mi hermano, cuyo amor y apoyo inquebrantable me ha impulsado en los momentos más difíciles de mi vida. Lo logré *kakaroto* y voy por más...



# Índice

	Página
<b>Índice de cuadros</b>	<b>ix</b>
<b>Índice de figuras</b>	<b>xi</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación y descripción del problema . . . . .	2
1.2 Justificación . . . . .	4
1.3 Objetivos . . . . .	7
1.3.1 Objetivo general . . . . .	7
1.3.2 Objetivos particulares . . . . .	7
1.3.3 Contribuciones . . . . .	7
1.3.4 Organización de la tesis . . . . .	7
<b>2 Marco teórico</b>	<b>9</b>
2.1 Servicios web ( <i>Web Services</i> ) . . . . .	9
2.1.1 SOAP ( <i>Simple Object Access Protocol</i> ) . . . . .	11
2.1.2 REST ( <i>Representational State Transfer</i> ) . . . . .	12
2.2 Aplicaciones web de mapas ( <i>web mapping applications</i> ) . . . . .	15
2.2.1 Mapas en Internet ( <i>Internet mapping</i> ) . . . . .	15
2.2.2 Estándares OGC ( <i>Open Geospatial Consortium</i> ) . . . . .	16
2.2.3 Sistemas de posicionamiento . . . . .	18
2.2.4 Representación de la información . . . . .	20
2.3 Modelo WRF . . . . .	23
<b>3 Descripción de la aplicación</b>	<b>25</b>

3.1	Planificación . . . . .	26
3.1.1	Definición del problema . . . . .	26
3.1.2	Estrategia de solución . . . . .	26
3.1.3	Operaciones del sistema . . . . .	27
3.1.4	Infraestructura (hardware) . . . . .	27
3.2	Análisis . . . . .	29
3.2.1	Requerimientos del sistema . . . . .	29
3.2.2	Requerimientos no funcionales . . . . .	29
3.2.3	Identificación y listado de los casos de uso . . . . .	30
3.2.4	Criterios generales para la selección de software . . . . .	34
<b>4</b>	<b>Diseño e implementación del sistema</b>	<b>39</b>
4.1	Diseño . . . . .	39
4.1.1	Arquitectura . . . . .	39
4.1.2	Flujo de datos . . . . .	39
4.1.3	Diagrama de componentes . . . . .	40
4.2	Implementación . . . . .	42
4.2.1	Implementación del servicio web basado en la arquitectura REST . . . . .	42
4.2.2	Implementación de la aplicación web de mapas ( <i>web mapping application</i> )	61
<b>5</b>	<b>Resultados</b>	<b>73</b>
5.1	Conclusiones . . . . .	73
5.2	Trabajo futuro . . . . .	73
	<b>Apéndice A Conceptos básicos de Cartografía</b>	<b>75</b>
A.1	Sistemas de Coordenadas . . . . .	75
A.1.1	Latitud y longitud . . . . .	76
A.1.2	Proyecciones . . . . .	76
	<b>Apéndice B Sistemas de información geográfica</b>	<b>79</b>
	<b>Bibliografía</b>	<b>81</b>

## Índice de cuadros

<b>Tabla</b>	<b>Página</b>
1.1 Características de las dos versiones del pronóstico numérico meteorológico. . . . .	2
2.1 Ventajas y desventajas de los modelos raster y vectorial. . . . .	22
3.1 Configuración de los discos duros servidor DELL POWER EDGE T610 . . . . .	28
3.2 Requerimientos funcionales del sistema. . . . .	29
3.3 Requerimientos no funcionales del sistema. . . . .	29
3.4 Caso de uso: Despliegue de mapa. . . . .	31
3.5 Caso de uso: Cambiar el mapa base. . . . .	32
3.6 Caso de uso: Visualización sobre el mapa de las entidades federativas del país. . . .	32
3.7 Caso de uso: Consultar pronóstico del tiempo para una entidad federativa . . . . .	33





## Índice de figuras

<b>Figura</b>	<b>Página</b>
1.1 Dominio del pronóstico numérico meteorológico. . . . .	2
1.2 Consulta y visualización del pronóstico operativo. Portal web del grupo IOA . . . .	3
1.3 Base de datos climatológica nacional del Centro de Investigación Científica y de Educación Superior de Ensenada Baja California (CICESE) . . . . .	5
1.4 Atlas Climático Digital de México - Unidad de Informática para las Ciencias Atmosféricas y Ambientales (UNIATMOS) . . . . .	5
1.5 Accuweather- Predicciones meteorológicas locales e internacionales . . . . .	6
1.6 Meteored . . . . .	6
2.1 Ejemplo de un servicio web en funcionamiento . . . . .	10
2.2 Arquitectura de un servicio web . . . . .	11
2.3 Estructura de un mensaje SOAP . . . . .	12
2.4 Estructura de un servicio REST . . . . .	13
2.5 Ejemplo de respuesta de un servicio REST con representación JSON. . . . .	14
2.6 Proceso de digitalización de mapas. . . . .	16
2.7 Ejemplo de petición a un servicio WMS. . . . .	17
2.8 Modelo raster y modelo vectorial. . . . .	21
3.1 Metodología empleada para el desarrollo del sistema. . . . .	26
3.2 Listado de las operaciones del sistema . . . . .	34
3.3 Listado de los casos de uso. . . . .	34
4.1 Flujo general de los datos . . . . .	40
4.2 Ejemplo de archivo de texto para la entidad federativa Distrito Federal . . . . .	41

4.3	Diagrama de componentes . . . . .	42
4.4	Ejemplo del pronóstico a un vistazo . . . . .	47
4.5	Ejemplo del pronóstico de zona digital . . . . .	48
4.6	Cabecera DWML . . . . .	49
4.7	Datos DWML parte 1 . . . . .	50
4.8	Datos DWML parte 2 . . . . .	51
4.9	VARIABLES DWML parte 1 . . . . .	52
4.10	VARIABLES DWML parte 2 . . . . .	53
4.11	VARIABLES DWML parte 3 . . . . .	54
4.12	VARIABLES DWML parte 4 . . . . .	55
4.13	Estructura de las carpetas en ZF . . . . .	57
4.14	Template creado de acuerdo a la especificación DWML . . . . .	58
4.15	Clase ZendControllerAction . . . . .	59
4.16	Función getTemperatures . . . . .	59
4.17	Cabecera del servicio web . . . . .	60
4.18	Cuerpo del servicio web . . . . .	60
4.19	Archivo war - GeoServer . . . . .	62
4.20	Ejemplo de solicitud HTTP GET WMS con GeoServer . . . . .	62
4.21	Interfaz de administración de GeoServer . . . . .	64
4.22	Importador de datos de GeoServer . . . . .	64
4.23	Proceso para importar datos con GeoServer . . . . .	66
4.24	Capas importadas con GeoServer . . . . .	67
4.25	Despliegue de una capa con GeoServer . . . . .	67
4.26	Visualización de una capa con GeoServer usando OPenLayers . . . . .	68
4.27	Interfaz gráfica de usuario . . . . .	69
4.28	Selección de una entidad federativa . . . . .	70
4.29	Despliegue del pronóstico detallado . . . . .	70
4.30	Menú de estados . . . . .	71
4.31	Pronóstico detallado . . . . .	72
4.32	Selector de capas . . . . .	72
A.1	Coordenadas geográficas . . . . .	77

## Introducción

El ser humano se encuentra expuesto a fenómenos atmosféricos extremos que representan una amenaza para su bienestar y entorno, motivos que han despertado su interés para comprender y tratar de explicar la ocurrencia de dichos eventos en su hábitat. Durante el siglo XX, gran parte del estudio desarrollado para la comprensión de fenómenos atmosféricos, impulsó el desarrollo de modelos numéricos de predicción del tiempo que hoy son una potente herramienta para estudiar fenómenos relacionados con la dinámica atmosférica.

El pronóstico numérico meteorológico del tiempo, mediante el uso de modelos numéricos busca predecir el estado de la atmósfera en un período cercano-futuro y en una localidad determinada haciendo uso de un conjunto de datos iniciales como:

- Temperatura
- Precipitación
- Viento
- Humedad
- Presión, etc.

Sin embargo, la naturaleza dinámica de la atmósfera y el entendimiento incompleto de los procesos hacen que los pronósticos sean menos seguros al incrementarse el rango temporal del pronóstico. En la actualidad la mayor parte de los pronósticos del tiempo se basan en modelos numéricos de mesoescala. Entre ellos destaca el modelo atmosférico *Weather Research and Forecasting*(WRF).

## 1.1 Motivación y descripción del problema

En el Centro de Ciencias de la Atmósfera (CCA) de la UNAM se realizan pronósticos sobre diferentes fenómenos meteorológicos y climáticos utilizando diversos modelos numéricos y herramientas de graficación. Estos pronósticos son realizados como parte de las actividades diarias de investigación y representan un esfuerzo de trabajo de varios años. En particular, en el grupo Interacción Océano-Atmósfera (IOA) una de las áreas de investigación es la modelación numérica del pronóstico meteorológico.

En el año 2007, el grupo IOA puso en marcha la primera versión del pronóstico operativo numérico meteorológico en la cual se pronosticaba a cinco días las variables de temperatura, viento en superficie, viento en diferentes niveles verticales, y precipitación acumulada. El pronóstico se realizaba con el modelo numérico meteorológico de mesoescala *Weather Research and Forecasting* (WRF) en su versión 2.2, con un dominio que abarcaba a toda la República Mexicana con una resolución aproximada de 20 km.

Para el año 2010, el grupo IOA puso en marcha la segunda versión del sistema de pronóstico, convirtiéndose en el pronóstico meteorológico oficial del CCA. En esta versión, fue actualizado el modelo WRF a la versión 3.0.1 y se agregó un segundo dominio para el centro de México con una resolución horizontal aproximada de 7km. También se realizaron mejoras en la operatividad, automatización, visualización y disponibilidad de gráficas de simulaciones históricas. La tabla 1.1 y la figura 1.1 muestra en resumen las características principales entre las dos versiones del Pronóstico Numérico Meteorológico WRF empleado por el CCA.

<b>Pronostico numérico meteorológico.</b>		
	<b>Primera versión</b>	<b>Segunda versión</b>
<b>Año</b>	2007	2010
<b>Número de versión</b>	2.2	3.0.1
<b>Dominio</b>	República Mexicana	Centro de México
<b>Resolución</b>	Aprox. 20 km	Aprox. 7 km

Cuadro 1.1: Características de las dos versiones del pronóstico numérico meteorológico.

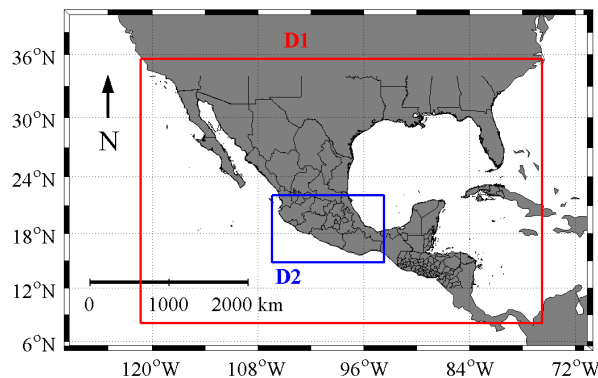


Figura 1.1: Dominio del pronóstico numérico meteorológico.

## 1.1. MOTIVACIÓN Y DESCRIPCIÓN DEL PROBLEMA

Actualmente, el grupo IOA a través de su portal web (<http://grupo-ioa.atmosfera.unam.mx/pronosticos>) pone a disposición del público los pronósticos diarios del tiempo o pronósticos operativos teniendo como dominios la República Mexicana y el Centro de México. Para el dominio del Centro de México las variables pronosticadas son: precipitación acumulada, precipitación acumulada cada 3 horas, temperatura y viento. Mientras que para el dominio de la República Mexicana las variables pronosticadas son: precipitación acumulada, precipitación acumulada cada 3 horas, temperatura, viento en superficie y viento a diferentes niveles (850mb, 700mb, 500mb, 250mb). La aplicación web del portal permite obtener para cada una de las variables, la consulta de pronósticos históricos junto con su pronóstico visualizado.

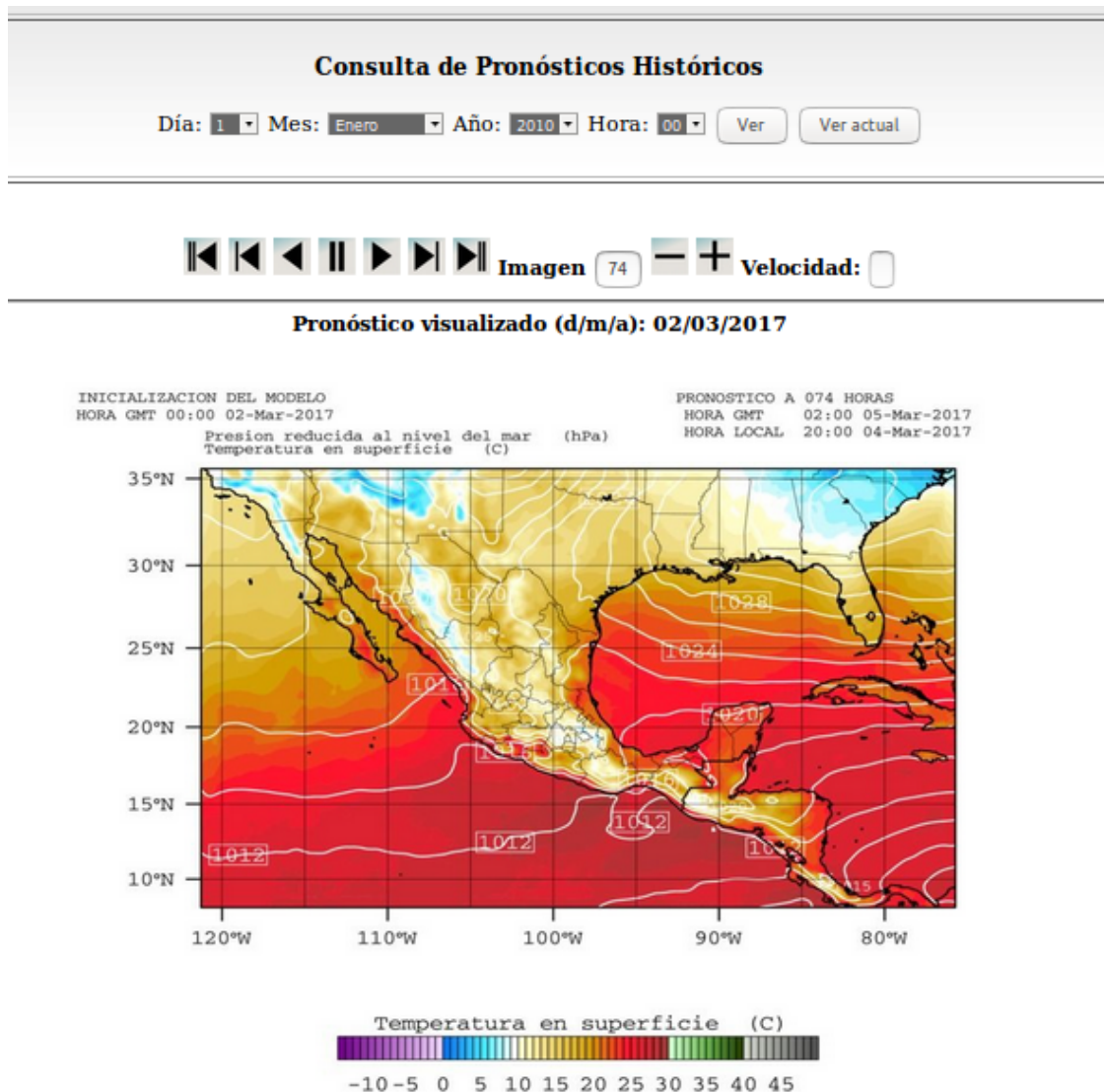


Figura 1.2: Consulta y visualización del pronóstico operativo. Portal web del grupo IOA

En la figura 1.2 se puede observar que la aplicación web permite a un usuario seleccionar un año que puede ir desde el 2010 hasta el 2017 para inmediatamente desplegar la visualización del pronóstico operativo a 5 días para el año seleccionado, para cada hora GMT y su correspondiente

hora local en el dominio y la variable elegida. No obstante, para un usuario no experto en el tema puede resultar difícil la interpretación de la información proporcionada, en especial si el usuario deseara conocer por ejemplo, la temperatura pronosticada para un punto o localidad en específico a una hora determinada.

Hoy en día, el grupo IOA está trabajando en una nueva versión del pronóstico operativo la cual considera desde la actualización del modelo WRF a la última versión hasta el desarrollo de nuevos productos. Los puntos que considera la nueva versión del pronóstico operativo son:

- Versión 3.8 del modelo WRF
- Mayor resolución horizontal (5 y 15 km)
- Condiciones iniciales de mayor resolución 0.25 (GFS)
- Nuevos productos
- Actualización de la página web
- Facilitar la visualización y comprensión de la información meteorológica al público no experto en el tema
- Hacer uso de aplicaciones web mapping para interpretar cualitativamente con símbolos meteorológicos las variables pronosticadas

## 1.2 Justificación

Se debe tomar en cuenta que la visualización de datos a través de mapas en la web ha ido en constante desarrollo conforme el pasar de los años junto con el surgimiento y uso de nuevas tecnologías web que permiten ya no solo a las instituciones de investigación y estudio científico sino también a empresas privadas de software y consultorías, elaborar aplicaciones web mapping de gran calidad que proporcionan información para la toma de decisiones.

En las figuras 1.3 y 1.4 se muestran diferentes ejemplos de aplicaciones web mapping.

Estas aplicaciones web mapping son ejemplos reales de diferentes soluciones para mostrar información en un mapa por parte de instituciones como el Centro de Investigación Científica y de Educación Superior de Ensenada Baja California (CICESE) y la Unidad de Informática para las Ciencias Atmosféricas y Ambientales (UNIATMOS) del CCA. Como se puede observar, no existe una tipología de mapas estándar y cada institución puede seleccionar como se representa mejor su información.

Existen también compañías que se dedican al desarrollo y venta de aplicaciones web mapping de predicción meteorológica que permiten consultar el estado del tiempo antes de salir a realizar actividades extremas y que además funcionan en dispositivos móviles. Algunos ejemplos de este tipo de aplicaciones se muestran en las figuras 1.5 y 1.6.

Este tipo de aplicaciones por lo general cuentan con sistemas de alarma para fenómenos adversos que envían notificaciones a dispositivos móviles según la geoposición. Algunos de estos

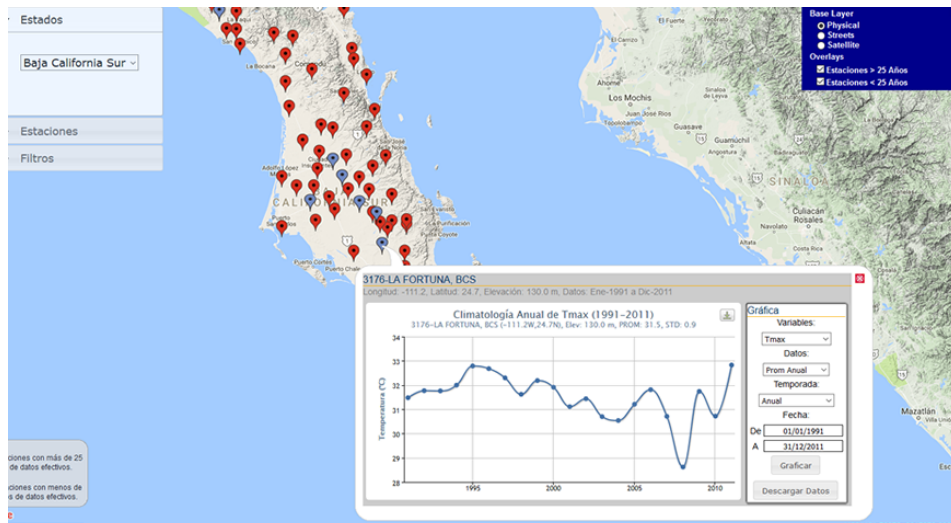


Figura 1.3: Base de datos climatológica nacional del Centro de Investigación Científica y de Educación Superior de Ensenada Baja California (CICESE)

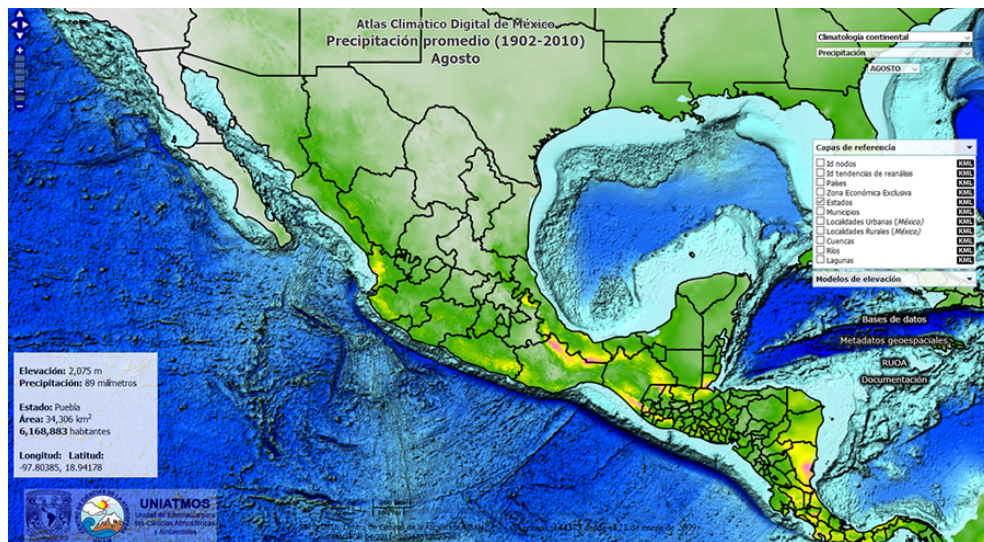


Figura 1.4: Atlas Climático Digital de México - Unidad de Informática para las Ciencias Atmosféricas y Ambientales (UNIATMOS)

sistemas proporcionan servicios web con las salidas del pronóstico del tiempo obtenidas con algún modelo numérico de mesoescala como parte de una renta mensual.

Por lo anterior, se propone que el grupo IOA de difusión del pronóstico meteorológico y sus datos (para las variables citadas anteriormente) que se lleva acabo dentro del CCA en formato digital accesible por Internet que permita la disponibilidad y utilidad de los mismos. Una vez que los datos se entregan donde se necesitan, su forma digital aumenta su utilidad dando la posibilidad a nuevos usuarios y aplicarlos a una amplia gama de problemas.



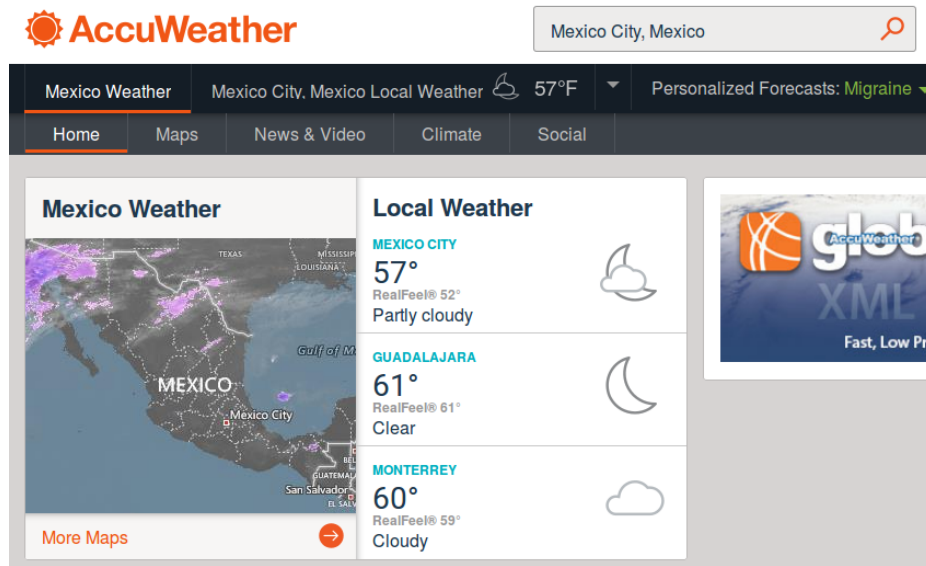


Figura 1.5: Accuweather- Predicciones meteorológicas locales e internacionales



Figura 1.6: Meteored

Las posibles aplicaciones de los datos generados por el pronóstico meteorológico del grupo IOA del Centro puede ser variada, desde mostrarlos en páginas web personales, redistribuirlos en un paquete de valor agregado, hasta integrarlos en sistemas de apoyo a la toma de decisiones sensibles al clima. Cada uno de estos ejemplos sugiere una era en la que los servicios digitales máquina a máquina permiten al pronóstico meteorológico del grupo IOA difundir sus datos de



manera más rápida, más eficiente y en formas adecuadas para una mayor gama de usuarios a través de servicios web.

## 1.3 Objetivos

### 1.3.1 Objetivo general

Diseñar, desarrollar e implementar una aplicación web de mapas (*web mapping application*) dinámica y amigable que permita la difusión masiva de productos meteorológicos del pronóstico del CCA-IOA usando software libre de visualización de mapas georeferenciados y símbolos meteorológicos.

### 1.3.2 Objetivos particulares

- Diseñar una aplicación web de mapas haciendo uso de patrones de diseño web que faciliten el uso de la aplicación.
- Crear un servicio web para la distribución de los datos obtenidos por el modelo WRF utilizando el esquema para la creación de servicios web meteorológicos *Digital Weather Markup Language (DWML)*, proporcionado por la *National Oceanic and Atmosphere Administration (NOAA)*.
- Implementar un servidor de mapas que proporcione servicios WMS (*Web Map Service*) para manipular las capas base de la aplicación e información geográfica adicional.
- Desarrollar un módulo de visualización web para el pronóstico del día y de cinco días haciendo uso de símbolos meteorológicos, gráficas e información tabular.
- Caracterizar con símbolos meteorológicos la información pronosticada de las variables de temperatura mínima, temperatura máxima, precipitación y viento en superficie.
- Desarrollar la compatibilidad de la aplicación web con dispositivos móviles haciendo uso de frameworks de diseño.

### 1.3.3 Contribuciones

El resultado de este trabajo de tesis será una aplicación web de mapas dinámica que dará continuidad a la investigación desarrollada en versiones anteriores del sistema de pronóstico meteorológico operativo del CCA-UNAM, la cual permite el acceso a la información meteorológica de manera más clara al público en general. Además, da difusión masiva en Internet de los principales productos meteorológicos desarrollados por el grupo IOA del CCA-UNAM.

### 1.3.4 Organización de la tesis

Este documento muestra dos partes claramente diferenciadas. Una de ellas de carácter teórico y otra de contenido práctico y explicativo de un desarrollo de software.

En la parte teórica se presentan conceptos para la comprensión de los servicios web y las aplicaciones *web mapping*, tales como: Servicios Web, ¿Qué funcionalidades tienen?, ¿Cuáles son sus elementos principales?, se abarca también conceptos básicos de aplicaciones *web mapping*,

como la representación de su información mediante el uso de modelos *raster* o Vectoriales hasta el *Open Geospatial Consortium (OGC)*. Se finaliza la parte teórica haciendo mención a los conceptos fundamentales en la modelación numérica meteorológica de mesoescala y el formato *netCDF* de datos científicos empleado para representarlos; así como los operadores *netCDF* empleados para manipularlos. Para el desarrollo de la parte práctica se analizan los requisitos, los diagramas de casos de uso, el diseño de la aplicación y la metodología empleada que definen la arquitectura del sistema. La tesis termina mostrando el software generado y las conclusiones sobre los puntos más importantes del desarrollo en el trabajo realizado.

## Marco teórico

La tesis se dividió en la creación de un servicio web para la difusión del pronóstico meteorológico para las variables de temperatura, precipitación y viento en superficie a 5 días obtenido por el modelo WRF y en el consumo del servicio web mediante una aplicación *web mapping* que facilite la comprensión de las variables pronosticadas. Para ello, se comienza explicando que son los servicios web (*web services*), para qué sirven, cómo funcionan y los tipos de servicios web que existen.

### 2.1 Servicios web (*Web Services*)

Previamente se utilizaban tecnologías como RMI, CORBA y DCOM para crear aplicaciones cliente servidor. Estas tecnologías se empleaban en sistemas distribuidos altamente acoplados, en los que tanto el servidor como el cliente dependían uno del otro. Sin embargo, al usar estas técnicas surgieron problemas tanto de compatibilidad como de seguridad. Un ejemplo de ello es, que las técnicas anteriores producían un tráfico que era bloqueado mediante firewalls y servidores proxy. Las aplicaciones altamente acopladas como las que se producían con el uso de estas tecnologías se utilizaban sobre todo para la intranet debido a que las plataformas y tecnologías usadas eran conocidas tanto del lado del servidor como del cliente.

Los servicios web son un conjunto de aplicaciones o tecnologías con capacidad para

interoperar en Internet. Estas tecnologías intercambian datos entre ellas con el fin de ofrecer servicios de una forma estándar a través de la web. Los clientes no tienen ningún conocimiento previo de los servicios web antes de que realmente los utilicen, por lo tanto, los servicios web son independientes de la plataforma y además son ligeramente acoplados.

Existen dos tipos de servicios web, los basados en el principio *SOAP* y los basados en el principio *REST*. Aplicaciones como las conferencias a través de Internet o las propias aplicaciones web se pueden desarrollar utilizando servicios web tanto los basados en *SOAP* como en *REST*.

La figura 2.1 muestra un ejemplo de servicio web para un usuario que solicita información sobre un viaje que desea realizar haciendo una petición a una agencia de viajes que ofrece sus servicios a través de Internet.

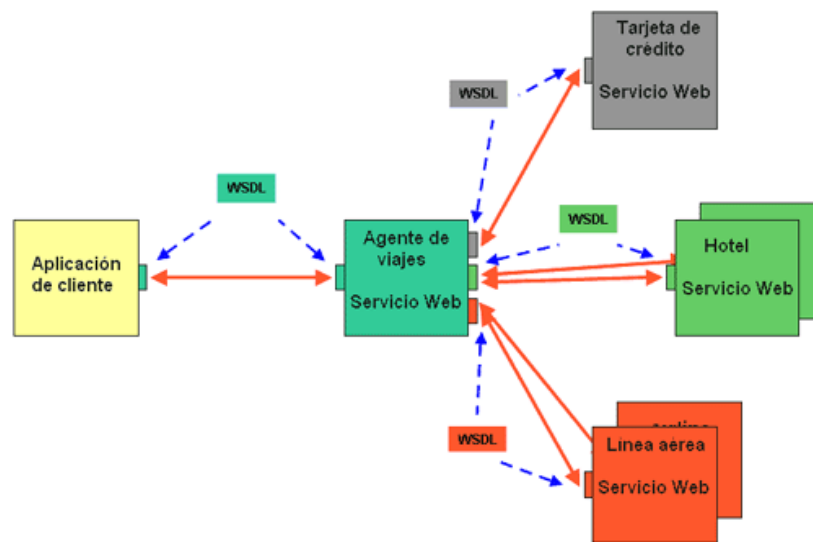


Figura 2.1: Ejemplo de un servicio web en funcionamiento

Para proporcionar al cliente la información que necesita, la agencia de viajes solicita a su vez información a otros servicios web en relación con el hotel y la compañía aérea. La agencia de viajes obtendrá información de estos servicios, lo que la convierte a su vez en cliente de otros servicios web que le van a proporcionar la información solicitada sobre el hotel y la línea aérea. Por último, el usuario realizará el pago del viaje a través de la agencia de viajes que servirá de intermediario entre el usuario y el servicio web que gestionará el pago.

### 2.1.1 SOAP (*Simple Object Access Protocol*)

La arquitectura de servicios web basada en *SOAP* puede verse en la figura 2.2. La cual define tres entidades:

- Proveedor de servicios
- Registro de servicios
- Solicitante o consumidor de servicios

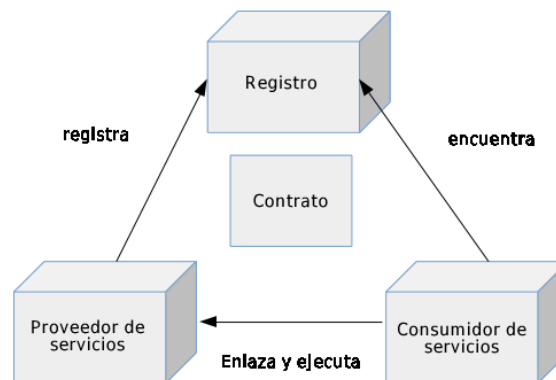


Figura 2.2: Arquitectura de un servicio web

El proveedor de servicios es el servicio o la entidad direccionable de red que acepta y ejecuta la solicitud del consumidor. El consumidor de servicio es una aplicación, servicio o algún otro tipo de módulo de software que requiere un servicio. Un registro de servicios es una red que contiene los servicios disponibles. El consumidor del servicio encuentra la descripción del servicio en el registro que es publicado por el proveedor de servicios. Utilizando esta descripción, el consumidor comienza a interactuar con el servicio.

Las comunicaciones entre estas entidades se basan en *XML* y *SOAP*. Los mensajes SOAP están compuestos por un sobre, encabezado y cuerpo. El elemento de sobre identifica el documento *XML* como un mensaje *SOAP*. Un elemento de encabezado contiene información de llamada y respuesta. Los mensajes y las invocaciones de métodos se definen como documentos *XML* y se envían a través de un protocolo de transporte *SMTP*, *FTP* y *HTTP*. La figura 2.3

muestra el esquema de la estructura de un mensaje *SOAP* para el ejemplo del usuario que desea saber su reserva e itinerario del ejemplo de servicio web anterior.

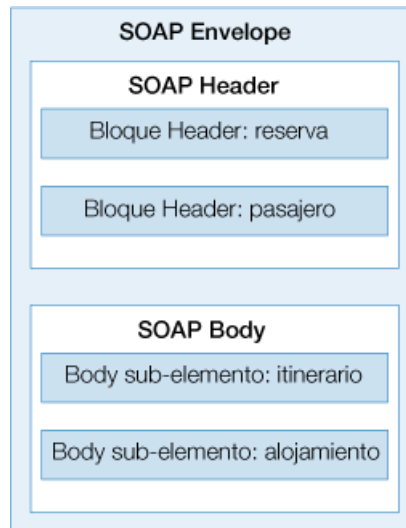


Figura 2.3: Estructura de un mensaje SOAP

### 2.1.2 REST (*Representational State Transfer*)

El término Transferencia de Estado Representacional o *REST* por sus siglas en inglés, fue introducido por RoyFielding. La arquitectura de estilo *REST* es la arquitectura de cliente - servidor en la que el cliente envía una solicitud al servidor y luego el servidor procesa las respuestas de solicitud y devolución. Estas solicitudes y respuestas se basan en la transferencia de representaciones de recursos. Un recurso es algo identificado por un *URI (Uniform Resource Identifier)*.

La representación del recurso suele ser un documento que captura el estado actual o deseado de un recurso. El lenguaje *REST* se basa en el uso de sustantivos y verbos. *REST* no requiere formato de mensajes como por ejemplo el sobre (*envelope*) y el encabezado (*header*) los cuales son requeridos en mensajes *SOAP*. Por lo tanto, no se requiere de un análisis del *XML* y en consecuencia el ancho de banda requerido es menor. El principio de diseño de un servicio *REST* es el siguiente:

En un servicio web de tipo *REST* cada transacción es independiente y no está relacionada con la transacción anterior ya que todos los datos requeridos para procesar la solicitud están

contenidos únicamente en esa solicitud, los datos de sesión del cliente no se mantienen en el servidor, por lo tanto; las respuestas del servidor son también independientes. Estos principios hacen que la aplicación *REST* sea simple y ligera. La aplicación web que sigue la arquitectura *REST* se le llama servicio web *RESTful*. Los servicios web *RESTful* utilizan los métodos *http*, *GET*, *PUT*, *POST* y *DELETE* para recuperar, crear, actualizar y eliminar los recursos. La figura 2.4 muestra la arquitectura de un servicio *REST*.

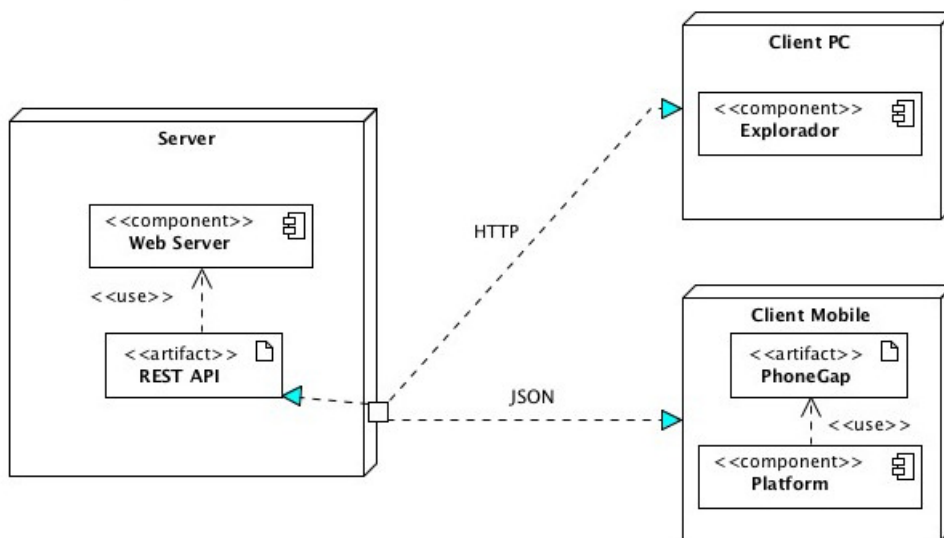


Figura 2.4: Estructura de un servicio REST

Conceptualmente hablando, lo correcto sería que cada petición responda realizando los métodos descritos. Sin embargo, en muchos sistemas reales, las funciones que deberían de realizar las peticiones de tipo *PUT* y *DELETE* son realizadas por peticiones *POST*. La petición *POST* se utiliza tanto para crear, borrar o actualizar un recurso. Y por lo tanto las peticiones *PUT* y *DELETE* no son utilizadas.

La implementación de un recurso decide que información es visible o no desde el exterior, y que representaciones de dicho recurso se soportan. Existen 3 posibles representaciones más comunes para un recurso:

- *XHTML*
- *XML*

- *JSON*

Pero no sólo se puede representar el recurso como datos estructurados. Se puede pedir por ejemplo, una representación en formato imagen. Esta variedad de representación de los recursos del servicio permite que puedan ser usados por clientes diseñados en diferentes lenguajes de programación y que pueden ejecutarse en arquitecturas completamente distintas. El uso de los tipos *MIME* y del encabezado *HTTP-Accept* es un mecanismo conocido como negociación de contenido, el cual le permite a los clientes elegir qué formato de datos pueden leer, y permite un desacoplamiento de datos entre el servicio y las aplicaciones que lo consumen.

Un ejemplo de servicio web *RESTful* sería la petición *GET* a un recurso “usuario“ de twitter:

```
http://api.twitter.com/1/users/show.json?screen_name=twitter
```

y la respuesta, con la representación JSON del recurso es la siguiente:

```
{ "id": 783214, "id_str": "783214", "name": "Twitter", "screen_name": "twitter", "location": "San Francisco, CA", "url": "http://blog.twitter.com/", "description": "Your official source for news, updates and tips from Twitter, Inc.", "protected": false, "followers_count": 19286285, "friends_count": 121, "listed_count": 77531, "created_at": "Tue Feb 20 14:35:54 +0000 2007", "favourites_count": 22, "utc_offset": -28800, "time_zone": "Pacific Time (US & Canada)", "geo_enabled": true, "verified": true, "statuses_count": 1588, "lang": "en", "status": { "created_at": "Wed May 22 19:36:13 +0000 2013", "id": 337290822204653569, "id_str": "337290822204653569", "text": "Make your Twitter account more secure with login verification, in 4 easy steps: ..." }
```

Figura 2.5: Ejemplo de respuesta de un servicio REST con representación JSON.

De todo esto, podemos observar que el uso de mensajes *SOAP* requiere un gran ancho de banda, codificación y decodificación de mensajes *SOAP* basados en XML los cuales consumen recursos, dando como resultado gastos de funcionamiento inadecuados. Por lo tanto, para el trabajo de tesis propuesto se utilizan servicios web *RESTful* como alternativa para obtener un mejor rendimiento en tiempo de procesamiento.



## 2.2 Aplicaciones web de mapas (*web mapping applications*)

Como se mencionó al inicio de este capítulo, la otra componente de la tesis consiste en la creación de un visualizador de mapas en la web que será el cliente que consuma los recursos proporcionados por el servicio web *RESTful* para de esta forma visualizar los datos obtenidos del pronóstico meteorológico numérico de mesoescala (WRF) presentando dicha información de manera gráfica. Por lo tanto, resulta importante entender los conceptos que se derivan de las aplicaciones web de mapas.

### 2.2.1 Mapas en Internet (*Internet mapping*)

La aparición de los mapas en Internet comenzó poco después del surgimiento de la *World Wide Web (WWW)*. Las primeras aplicaciones que mostraban mapas en la Internet eran muy rudimentarias, tenían la capacidad de presentar un mapa del mundo, haciendo zoom en escalas predefinidas y podían controlar la visibilidad de ríos y los bordes de algunas figuras.

Técnicamente, la capacidad de la WWW en ese tiempo era la de crear una página web en la cual era incrustada una imagen que representaba un mapa. La interacción entre el usuario y el mapa era proporcionada por un código informático (*Common Gateway Interface* o CGI) que se desarrollaba en el servidor web. Cada vez que el usuario hacía clic en uno de los enlaces de la página, el navegador web del usuario enviaba una solicitud al servidor.

La solicitud codificaba las coordenadas del área en las que el usuario estaba interesado y algunas otras opciones como por ejemplo, las capas que se querían desplegar. Una vez que el servidor recibía la solicitud ejecutaba el CGI necesario para producir la página HTML y la imagen asociada que representaba el nuevo mapa, y después transfería estos nuevos archivos a través de Internet hacia la computadora del usuario.

Una vez que los archivos eran recibidos, el navegador del usuario debía presentar los resultados en la pantalla de la computadora. Este modo de interacción daba lugar a un retraso de unos pocos segundos entre la acción del usuario (el clic en el mapa) y la representación del mapa en la pantalla con un refresh visible en toda la ventana del navegador cuando la nueva página era descargada. Este modelo de interacción era el núcleo de la mayoría de las aplicaciones web de mapas.

La figura 2.6 proporciona un ejemplo del proceso de digitalización de un área en un sitio de mapas de Internet, usando este modo de interacción.

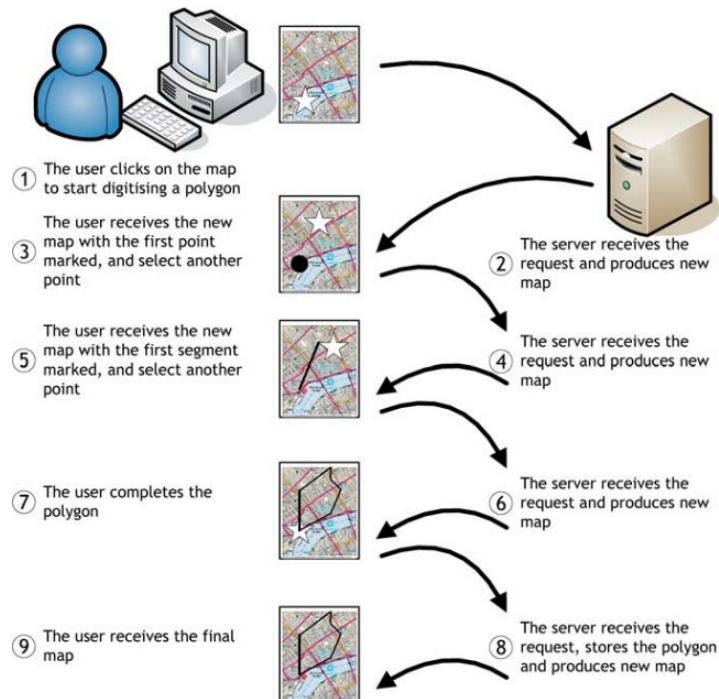


Figura 2.6: Proceso de digitalización de mapas.

### 2.2.2 Estándares OGC (*Open Geospatial Consortium*)

Los estándares OGC son el elemento final que es importante entender antes de migrar al concepto de web mapping.

La *Open Geospatial Consortium (OGC)* es una organización internacional sin fines de lucro comprometida con la creación de estándares de calidad abiertos para la comunidad geoespacial mundial. Estos estándares se hacen a través de un proceso de consenso y están libremente disponibles para que cualquiera pueda utilizarlos para mejorar el intercambio de datos geoespaciales del mundo.

El principal objetivo de dicho consorcio es la definición de estándares abiertos e interoperables dentro de la *World Wide Web* que faciliten el intercambio de la información geográfica, entre distintos sistemas, en beneficio de los usuarios y de la industria.

Dentro de las especificaciones definidas por el OGC destacamos dos por su importancia y

grado de implantación en el mundo de las aplicaciones web de mapas: los Servicios de Mapas Web (*Web Map Service (WMS)*) y los *Web Feature Services (WFS)*.

### 2.2.2.1 Servicios de Mapas Web (WMS)

Es un estándar OGC, que produce mapas de datos referenciados espacialmente de forma dinámica a partir de información geográfica. Este estándar internacional define un mapa como una representación de la información geográfica, en forma de un archivo de imagen digital conveniente para su exhibición en la pantalla del ordenador.

Está organizada en una o más capas que pueden visualizarse u ocultarse una a una. Se puede consultar cierta información disponible y las características de la imagen del mapa. Los mapas producidos por WMS se generan normalmente en un formato de imagen como PNG, GIF o JPEG, y opcionalmente como gráficos tal y como lo muestra la figura 2.7.

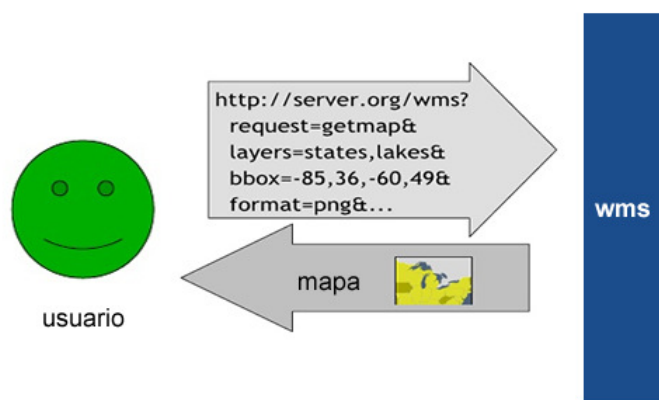


Figura 2.7: Ejemplo de petición a un servicio WMS.

Las operaciones WMS pueden ser invocadas usando un navegador estándar realizando peticiones en la forma de URLs (*Uniform Resource Locators*). El contenido de tales URLs depende de la operación solicitada. Concretamente, al solicitar un mapa, la URL indica qué información debe ser mostrada en el mapa, qué porción de la tierra debe dibujar, el sistema de coordenadas de referencia, y la anchura y altura de la imagen de salida.

Cuando dos o más mapas se producen con los mismos parámetros geográficos y tamaño de salida, los resultados se pueden solapar para producir un mapa compuesto. El uso de formatos de imagen que soportan fondos transparentes (GIF o PNG) permite que los mapas subyacentes sean visibles. Además, se pueden solicitar mapas individuales de diversos servidores.

### **2.2.2.2 Web Feature Service (WFS)**

Es un servicio estándar, que ofrece una interfaz de comunicación que permite interactuar con los mapas servidos por el estándar WMS, como por ejemplo, editar la imagen que nos ofrece o analizarla siguiendo criterios geográficos. Para realizar estas operaciones se utiliza el lenguaje GML (Lenguaje de Marcado Geográfico) que deriva del XML, y que es el estándar a través del cual se transmiten las ordenes WFS.

Los WFS pueden ser del tipo transaccional o no transaccional. La diferencia es que el transaccional sólo permite hacer consultas y recuperación de elementos geográficos, frente al no transaccional que permite además la creación, eliminación y actualización de estos elementos geográficos del mapa.

## **2.2.3 Sistemas de posicionamiento**

La disponibilidad de una mayor capacidad en las conexiones a Internet de uso doméstico, y la reducción de los costos de estos dispositivos crearon las condiciones necesarias para permitir un cambio en la entrega de información geográfica a través del Internet. Existen otros factores importantes que han contribuido al desarrollo de las aplicaciones web de mapas tal como las conocemos hoy en día, Pero dos tipos de tecnologías han tenido una importancia especial en su desarrollo: Los sistemas de posicionamiento global (GPS) y las tecnologías Web 2.0 particularmente *Asynchronous JavaScript and XML (AJAX)*.

### **2.2.3.1 Sistemas de posicionamiento Global (GPS)**

El sistema de posicionamiento global (GPS) es un sistema que permite determinar en toda la Tierra la posición de un objeto (persona o vehículo) con una precisión de 6 a 10 metros de exactitud.

### 2.2.3.2 Intercambio de información GPS

En los últimos años, la tecnología GPS ha impulsado a que más personas colecten información de diferentes lugares y suban esa información a sus computadoras. Sin embargo, no fue posible sino hasta la aparición de una norma de intercambio conocida como Formato de Intercambio GPS (*GP eXchange Format*) en el 2002 lo que permitió un mejor uso de la tecnología GPS ya que antes de esta norma, el intercambio de la información recabada con estos dispositivos era una tarea complicada que necesitaba de conocimientos técnicos de informática para la manipulación de los datos.

El GPX pertenece a una clase de normas y tecnologías que proporcionan la infraestructura para lo que llegó a ser conocido como la Web 2.0. Mientras que el uso popular al inicio de la Web se caracterizó por los sitios web que permitían la distribución de información de nuevas maneras, pero con una interacción limitada, las tecnologías de la Web 2.0 proporcionaron una interacción con el usuario mucho más rica.

Muchos factores han provisto una plataforma para esas nuevas aplicaciones. Primero, a finales de los noventa se incrementó la capacidad de transferencia de datos a costos muy bajos, permitiendo la proliferación de servicios de banda ancha hacia los usuarios caseros. En segundo lugar, las empresas que se dedican a la tecnología desarrollaron las normas que permitieron la transferencia de información entre sistemas distribuidos en diferentes lugares. Esta familia de estándares o normas (incluidas las normas OGC y GPX) se basaron en XML (*Extensible Markup Language*).

Otra innovación que integra los estándares basados en XML y que permite el desarrollo de aplicaciones más sofisticadas, fue la aparición de AJAX (*Asynchronous JavaScript*). La innovación más importante en el uso de esta técnica consiste en la capacidad de capturar información desde un servidor remoto a la espera de la acción de un usuario y proveer la interacción sin la necesidad de recargar toda la página Web. Esto cambió la experiencia del usuario dramáticamente e hizo que las aplicaciones Web fueran más similares a las aplicaciones de escritorio donde el modo de interacción entre el usuario y la aplicación es mejor.

Las aplicaciones geográficas basadas en AJAX se ven y se sienten muy diferentes. En primer lugar, el área de la pantalla en donde se despliega el mapa se ha incrementado dramáticamente,

mejorando así la facilidad del uso de mapas en la web de forma significativa. En segundo lugar, la capacidad de interactuar dentro del navegador del usuario cambió el modo de "hacer un clic y esperar a que la página se recarge" a la directa manipulación del mapa, un modo de interacción familiar en otras aplicaciones de escritorio, y más parecida a un SIG (ver apéndice B ) de escritorio.

### **2.2.4 Representación de la información**

En este apartado se explican las principales técnicas que utilizan las Aplicaciones Web de Mapas para representar la información georeferenciada, es decir, el mundo real, datos como carreteras, uso del suelo, contaminación, etc. Estos a su vez se pueden dividir en dos tipos: objetos discretos (ejemplo: casa, carretera) y continuos (ejemplo: nube de contaminación, acuíferos). Para almacenar estos datos hay dos formas principales de hacerlo, se trata del modelo raster y el vectorial, a continuación se explica cada uno de ellos para después realizar una comparación entre los mismos.

#### **2.2.4.1 Modelo raster**

La idea principal del modelo de datos raster, es una imagen digital dividida en mallas. Se divide el espacio en celdas regulares (píxeles) y cada una de ellas representa un único valor, el cual puede ser discreto, como el uso del suelo, o un valor continuo, como la temperatura, o un valor nulo, en caso de no existir información. Como ya hemos dicho la imagen queda como una rejilla de celdas, por lo tanto la posición de cada una viene implícita según el orden que ocupa en la malla, el espacio no es continuo sino que se divide en unidades discretas.

Este tipo de estructuras puede implicar un incremento del espacio de almacenamiento, ya que se guarda cada celda de la matriz sin tener en cuenta si se trata de un valor numérico o simplemente de un espacio vacío que representa no información significativa. Para obtener una descripción precisa de los objetos geográficos contenidos en la base de datos, el tamaño del píxel ha de ser reducido (en función de la escala), lo que hará a la malla tener una resolución alta. Sin embargo, a mayor número de filas y columnas se obtendrá más resolución, pero también, mayor esfuerzo en el proceso de captura de la información y mayor costo computacional al momento de procesarla.

Los datos raster se almacenan en diferentes formatos, como por ejemplo archivos estándar con formatos TIFF, JPEG, etc.

#### 2.2.4.2 Modelo vectorial

En este caso se modelan digitalmente las entidades del mundo real a partir de elementos geométricos. Para ello se utilizan vectores definidos por pares de coordenadas relativas a algún sistema cartográfico. Los principales componentes son:

- Punto o nodo: unidad básica para representar entidades geográficas que pueden ser expresadas por un único punto de referencia, en otras palabras, la simple ubicación.
- Línea: a partir de dos puntos, como mínimo para formar una línea. En este tipo de elementos se puede medir la distancia.
- Polígono: en este caso a partir de líneas o agrupaciones de puntos crearemos polígonos.

El interés de este tipo de representaciones se centra en la precisión de localización de los elementos geográficos sobre el espacio, donde los fenómenos a representar son discretos, es decir, de límites definidos.

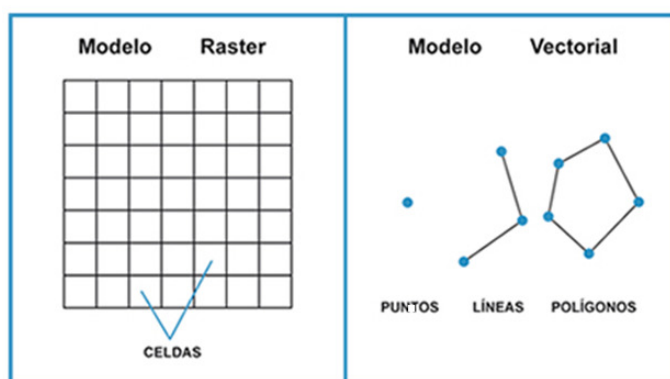


Figura 2.8: Modelo raster y modelo vectorial.

#### 2.2.4.3 Comparación entre modelo raster y modelo vectorial

Por la gran cantidad de información que se maneja en cada píxel, los modelos raster necesitan potentes computadoras con una gran capacidad en memoria y en disco duro.

Como resultado, los sistemas raster tienen mayor poder analítico que el vectorial en el análisis del espacio continuo, y por tanto, es idóneo para el estudio de fenómenos cambiantes en el espacio como las variables de los suelos, elevación del terreno, los índices de vegetación, precipitaciones, temperaturas, etc.

La segunda gran ventaja del modelo raster es que su estructura se ajusta de manera parecida a la de las computadoras. Por tal motivo, los modelos raster son más rápidos en la evaluación de problemas que impliquen diversas operaciones matemáticas. Es de nuevo, un excelente medio para explicar modelos medioambientales.

Mientras que los sistemas raster están predominantemente orientados al análisis espacial, los modelos vectoriales son eficientes en el almacenamiento de mapas, ya que sólo distinguen entre límites de características, y no lo que existe en el interior de las mismas.

<b>Modelo raster</b>	<b>Modelo vectorial</b>
<p>Ventajas</p> <ul style="list-style-type: none"> <li>• Estructura de datos sencilla.</li> <li>• La operación de superposición entre capas se realiza de forma fácil.</li> <li>• Representa mejor variabilidad espacial.</li> <li>• Es necesario para el manejo y modificación de imágenes digitales y satelitales.</li> </ul>	<p>Ventajas</p> <ul style="list-style-type: none"> <li>• Proporciona una estructura de datos compacta que ocupa menos espacio.</li> <li>• Codifica de manera más eficaz las relaciones topográficas entre elementos.</li> <li>• Está diseñado para trabajar con gráficos.</li> </ul>
<p>Desventajas</p> <ul style="list-style-type: none"> <li>• Ocupan más memoria.</li> <li>• Las relaciones topológicas son más difíciles de representar.</li> <li>• Los mapas resultantes son menos estéticos, este fenómeno puede superarse reduciendo el tamaño del pixel, pero incrementa en gran medida el volumen del archivo.</li> </ul>	<p>Desventajas</p> <ul style="list-style-type: none"> <li>• Tiene una estructura de datos más compleja que el raster.</li> <li>• La representación de mapas con elevada variabilidad es ineficiente.</li> <li>• Manejo y mejora de imágenes digitales y satelitales no pueden realizarse de una manera eficaz en el modelo vectorial</li> </ul>

Cuadro 2.1: Ventajas y desventajas de los modelos raster y vectorial.

Por último, aunque este trabajo no cubre el proceso empleado en la obtención del pronóstico



del tiempo mediante el modelo numérico de mesoescala WRF. Es importante dedicarle un espacio para conocer sus bases.

## 2.3 Modelo WRF

El modelo WRF (*Weather Research and Forecasting*) es un sistema de predicción numérica del tiempo de última generación, diseñado tanto para aplicaciones operacionales como para realizar investigación de la atmósfera. Tiene la posibilidad de realizar anidamientos múltiples, que sirven para poder alcanzar una resolución muy alta en zonas específicas y, a su vez, menos recursos de cómputo.

El desarrollo del modelo WRF ha sido producto de una labor conjunta realizada por organismos e instituciones como los siguientes: *National Center for Atmospheric Research (NCAR)*, *National Oceanic and Atmospheric Administration (NOAA)*, *National Centers for Environmental Prediction (NCEP) Forecast Systems Laboratory (FSL)*, *Air Force Weather Agency (AFWA)*, *Naval Research Laboratory (NRL)*, *University of Oklahoma* y *la Federal Aviation Administration (FAA)* entre otras.

El modelo proporciona a los usuarios la posibilidad de llevar a cabo tanto simulaciones numéricas con datos reales como a partir de condiciones idealizadas, pudiéndose utilizar de manera eficiente como pronóstico meteorológico operativo y como herramienta fundamental para el desarrollo de la investigación.



## Descripción de la aplicación

La metodología de software que se utilizó para estructurar, planificar y controlar el proceso de desarrollo de este sistema web, corresponde al modelo en cascada propuesto por Winston Royce W. El modelo en Cascada también conocido como Ciclo de Vida del software da las pautas que permiten la organización en el desarrollo de software a través de la implementación de sus características por etapas. De esta manera la estructura empleada para el desarrollo del sistema web quedó de la siguiente forma:

- Análisis de requerimientos
- Diseño
- Codificación
- Implementación

La figura 3.1 muestra la metodología usada así como las diferentes etapas que la conforman.

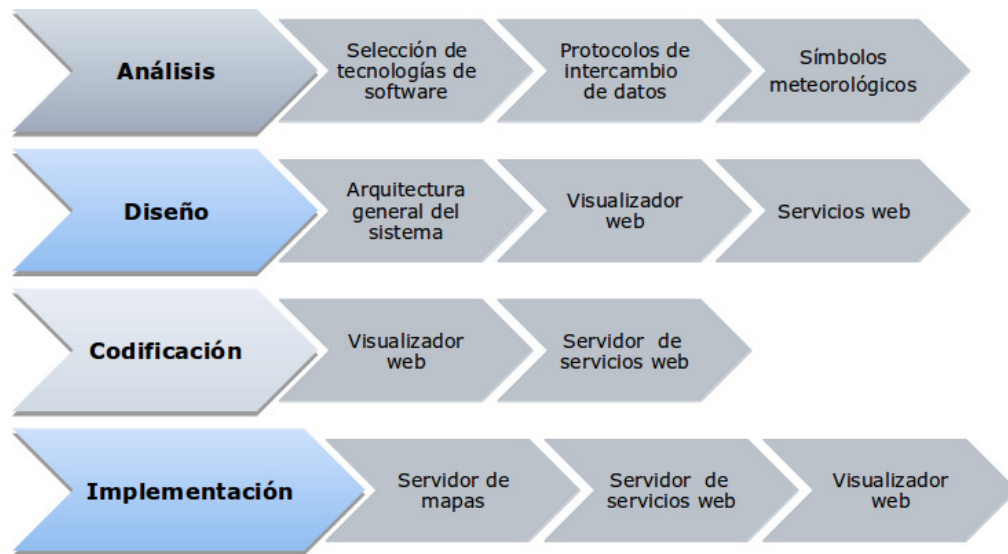


Figura 3.1: Metodología empleada para el desarrollo del sistema.

### 3.1 Planificación

En la fase de planificación se describió el enunciado del problema a desarrollar, la estrategia de solución empleada, se identificaron las operaciones del sistema y se incluyó la infraestructura con la que se cuenta.

#### 3.1.1 Definición del problema

Crear un aplicación web que permita la distribución de los datos obtenidos por el modelo *Weather Research and Forecasting (WRF)* que habilite a los desarrolladores de plataformas meteorológicas la creación de sus propias interfaces haciendo uso de la información atmosférica pronosticada en el CCA-UNAM. Además, la aplicación web debe de ser capaz también de interpretar con la ayuda de un mapa georeferenciado y símbolos meteorológicos, el pronóstico a 5 días de las variables de temperatura, precipitación y viento en superficie. La aplicación web debe de responder a los avances tecnológicos en materia de dispositivos móviles.

#### 3.1.2 Estrategia de solución

Con base en el enunciado anterior, para la distribución de los datos generados por el modelo WRF se propuso la creación de un servicio web basado en la arquitectura REST empleando el lenguaje *Digital Weather Markup Language (DWML)* proporcionado por la *National Oceanic*

and Atmospheric Administration (NOAA). Mientras que para la visualización de las variables meteorológicas pronosticadas se propuso el uso de la tecnología *web mapping* para la creación de un visualizador de mapas georeferenciado que funcione como el consumidor del servicio web.

### 3.1.3 Operaciones del sistema

- El sistema debe proporcionar un servicio web con los datos del pronóstico numérico meteorológico WRF que será consumido por el cliente o la aplicación *web mapping*.
- La aplicación *web mapping* debe permitir a un usuario las siguientes operaciones:
  - Mostrar un mapa de la República Mexicana georeferenciado
  - Navegación interactiva sobre el mapa (*zoom in*, *zoom out*, *paneo*)
  - Manejar diferentes capas base de la República Mexicana (proporcionadas por *Bing Map: Aerial, Satellite y Road*)
  - Visualización sobre el mapa de las entidades federativas del país
  - Menú complementario para acceder a las entidades federativas del país
  - Visualización de las variables pronosticadas por el modelo WRF para el pronóstico del día y de 5 días (temperatura máxima, temperatura mínima, precipitación y viento en superficie)
  - Uso de símbolos meteorológicos
  - Graficación de las variables pronosticadas

### 3.1.4 Infraestructura (hardware)

A continuación se describen los recursos de cómputo utilizados durante el desarrollo del sistema.

#### **Servidor Dell Power Edge T 610**

**Procesador:** Dos procesadores Intel Xeon X5675 con las siguientes características:

- Número de núcleos: 6
- Hiperthreading: SI

- Número de Threads: 12
- Velocidad de procesador: 3-06 GHz
- Memoria cache: 12 MB
- Conjunto de instrucciones: 64 bits

**Memoria:** 8DIMMs

- 64GB de Memoria optimizada para 2 procesadores
- Tipo de memoria DDR3 a 1333MHz,( 8x8 GB Dual Ranked LV RDIMMs)

**Disco Duro:** Cuatro discos duros SATA de 2TB HotPlug marca Hitachi con software RAID integrado. Configuración a dos RAID1 con controlador SAS 6/IR integrado

Partición	Tamaño	Ubicación	Formato
swap	8Gb	principio	área de intercambio
/	40Gb	principio	ext4
/var	50Gb	principio	ext4
/home	1.8 Tb	principio	ext4
/datos	1.8 Tb	principio	ext4

Cuadro 3.1: Configuración de los discos duros servidor DELL POWER EDGE T610

**Sistema Operativo:** Ubuntu Server 16.04.3 LTS

**Computadora ensamblada LUFAC**

- Procesador: Intel i7-4790: 3.60 GHz, 4 núcleos, 8 hilos, 8 MB caché
- Mother Board: Asus H97M-Plus: Formato mATX, cto. base Intel H97, 4 ranuras 240 terminales DDR3
- Memoria: 16 GB, DDR3, 1600 MHz

## 3.2 Análisis

En la parte del análisis del sistema se identificaron los requerimientos tanto funcionales como los no funcionales, se obtuvieron y se listaron los casos de uso, así como los criterios generales para la selección del software.

### 3.2.1 Requerimientos del sistema

Los requerimientos funcionales son declaraciones de los servicios que proveerá el sistema de manera en que este reaccionará a entradas particulares. En algunos casos los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer.

Identificador	Descripción
RF01	Navegación interactiva sobre el mapa.
RF02	Visualización sobre el mapa de las entidades federativas del país.
RF03	Visualización de las variables pronosticadas por el modelo WRF para el pronóstico del día y de 5 días (temperatura máxima, temperatura mínima, precipitación y viento en superficie.)
RF04	Graficación de las variables pronosticadas.

Cuadro 3.2: Requerimientos funcionales del sistema.

### 3.2.2 Requerimientos no funcionales

Los requerimientos no funcionales son los que especifican criterios para evaluar la operación de un servicio u operación de un sistema de software.

Identificador	Descripción
RNF01	La aplicación web debe ser probada en un ambiente de producción.
RNF02	La aplicación web debe estar accesible y operable en un 90 % del total del tiempo desde el momento en que se sube a Internet.
RNF03	La aplicación debe permitir la integración de otras variables del pronóstico numérico meteorológico WRF a futuro.

Cuadro 3.3: Requerimientos no funcionales del sistema.

### 3.2.3 Identificación y listado de los casos de uso

Un caso de uso es la captura de los requerimientos de un usuario para traducirlos a los requerimientos de la aplicación o sistema. Se pueden utilizar tantos casos de uso como operaciones tenga el sistema. Así, nuestra aplicación *web mapping* posee las siguientes operaciones:

- Ver mapa
- Cambiar mapa base
- Realizar zoom sobre mapa
- Realizar paneo sobre mapa
- Ubicación en el mapa
- Visualización sobre el mapa de las entidades federativas del país
- Consultar pronóstico del tiempo para alguna entidad federativa en específico ya sea haciendo click en la entidad posicionada en el mapa o mediante el botón de búsqueda de la aplicación

Cada una de las operaciones anteriores del sistema forman un caso de uso:

---

#### **Caso de uso:** Desplegar mapa

---

**Funcionalidad:** Visualización permitiendo la interacción sobre un mapa base, de los puntos o marcadores que representan las entidades federativas del país.

**Actores:** Usuario

**Precondición:** Ninguna

**Poscondición:** Se visualiza sobre un mapa base los puntos o marcadores que representan las entidades federativas del país.

---



**Proceso normal principal:**

- Se inicializa el mapa.
- Se definen las capas base. Se utilizarán como capas base las proporcionadas por Map Bings : “Aerial”, “Satellite”, “Road”, a fin de que el usuario pueda posteriormente seleccionar la que más le convenga.
- Se definen los WMS que proporciona los estados de la República Mexicana y los puntos que representan las entidades federativas del país, proporcionados por el servidor de mapas (*geoserver*).
- Se inicia la aplicación web de mapas, para ello se manda a llamar a los mapas base, las capas definidas para la aplicación y se establece el zoom inicial que permite visualizar el mapa completo.

---

Cuadro 3.4: Caso de uso: Despliegue de mapa.

---

**Caso de uso: Cambiar el mapa base**

---

**Funcionalidad:** El sistema deberá mostrar una lista con diferentes capas base permitiendo al usuario la elección de la capa base a utilizar.

**Actores:** Usuario

**Precondición:** El sistema debe tener acceso vía internet a los servidores de mapas. Además los servidores de mapas deberán tener disponibles sus servicios.

**Poscondición:** La capa base seleccionada por el usuario es mostrada en el visualizador.

---

**Proceso normal principal:**

- El usuario pulsa un botón para obtener la lista de capas base disponibles.
  - Se muestra una lista de capas base en las que se identifica la capa base usada. Se selecciona la capa deseada.
-

---

**Alternativas de proceso y excepciones:**

- En caso de que no se obtenga respuesta del servidor de mapas correspondiente a la capa seleccionada por el usuario, se mostrará una capa base por default. La capa base por default estará almacenado en un servidor de mapas local para evitar que dicha capa no este disponible.

---

Cuadro 3.5: Caso de uso: Cambiar el mapa base.

---

**Caso de uso:** Visualización sobre el mapa de las entidades federativas del país

---

**Funcionalidad:** Se muestran sobre el mapa, los puntos o marcadores que representan las entidades federativas del país.

**Actores:** Usuario.

**Precondición:** Las capas base deben de estar correctamente cargadas en el mapa.

**Poscondición:** Se muestra un combo box con todas las entidades federativas del país definidas en la aplicación.

---

**Proceso normal principal:**

- Se muestra una lista con los nombres de las entidades federativas del país.

---

**Alternativas de proceso y excepciones:**

- Los wms de las entidades federativas no están correctamente cargadas en la aplicación.
- La aplicación no mostrará la lista con las entidades federativas del país.

---

Cuadro 3.6: Caso de uso: Visualización sobre el mapa de las entidades federativas del país.

---

**Caso de uso:** Consultar pronóstico del tiempo para una entidad federativa

---

**Funcionalidad:** Se muestra la información referente al pronóstico del tiempo de una entidad federativa en forma de “pop-up” mostrando solamente la temperatura máxima para esa entidad en la hora consultada. Permite además, mediante un clic al vínculo: “detalle de pronóstico” la visualización completa del pronóstico actual de dicha entidad, así como el pronóstico a 5 días.

**Actores:** Usuario, servidor Web.

**Precondición:** El wms que contiene las entidades federativas debe de estar correctamente cargada en el mapa.

**Postcondición:** Se muestran los atributos del elemento “Entidad federativa” seleccionada, en forma de “atributo: valor.”

---

**Proceso normal principal:**

- Se selecciona una entidad federativa directamente en el mapa.
  - Se muestran los atributos del elemento “Entidad federativa”(temperatura máxima).
  - Se hace clic en el vínculo: “detalle del pronóstico”.
  - Se visualiza el pronóstico completo actual y a 5 días de dicha entidad federativa en una ventana emergente dentro del mapa.
- 

**Alternativas de proceso y excepciones:**

- Los wms de las entidades federativas no están correctamente cargadas en la aplicación.
  - La aplicación no mostrará la lista con las entidades federativas del país.
- 

Cuadro 3.7: Caso de uso: Consultar pronóstico del tiempo para una entidad federativa

La representación gráfica de las operaciones del sistema y de los casos de uso se muestran en las siguientes figuras: figura 3.2 y figura 3.3

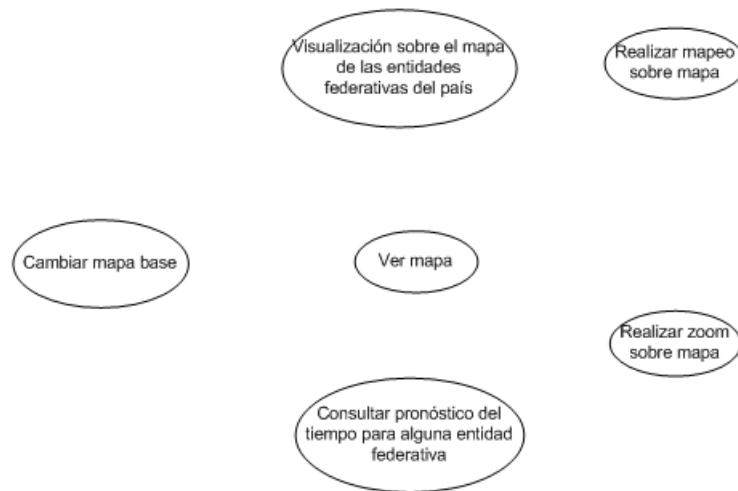


Figura 3.2: Listado de las operaciones del sistema

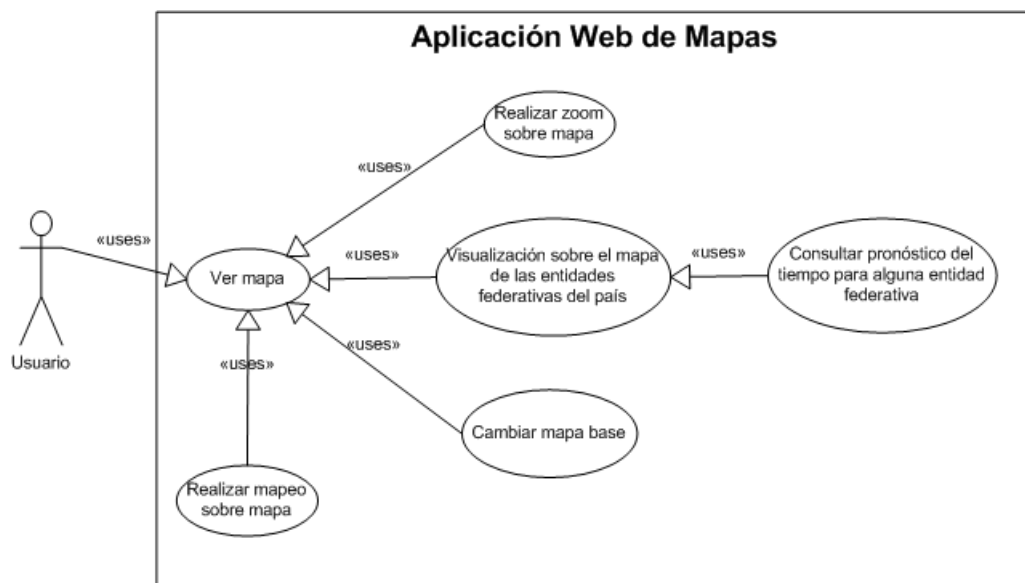


Figura 3.3: Listado de los casos de uso.

### 3.2.4 Criterios generales para la selección de software

Para la elección de las tecnologías de software a utilizar en el sistema web, se tomó en cuenta el modelo general para la implementación exitosa de aplicaciones web de mapas *web mapping applications* propuesto por A. Alesheikh, en el año 2002. En el cual se establecen lineamientos a considerar al momento de elegir el software para el desarrollo de aplicaciones web, el modelo abarca consideraciones de rendimiento, accesibilidad y economía tales como:

- Uso de protocolos estándar aceptados por la industria (OGC , W3C, etc)
- Portabilidad
- Procesamiento de los datos el mayor tiempo posible del lado del servidor
- Soporte y documentación
- Fácil de aprender, instalar y mantener
- Eficiencia en la gestión de memoria para responder a las peticiones HTTP
- Seguridad

Tomando en consideración los puntos del modelo, se eligieron las siguientes tecnologías de software para el desarrollo de esta aplicación:

#### 3.2.4.1 Servidor WEB HTTP Apache



El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la *Apache Software Foundation*.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociación de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Tiene amplia aceptación en la red. Desde 1996 es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70 % de los sitios web en el mundo.

Apache es usado primariamente para enviar páginas web estáticas y dinámicas en la *World Wide Web*. Es el componente de servidor web en la popular plataforma de aplicaciones *LAMP*, junto a *MySQL* y los lenguajes de programación *PHP*, *Perl* y *Python* entre otros.

La mayor parte de su configuración se realiza en el archivo *apache.conf* o *httpd.conf* según el sistema donde se esté ejecutando. Cualquier cambio en éste archivo requiere reiniciar el servidor, o forzar la lectura de los archivos de configuración nuevamente.

La licencia Apache (*Apache License* o *Apache Software License* para versiones anteriores a 2.0) es una licencia de software libre creada por la *Apache Software Foundation (ASF)*. La licencia Apache (con versiones 1.0, 1.1 y 2.0) requiere la conservación del aviso de copyright y el *disclaimer*, pero no es una licencia copyleft, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.

Como cualquier otra de las licencias de software libre, la Licencia Apache permite al usuario del software la libertad de usarlo para cualquier propósito, distribuirlo, modificarlo, y distribuir versiones modificadas de ese software. La licencia Apache sólo exige que se mantenga una noticia que informe a los receptores que en la distribución se ha usado código con la licencia Apache.

### 3.2.4.2 Lenguaje de programación PHP



PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en la interpretación del lado del servidor (*server side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

PHP es un acrónimo recursivo que significa PHP *Hypertext Preprocessor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf en 1994 y publicado bajo la *PHP License*. La *Free Software Foundation* considera esta licencia como software libre.

Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

#### 3.2.4.3 Zend Framework



Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100 % orientado a objetos. En la estructura de los componentes de ZF; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como “use-at-will” (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un framework de aplicaciones web a combinarse. ZF ofrece una implementación MVC, una abstracción de base de datos, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos.

También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los servicios web más populares.

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como *Google*, *Microsoft* y *StrikeIron* se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework.

#### 3.2.4.4 Servidor de Mapas



GeoServer es un servidor de código abierto escrito en Java que permite a los usuarios compartir y editar datos geospaciales. Diseñado para la interoperabilidad, publica datos de cualquier fuente de datos espaciales usando estándares abiertos. GeoServer ha evolucionado hasta llegar a ser un método sencillo para conectar información geoespacial existente como Google Earth y NASA World Wind.

Geoserver es desarrollado, probado y soportado por diversos grupos de individuos y organizaciones de todo el mundo, además sirve de implementación de referencia del estándar *Open Geospatial Consortium (OGC)*, *Web Feature Service (WFS)*, y también implementa las especificaciones de *Web Map Service (WMS)* y *Web Coverage Service (WCS)*.

### 3.2.4.5 Entorno de desarrollo del cliente web

Para construir la interfaz de la aplicación web de mapas se utiliza HTML5/JavaScript/Ajax y CSS3. En específico, se hace uso de los frameworks de diseño:

- Bootstrap en su versión 3.5: es un framework o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML5 y CSS3, así como extensiones de JavaScript opcionales y adicionales.
- Less (lenguaje de hojas de estilo): es un lenguaje dinámico que puede ser compilado para producir Hojas de Estilo en Cascada (CSS) y que puede ser ejecutado tanto del lado del cliente como del lado del servidor. Less proporciona los siguientes mecanismos: variables, anidamiento, operadores y funciones.

Mientras que para la navegación sobre la Cartografía se usará la librería:

- OpenLayers 3: OpenLayers es una biblioteca JavaScript de código abierto para crear mapas web interactivos, visibles en los navegadores web más populares (Mozilla Firefox, Google Chrome e Internet Explorer). Como es una biblioteca del lado del cliente, no requiere software especial del lado del servidor. Originalmente fue desarrollado por MetaCarta como respuesta, en parte, a Google Maps.



## Diseño e implementación del sistema

### 4.1 Diseño

En este apartado se describe el diseño arquitectónico de la aplicación.

#### 4.1.1 Arquitectura

La arquitectura que se implementó es de tipo web y está claramente dividida en la parte cliente y servidor. El cliente, quien además consume el servicio web con la información del modelo WRF es la aplicación *web mapping* que es desplegada por medio de un navegador web (*Safari, Opera, Mozilla FireFox o Google Chrome*) el cual hace las peticiones al servidor de mapas (*GeoServer*) para visualizar las capas e información adicional si la hay. Todo esto a través de peticiones HTTP mediante el servidor de páginas web Apache. También se encarga de mostrar mediante gráficas e información tabular los datos obtenidos por el servicio web.

#### 4.1.2 Flujo de datos

El flujo de información comienza cuando se accede a la aplicación por medio de una *url*, mostrando el visualizador de mapas con una capa satelital de tipo raster cargada al inicio y por defecto proporcionada por *Bing Maps*. También se cargan dos imagenes de tipo vectorial; una de las imagenes permite la visualización del límite político del país y la otra coloca marcadores

que representan las entidades federativas del país en el mapa. La figura 4.1 muestra el flujo general de los datos en el sistema web.

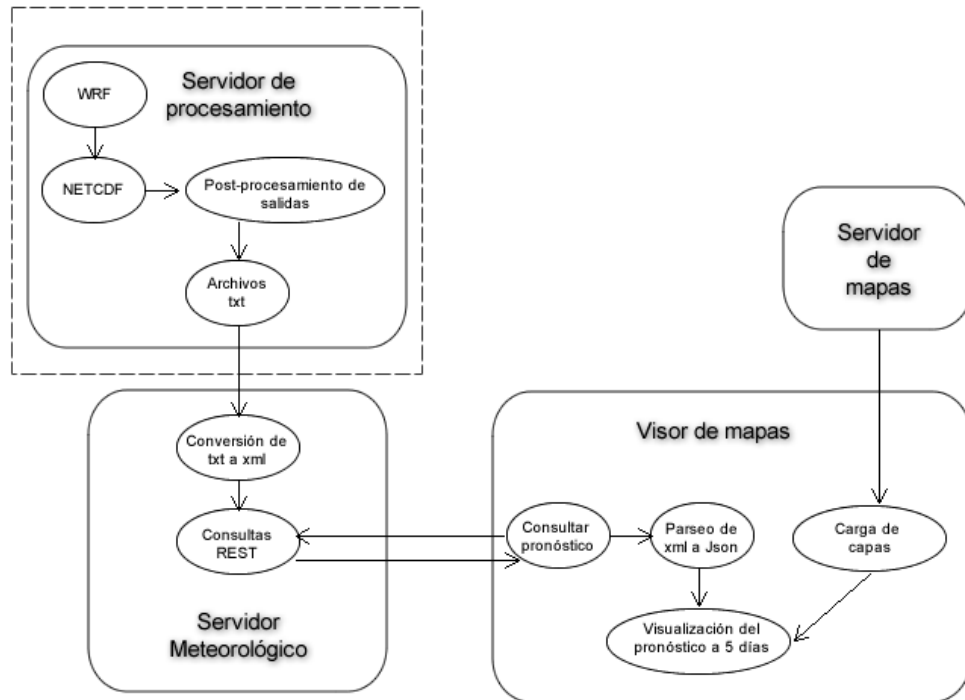


Figura 4.1: Flujo general de los datos

Cuando el usuario selecciona una entidad federativa (mediante un clic o por el menú de búsqueda) se hace una petición a un servidor que llamaremos servidor “meteorológico”, el cual se encarga de proporcionar el servicio web REST con los datos del pronóstico meteorológico obtenidos por el WRF. Los datos se envían a la aplicación web de mapas para ser procesados y mostrados con la ayuda de javascript y las librerías de OpenLayers tanto en una ventana emergente de texto que se despliega al haber hecho clic en la entidad federativa, así como en una caja de texto ubicada dentro del visor.

### 4.1.3 Diagrama de componentes

La arquitectura del diagrama de componentes del sistema web es la de un cliente servidor. La parte del servidor esta formada por tres componentes:

- Servidor de procesamiento: el cual se encarga mediante un post procesamiento aplicado

al formato netCDF obtenido por el modelo WRF de tomar las variables de temperatura, precipitación y viento en superficie y convertirlas a un archivo de texto. En la figura 4.2 se muestra un ejemplo de archivo de texto para una entidad federativa (Distrito Federal) con la información obtenida del WRF.

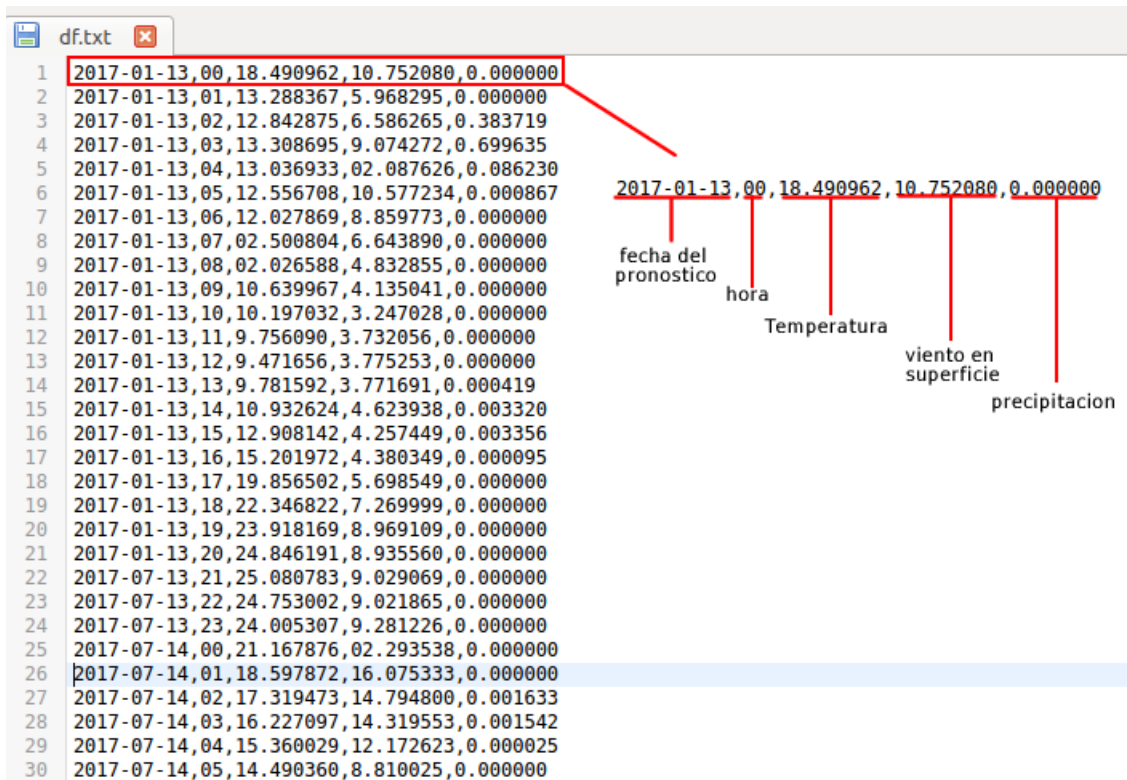


Figura 4.2: Ejemplo de archivo de texto para la entidad federativa Distrito Federal

- Servidor meteorológico: implementado en *ZendFramework* el cual hace la conversión del archivo de texto proporcionado por el servidor de almacenamiento a un archivo xml que pueda ser entregado mediante un servicio web REST.
- Servidor de mapas: implementado en *GeoServer* que proporciona los wms necesarios para la aplicación *web mapping*.

Mientras que la parte del cliente está formada por

- Visor de mapas: implementado en javascript y *OpenLayers* que hace las peticiones tanto al servidor de mapas (*GeoServer*) como al servidor meteorológico (*ZendFramework*) y que

muestra la información del pronóstico de manera gráfica.

De esta forma, la figura 4.3 muestra el diagrama de componentes para el sistema web

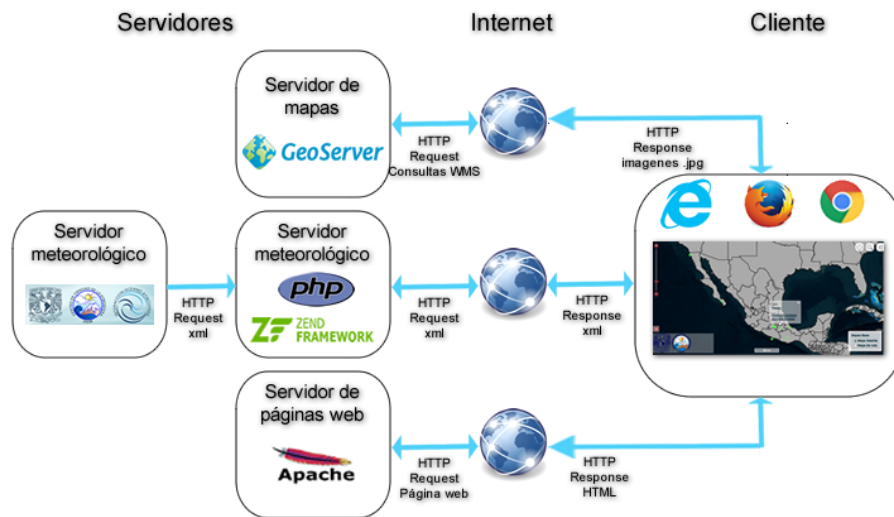


Figura 4.3: Diagrama de componentes

## 4.2 Implementación

La implementación del sistema web para este trabajo se dividió en dos partes:

- Implementación del servicio web basado en la arquitectura REST
- Implementación de la aplicación *web mapping*

### 4.2.1 Implementación del servicio web basado en la arquitectura REST

La implementación del servicio web contempla dos aspectos importantes. Tener un estándar adecuado para el formato de los datos obtenidos por el modelo WRF y la implementación de este estándar. Dado que se trata de un servicio web que maneja datos meteorológicos como temperatura, precipitación y viento en superficie, se dió a la tarea de investigar la existencia de posibles estándares para servicios web de pronósticos meteorológicos y se encontró con uno en específico: La especificación *Digital Weather Markup Language (DWML)*.

#### 4.2.1.1 Digital Weather Markup Language (DWML)

*Digital Weather Markup Language* (DWML o Lenguaje de Mercado Meteorológico Digital) es un nuevo lenguaje XML que se ha desarrollado para apoyar inicialmente el intercambio de datos del *National Weather Service* (NWS) y la *National Data Forecast* implementado por la *National Oceanic and Atmospheric Administration* (NOAA). Sin embargo, la especificación se ha escrito con la suficiente flexibilidad para abarcar otras aplicaciones de ciencias ambientales.

XML significa eXtensible Markup Language (XML) y es el estándar utilizado para el intercambio de datos en la *World Wide Web* (*www*), es soportado por software de dominio público. El DWML es el formato que utiliza nuevos productos XML para compartir datos meteorológicos.

DWML consta de tres partes principales:

- Una sección de encabezado que contiene una descripción del producto,
- una sección de definición y
- una sección de datos de pronóstico

Existen similitudes entre XML y HTML (*HyperText Markup Language*). Sin embargo, XML se utiliza para describir los datos, mientras que HTML se utiliza para describir el formato o para mostrar los datos.

#### 4.2.1.2 Componentes XML

DWML contiene tanto metadatos como datos meteorológicos. Ambos componentes están claramente separados dentro de un mensaje.

- **Metadatos DWML:** Los metadatos DWML proporcionan información sobre el mensaje DWML y los datos que contiene. Los metadatos proporcionados pueden ser opcionales para un emisor determinado.

– **Título del producto:** Un resumen conciso de lo que contiene el DWML.

- **Fecha de creación del producto:** La fecha y la hora de preparación del producto
- **Versión DWML:** Indica qué versión de DWML se está utilizando
- **Modo operativo:** El modo operativo le dice al usuario si el mensaje se produce como un producto de desarrollo, experimental, oficial o una prueba.
  - \* **De prueba:** Indica que es una instancia de un producto DWML existente que contiene algunos cambios y que están siendo evaluados. Comúnmente los usuarios no procesan este producto.
  - \* **De desarrollo:** Un nuevo producto que aun no está listo para la evaluación del público o para su uso.
  - \* **Experimental:** El producto está disponible para sus pruebas y evaluación por un período de tiempo específico con el fin de obtener una retro alimentación.
  - \* **Oficial:**
- **Clase de producto:** La clase de producto especifica el tipo general del producto contenido en una instancia de DWML. Los valores esperados incluyen datos meteorológicos, hidrológicos, oceanográficos, superficiales y espaciales.
- **Tipo de producto:** El tipo de producto define la categoría específica de la clase de producto a la que pertenece. Los ejemplos incluyen: observación, pronóstico y estadística.
- **Centro de producción:** Identifica qué organización crea el producto.
- **Sub - centro de producto:** El componente del centro que preparó el producto.
- **Disclaimer URL:** La dirección URL que contiene un deslinde de responsabilidades que el centro de producción desea que los usuarios conozcan.
- **URL de crédito:** La URL utilizada para acreditar el origen de los datos.
- **Logotipo de crédito:** La imagen opcional utilizada con la URL de crédito para confirmar la fuente de los datos.
- **Feedback URL:** La URL opcional de una página web utilizada para proporcionar comentarios sobre el producto al centro de producción.
- **Información adicional de URL:** Enlace a una previsión más completa o a la página web para la fuente del pronóstico.

- **Frecuencia de actualización sugerida:** Utilizado por el centro de producción para ayudar a los usuarios a saber con qué frecuencia devolver los datos actualizados. Dado que los centros productores se pueden actualizar cuando sea necesario, la frecuencia proporcionada no siempre garantizará que los usuarios se actualicen tan pronto como se disponga de nuevos datos. La frecuencia no garantiza que los datos recuperados sean los correspondientes a la hora de consulta. Sin embargo, la frecuencia de actualización sugerida ayudará a los usuarios a saber lo que el proveedor cree que es un tiempo razonable entre los accesos repetidos del sistema (es decir, segundos, días, meses, etc).
- **Datos DWML:** Los datos son un valor meteorológico con un componente geográfico y temporal. Para definir este valor, DWML necesita proporcionar la siguiente información:
  - **Valor de los datos:** Esta es la información recuperada o derivada de la bases de datos.
  - **Tipo de valor:** La categoría a la que pertenece el parámetro. Por ejemplo, la temperatura es la categoría que contiene la temperatura máxima, la temperatura del punto de rocío, etc.
  - **Nombre valor:** El nombre del parámetro que mide el valor (es decir, la temperatura máxima).
  - **Unidades de valor:** Las unidades del parámetro (por ejemplo, 71 F).
  - **Precisión de valor:** El valor debe proporcionarse con el número correcto de dígitos significativos.
  - **Tipo de resumen espacial:** Las colecciones de valores de punto de rejilla pueden resumirse en un solo valor. Si se hace esto, se proporcionará el tipo de resumen (media, mediana, modo, valor del centroide, etc.).
  - **Tipo de resumen temporal:** Se puede resumir un número de valores de punto de rejilla para un período de tiempo dado. Cuando se hace esto, se proporcionará el tipo de resumen temporal (período, hora, 3 horas, día, etc.).
- **Ubicación de los datos:** El lugar en tres dimensiones al que se aplica el valor de los datos.

- **Tipo de geografía:** Los tipos de geografía incluyen un punto (lat, lon), una ciudad y estado, condado, zona meteorológica del Servicio Nacional de Meteorología y código postal.
- **Ubicación horizontal:** La proyección de los datos proyectados sobre la superficie terrestre. longitud y latitud, una ciudad y estado específicos, etc.
- **Sistema de coordenadas verticales:** El tipo de coordenadas que definen la posición vertical de los datos (es decir, altura sobre el nivel del mar, sigma, etc.). Dependiendo del sistema de coordenadas verticales, también se necesitarán uno de los siguientes datos:
  - \* **Altura:** Esta es la distancia del punto por encima o por debajo de algún dato.
    - **Datum:** Esta es la referencia para la medición de altura (es decir, la superficie de la tierra, el nivel medio del mar, etc.).
    - **Unidades:** Las unidades de medida utilizadas para el valor de altura.
  - \* **Nivel:** La posición vertical de los datos digitales puede definirse en un nivel discreto.
  - \* **Capa:** La posición vertical de datos digitales puede definirse para una capa discreta.
- **Data time:** DWML proporcionará períodos de tiempo arbitrarios, así como referencias de tiempo comunes.
  - \* **Inicio del tiempo válido:** La hora de inicio del período para el cual los datos son válidos.
  - \* **Fin del tiempo válido:** La hora de finalización del período para el que los datos son válidos.
  - \* **Tiempo de coordinación:** Puede ser el tiempo local (LT) o tiempo Universal Coordinado (UTC).
  - \* **Nombre del período:** Una descripción del tiempo incorporado por los tiempos de inicio y fin. Por ejemplo, el tiempo podría corresponder a ESTA TARDE.

#### 4.2.1.3 Validación DWML

- DWML se ajustará a los requisitos que definen XML bien formados.



- DWML proporcionará un esquema para asegurar que cualquier instancia dada se ajusta a las especificaciones del lenguaje.
- DWML utilizará la codificación UTF - 8.

#### 4.2.1.4 Características generales DWML

- DWML utilizará nombres de atributo y elementos autodescriptivos.
- DWML empleará los estándares de la industria para asegurar que el idioma sea fácil de mantener y que pueda expandirse posteriormente.

#### 4.2.1.5 Ejemplos de productos con DWML

DWML deberá ser capaz de producir versiones XML de los siguientes tres productos web existentes.

- **Pronóstico de un vistazo:** El pronóstico actual de una cuadrícula experimental (ejemplo en línea) contiene la temperatura máxima y mínima, la condición del cielo y un icono meteorológico. La figura 4.4 muestra un producto del pronóstico meteorológico experimental.



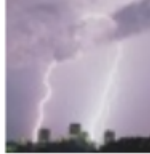




This Afternoon	Tonight	Saturday	Saturday Night	Sunday	Sunday Night	Monday
						
Partly Cloudy	Partly Cloudy	Mostly Cloudy	Mostly Clear	Partly Cloudy	Partly Cloudy	Partly Cloudy
Hi 74 °F	Lo 58 °F	Hi 79 °F	Lo 55 °F	Hi 81 °F	Lo 59 °F	Hi 77 °F

Figura 4.4: Ejemplo del pronóstico a un vistazo

- **Pronóstico tabular digital:** El pronóstico experimental actual contiene valores horarios de temperatura, temperatura del punto de rocío, probabilidad de precipitación, dirección y velocidad del viento, cobertura del cielo, así como información del tipo de precipitación. La figura muestra un ejemplo de este producto experimental.
- **Pronóstico de Zona Digital:** Este producto es similar al Pronóstico de un vistazo en que utiliza períodos de pronóstico de 12 horas para resumir los datos. El pronóstico de la zona digital es como el pronóstico tabular digital en términos de su uso de un gran número de elementos meteorológicos. La figura 4.5 muestra un ejemplo de este producto.

	<b>This Afternoon</b>	<b>Tonight</b>	<b>Saturday</b>	<b>Saturday Night</b>	<b>Sunday</b>	<b>Sunday Night</b>
Sky Cover	30 %	25 %	20 %	35 %	60 %	65 %
High/Low	74 °F	58 °F	79 °F	55 °F	81 °F	59 °F
Wind Direction	NW	NW	SW	SW	SW	SE
Wind Speed	5 -10	5 -10	5 -10	5 -10	10 - 15	5 -10
PoP	0	0	0	0	40	60
Weather Type	None	None	None	None	Thunderstorms	Showers

Figura 4.5: Ejemplo del pronóstico de zona digital

#### 4.2.1.6 Modelo de datos DWML

Las siguientes figuras muestran el modelo de datos DWML empleado para la generación del servicio Web.

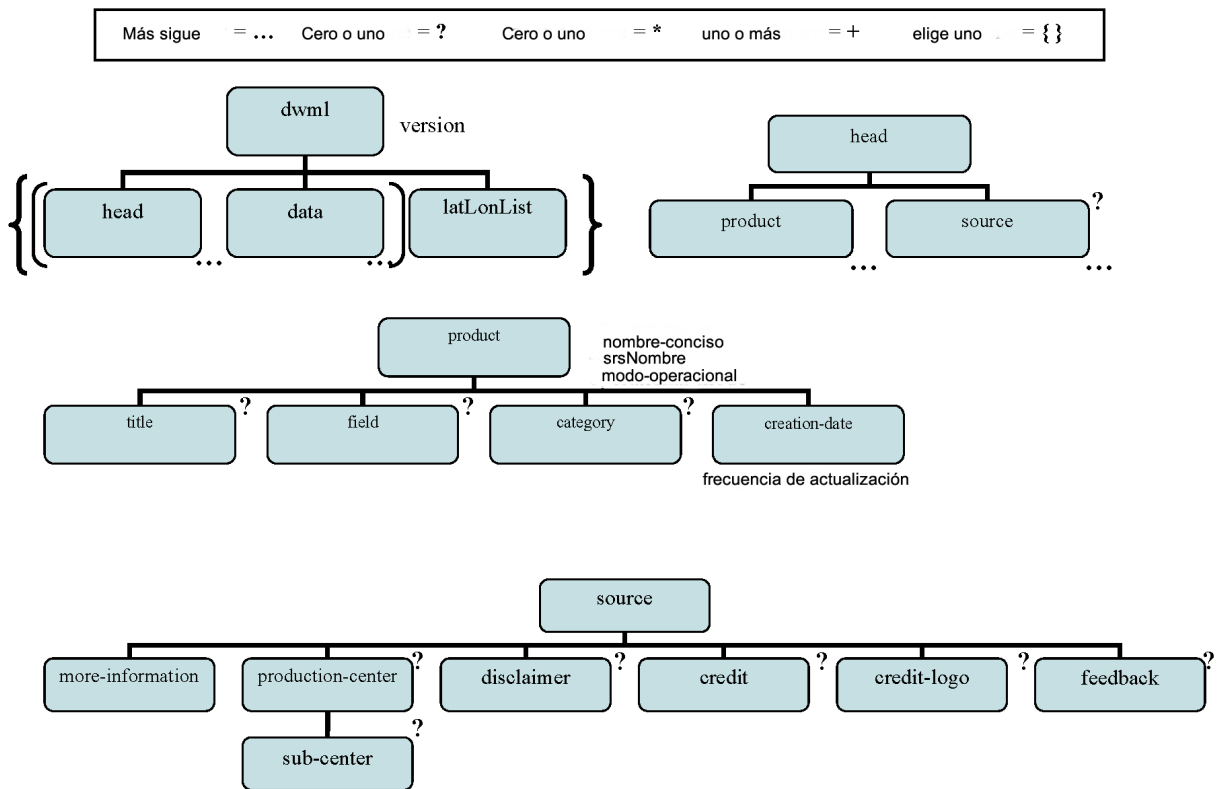


Figura 4.6: Cabecera DWML

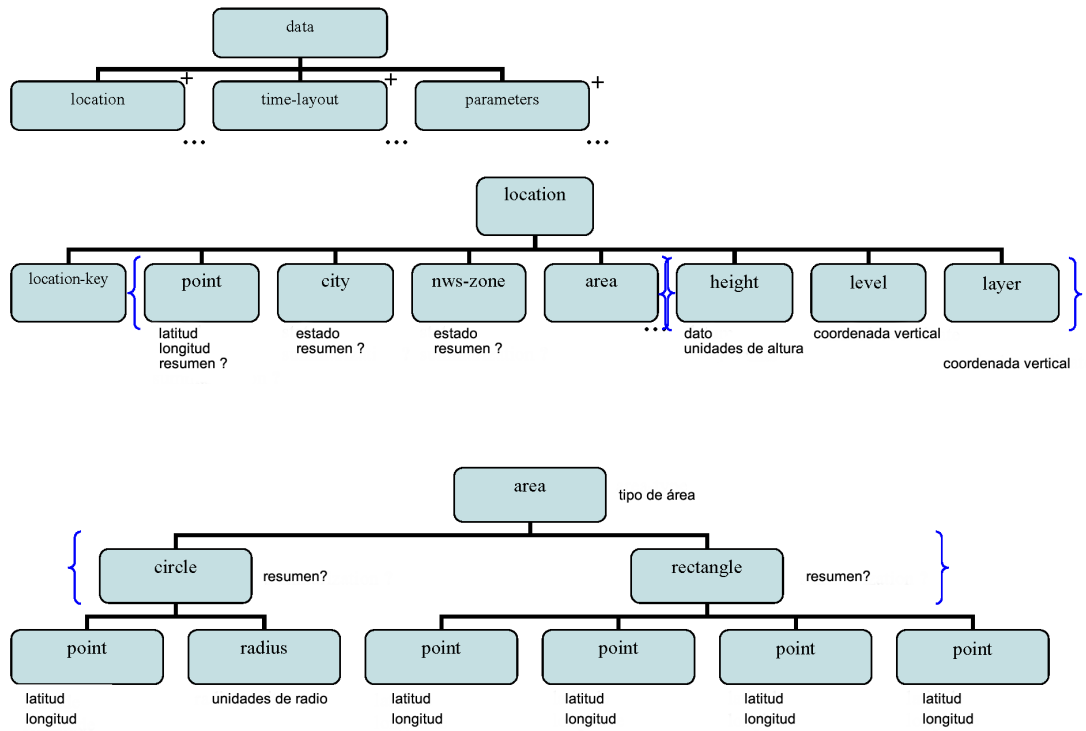


Figura 4.7: Datos DWML parte 1

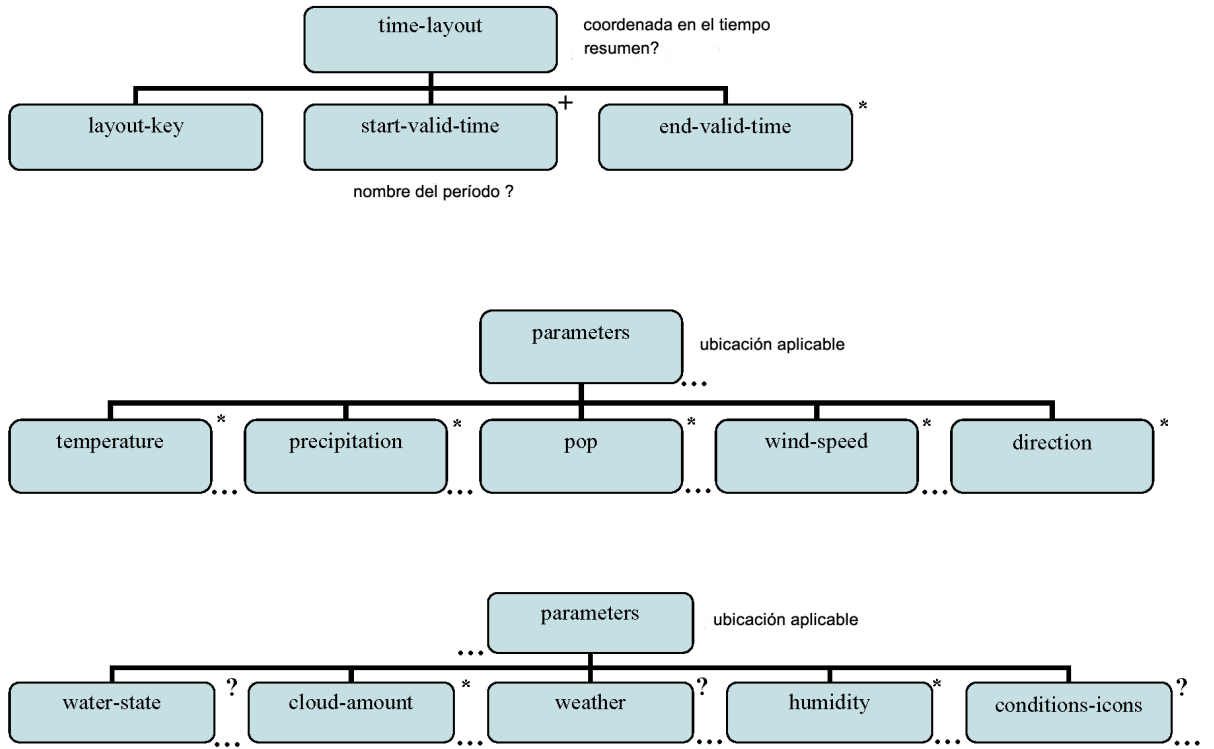


Figura 4.8: Datos DWML parte 2

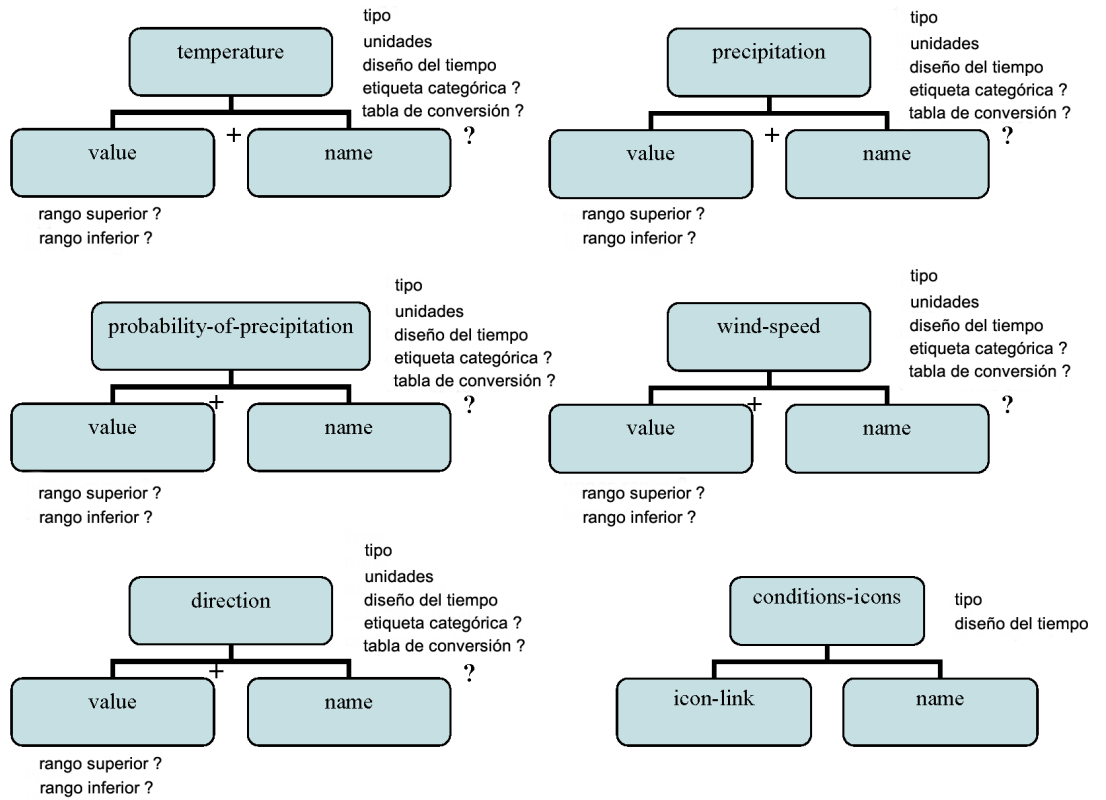


Figura 4.9: Variables DWML parte 1

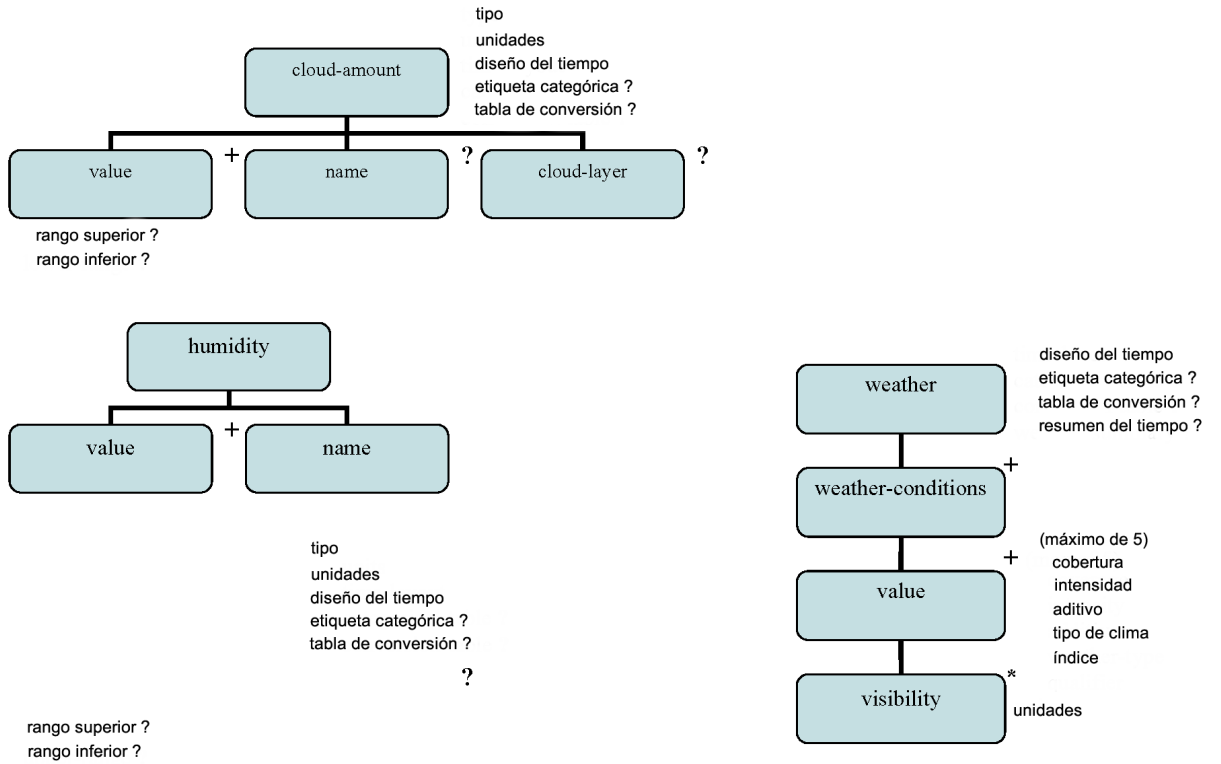


Figura 4.10: Variables DWML parte 2

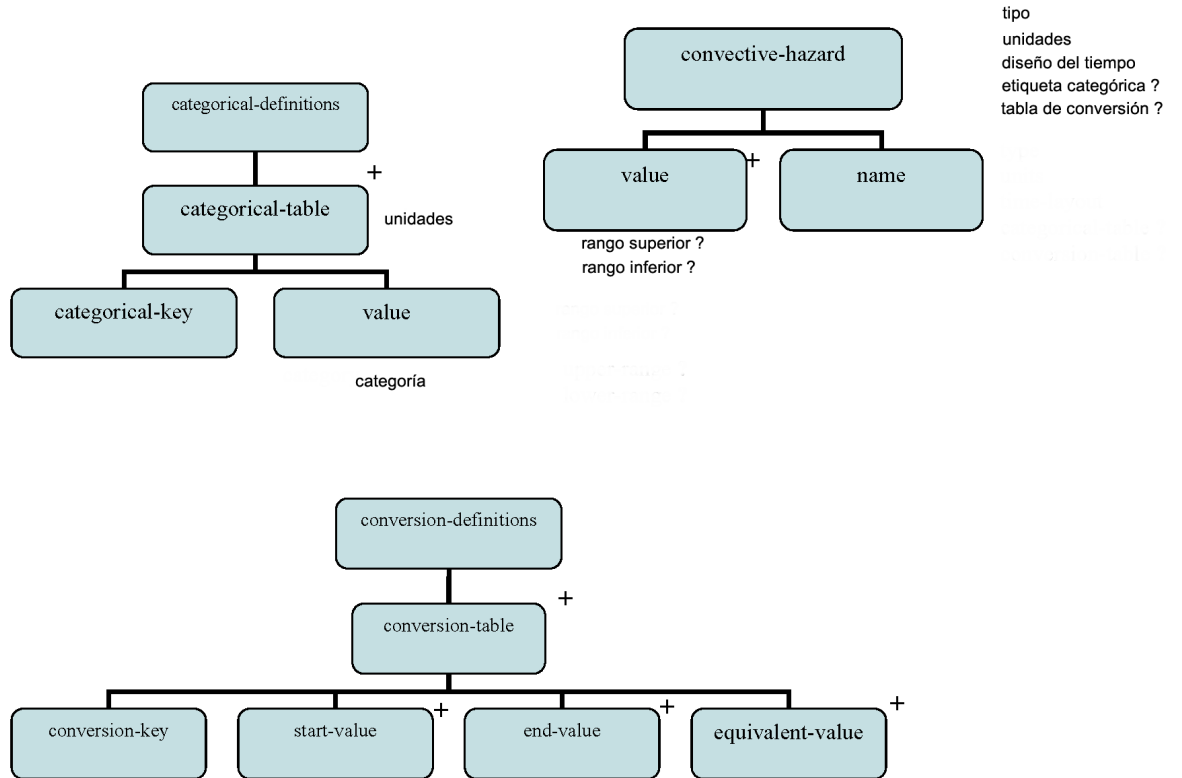


Figura 4.11: Variables DWML parte 3



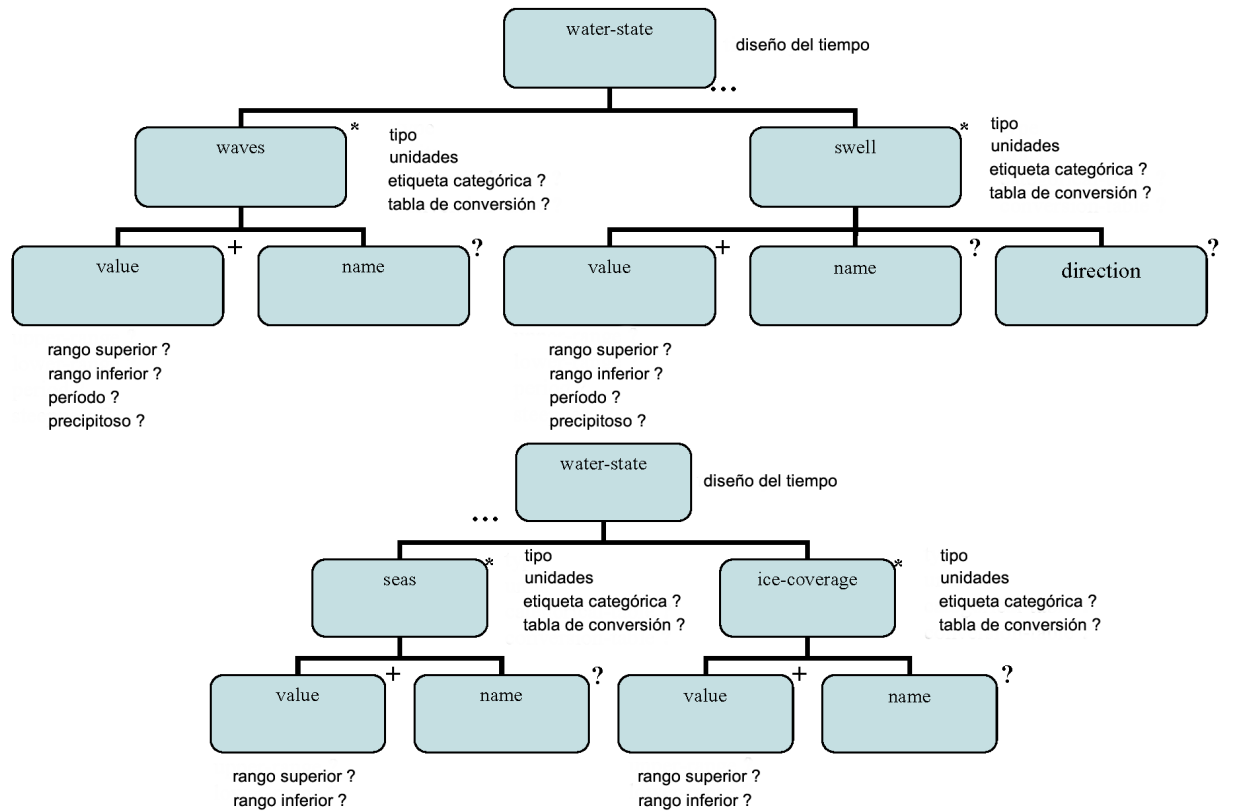


Figura 4.12: Variables DWML parte 4

#### 4.2.1.7 REST con PHP

Las rutinas básicas de PHP que se emplean para trabajar con servicios REST son:

- Creación de una solicitud
- Envío de la solicitud
- Recepción de la respuesta y tratamiento de la respuesta recibida

- Trabajar con verbos HTTP utilizando una biblioteca de cliente HTTP para enviar y recibir mensajes
- Utilizar la API del analizador de XML en PHP para generar y procesar solicitudes y respuestas XML
  - Crear solicitudes en el lado del cliente
  - Construir respuestas en el lado del servidor
  - Procesar las respuestas en el lado del cliente
  - Peticiones de proceso en el lado del servidor

Para ello utilizamos el framework de desarrollo *ZendFramework*(ZF) que como ya se mencionó en el capítulo 3: criterios para la selección del software. ZF es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos.

Cabe mencionar que ZF usa el Modelo Vista Controlador para la creación de las aplicaciones Web. A continuación se hablará un poco de este modelo.

### 4.2.1.8 Modelo-Vista-Controlador

El Modelo Vista Controlador o MVC (*Model-View-Controller*) es un patrón de arquitectura de software que divide una aplicación en tres partes:

- **Modelo:** Las vistas definen exactamente qué es lo que se presenta al usuario. Usualmente los controladores suelen pasar datos a cada vista para representarlos en algún formato. En las vistas generalmente trabajamos con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas requerirán los datos a los modelos y ellas generarán la salida tal como la aplicación requiera.
- **Vista:** Las vistas definen exactamente qué es lo que se presenta al usuario. Usualmente los controladores suelen pasar datos a cada vista para representarlos en algún formato. En las vistas generalmente trabajamos con los datos, sin embargo, no se realiza un acceso

directo a éstos. Las vistas requerirán los datos a los modelos y ellas generarán la salida tal como la aplicación requiera.

- **Controlador:** Los controladores enlazan el patrón completo. Éstos manipulan modelos, deciden que vista mostrar basados en las peticiones del usuario y otros factores, transmiten los datos que necesitará cada vista. Sin embargo, su responsabilidad no es manipular directamente los datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

Así, para instalar ZF basta con descargar el software desde su página oficial, desempaquetarlo y dejarlo en una ubicación adecuada (comunmente en: /usr/local/) e indicarle a PHP la ruta donde se encuentra para su utilización.

La figura 4.13 muestra la organización de las carpetas del ZF.

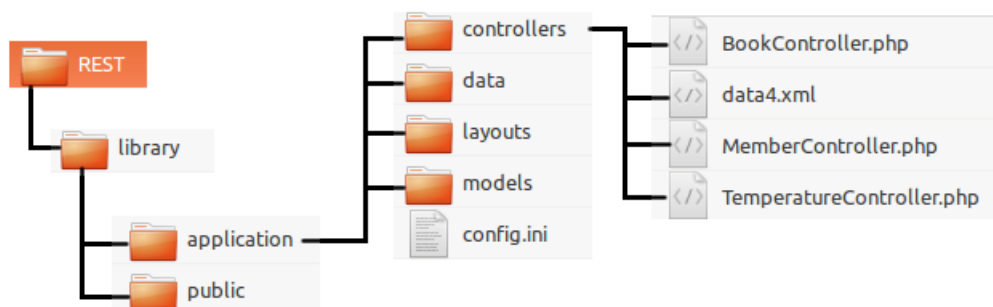


Figura 4.13: Estructura de las carpetas en ZF

Se puede observar que es en la carpeta *application* donde se encuentran las diferentes subcarpetas que se organizan de acuerdo al modelo MVC. De ahí que la carpeta *controllers* guarde los controladores del sistema. La carpeta *data* guarda las bases de datos que se pueden llegar a ocupar en el sistema, la carpeta *layouts* guarda las vistas del sistema y la carpeta *models* guarda los archivos relacionados con la lógica de negocios del sistema.

Para el desarrollo del servicio web que necesita el sistema en este trabajo de tesis se hace uso solamente de la carpeta *controllers* para definir el controlador en php que es el encargado de formar el servicio web necesitado. También se crea el template que contiene la cabecera con

los metadatos DWML de acuerdo a la especificación DWML descrita en el paratado anterior. La figura 4.14 muestra el template creado de acuerdo a la especificación.

```

<?xml version="1.0"?>
<dwml version="1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.nws.noaa.gov/forecasts/xml/DWMLgen/schema/DWML.xsd">
  <head>
    <product srsName="WGS 1984" concise-name="time-series" operational-mode="official">
      <title>IOA pronostico 5 dias</title>
      <field>meteorological</field>
      <category>forecast</category>
    </product>
    <source>
      <more-information>http://grupo-ia.atmosfera.unam.mx</more-information>
      <production-center>Centro de ciencias de la atmosfera UNAM <sub-center> Grupo Interaccion Oceano-Atmosfera
</sub-center></production-center>
      <disclaimer>http://grupo-ia.atmosfera.unam.mx</disclaimer>
      <credit>http://grupo-ia.atmosfera.unam.mx</credit>
      <credit-logo>http://grupo-ia.atmosfera.unam.mx/imagenes/ia-logo.jpg</credit-logo>
    </source>
  </head>
  <data>
    <location>
      <location-key>state1</location-key>
      </location>
      <moreWeatherInformation applicable-location="state1">http://grupo-ia.atmosfera.unam.mx/informacion-servicios
</moreWeatherInformation>
      </moreWeatherInformation>
      <time_layout time-coordinate="UTC" summarization="none">
        <layout-key>k-p1h-n120-1</layout-key>
      </time_layout>
      <parameters applicable-location="state1">
        <temperature type="hourly" units="Celsius" time-layout="k-p1h-n120-1">
          <name>Temperaturas </name>
        </temperature>
        <precipitation type="liquid" units="hundredths of centimeters" time-layout="k-p1h-n120-1">
          <name>Precipitacion acumulada</name>
        </precipitation>
        <wind_speed type="sustained" units="meters/second" time-layout="k-p1h-n120-1">
          <name>Velocidad de viento</name>
        </wind_speed>
      </parameters>
    </data>
  </dwml>

```

Figura 4.14: Template creado de acuerdo a la especificación DWML

Una vez que se tiene ya creado el template, el cual va a contener datos estáticos, se procede a la creación del controlador TemperatureController.php

El programa utiliza el archivo Server.php para poder heredar la clase ZendControllerAction

La función getTemperatures() carga el template creado anteriormente, abre el archivo de texto con los datos del modelo WRF y los empieza a leer para ir formando el resto del documento XML de acuerdo a la especificación DWML referida anteriormente.

El resultado de estas rutinas es la creación de un servicio web como el que muestra la

```

class TemperatureController extends Zend_Controller_Action
{
    function indexAction()
    {
        // $books = new Books();
        $this->_helper->viewRenderer->setNoRender();

        if ($this->_request->isGet())
        {
            $server = new Zend_Rest_Server();
            $server->addFunction('getTemperatures');
            $params['method'] = 'getTemperatures';
            $server->handle($params);
        }
    }
}

```

Figura 4.15: Clase ZendControllerAction

```

$xml = simplexml_load_file("/home/uniatmos/Sites/REST/library/application/data/template.xml");
// Obetner Fecha actual (falta)
$fecha_act = $xml->head->product->addChild('creation-date', '2016-07-21T15:30:03Z');
$fecha_act->addAttribute('refresh-frequency', 'PT1H' );

$location_act = $xml->data->location->addChild('nws-zone', '');
//obtener estado por parametros
$location_act->addAttribute('state', 'MX' );

//parametrizar
$fp = fopen('/home/uniatmos/Sites/REST/library/application/data/df.txt', 'r');
if (!$fp){
    echo 'ERROR: No ha sido posible abrir el archivo. Revise su nombre y sus permisos.';
    exit;
}
//leer archivos de datos
$row = 0;
$field = array();
while (!feof($fp)){
    $line = fgets($fp, 2048);
    array_push($field, explode(',', $line));
    $row++;
}
fclose($fp);

for($i=0; $i<count($field)-1; $i++)
{
    $dateS = $field[$i][0]. "T". $field[$i][1]. ":00:00Z";
    $xml->data->time_layout->addChild('start-valid-time', $dateS);

    $xml->data->parameters->temperature->addChild('value', $field[$i][2]);
    $xml->data->parameters->wind_speed->addChild('value', $field[$i][3]);
    $xml->data->parameters->precipitation->addChild('value', $field[$i][4]);
}

return $xml;
}

```

Figura 4.16: Función getTemperatures

figura

```
<dwml xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-
instance" version="1.0"xsi:noNamespaceSchemaLocation="http://www.nws.no
aa.gov/forecasts/xml/DWMLgen/schema/DWML.xsd">
```

```
<head>...</head>
<data>...</data>
```

```
</dwml>
<head>
  <product srsName="WGS 1984" concise-name="time-series" operational-
mode="official">
    <title>IOA pronostico 5 días</title>
    <field>meteorological</field>
    <category>forecast</category>
    <creation-date refresh-frequency="PT1H">2016-07-21T00:00Z</creation-
date>
  </product>
  <source>
    <more-information>http://grupo-ioa.atmosfera.unam.mx</more-information>
    <production-center>
      Centro de ciencias de la atmosfera UNAM<sub-center>Grupo Interaccion
      Oceano-Atmosfera</sub-center>
    </production-center>
    <disclaimer>http://grupo-ioa.atmosfera.unam.mx</disclaimer>
    <credit>http://grupo-ioa.atmosfera.unam.mx</credit>
    <credit-logo>http://grupo-ioa.atmosfera.unam.mx/images/ioa.jpg</credit-logo>
  </source>
</head>
```

Figura 4.17: Cabecera del servicio web

```
<data>
  <location>...</location>

  <moreWeatherInformation applicable-
location="state1">...</moreWeatherInformation>

  <time_layout time-coordinate="UTC" summarization="none">
    <layout-key>k-p1h-n120-1</layout-key>
    <start-valid-time>2016-08-01T00:00:00Z</start-valid-time>
    <start-valid-time>2016-08-01T01:00:00Z</start-valid-time>
    <start-valid-time>2016-08-01T02:00:00Z</start-valid-time>
    <start-valid-time>2016-08-01T03:00:00Z</start-valid-time>
    <start-valid-time>2016-08-01T04:00:00Z</start-valid-time>
    <start-valid-time>2016-08-01T05:00:00Z</start-valid-time>
    <start-valid-time>2016-08-01T06:00:00Z</start-valid-time>
    <start-valid-time>2016-08-01T07:00:00Z</start-valid-time>
    <start-valid-time>2016-08-01T08:00:00Z</start-valid-time>
    <start-valid-time>2016-08-01T09:00:00Z</start-valid-time>
```

Figura 4.18: Cuerpo del servicio web

## 4.2.2 Implementación de la aplicación web de mapas (*web mapping application*)

Lo primero que se implementó para la realización de la aplicación web mapping fue el servidor de mapas (GeoServer), para de esta forma elaborar los archivos vectoriales necesarios en la aplicación. Como se mencionó en el apartado anterior, uno de los archivos vectoriales representa la división política de la República mexicana y el otro archivo representa los puntos o entidades federativas de la República a los cuales se pueden consultar su pronóstico meteorológico. A continuación se describe la implementación del servidor web de mapas (GeoServer).

### 4.2.2.1 Implementación del servidor de mapas (GeoServer)

#### 4.2.2.2 Instalación

Existen varias formas de instalar GeoServer en diferentes plataformas (Windows, Mac OS X y Linux) para el desarrollo de la tesis fue necesario instalar GeoServer en su versión 2.10.2 estable bajo un sistema operativo Linux; esto llevó diferentes pasos:

- Se instaló el jdk de SUN Oracle en su versión 8
- Se descargó e instaló el contenedor de páginas jsp Tomcat en su versión 8.5.3:
  - se asignaron los puertos a usar por Tomcat (archivo: server.xml)
  - se creó el usuario admin para el manejo de la interfaz web de Tomcat (archivo: tomcat - users.xml)
  - se establecieron las variables JAVA HOME y JAVA JRE (archivo: catalina.sh)
  - se creó el bash runThisInstance para establecer las variables de entorno CATALINA HOME, CATALINA BASE y arrancar tomcat

Con el contenedor de páginas jsp Tomcat funcionando, se procedió a la descarga de GeoServer en su versión 2.10.2 y en formato war para que pudiera ser manejado por Tomcat. Esto se logró accediendo al manager de Tomcat a través de un navegador web (Mozilla Firefox) utilizando el usuario admin y su contraseña previamente creada.

<http://132.248.8.125:8085/manager/html>

Una vez dentro del manager de Tomcat se subió el archivo geoserver.war previamente descargado y se realizó su despliegue.

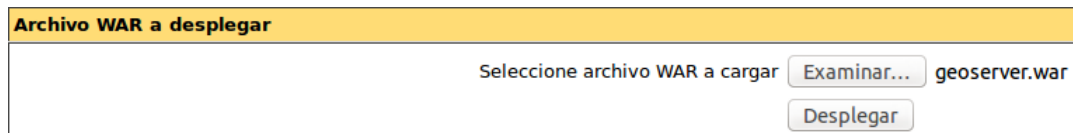


Figura 4.19: Archivo war - GeoServer

Como se mencionó en el capítulo 3 sección Análisis, subsección Criterios generales para la selección del software, GeoServer es una aplicación web de código abierto, escrito en Java, que permite a los usuarios compartir y editar datos geoespaciales. Diseñado para la interoperabilidad, GeoServer publica los datos de cualquier fuente de datos espaciales utilizando estándares abiertos.

#### 4.2.2.3 Solicitud HTTP GET WMS con GeoServer

Las operaciones WMS se invocan utilizando un navegador web estándar (p. ej. Firefox) realizando peticiones en la forma de URL. El contenido de la URL depende de la operación solicitada. El siguiente ejemplo es una petición WMS representada como una petición HTTP GET a una instancia alojada en GeoServer que representa la división política de la República mexicana y sus entidades federativas.

```

1 http://132.248.8.125:8085/geoserver/prueba/wms?
2 SERVICE=WMS&
3 VERSION=1.3.0&
4 REQUEST=GetMap&
5 LAYERS=prueba:division_repmex,entidades_fed&
6 SRS=EPSG:4326&
7 BBOX=-7.7,40.1,-1.7,43.2&
8 FORMAT=image/png&
9 WIDTH=600&HEIGHT=255

```

Figura 4.20: Ejemplo de solicitud HTTP GET WMS con GeoServer

La operación GetMap solicita que el servidor genere un mapa. Los parámetros básicos (sin entrar en detalles de parámetros obligatorios y opcionales) especifican:

- Nombre de servidor y número de puerto



- Nombre del servicio
- Versión
- Tipo de petición
- Capa o capas para representar en el mapa
- Sistema de referencia espacial de destino
- Cuadro delimitador para la extensión del mapa
- Formato para la salida
- Ancho y alto

La información necesaria para especificar los valores de parámetros tales como capas, estilos y srs se pueden obtener a partir del documento de Capacidades (operación GetCapabilities).

### 4.2.2.4 Implementación de los servicios WMS / WFS a partir de shapefile

GeoServer incluye una interfaz de administración web a través de la cual se accede a la mayor parte de las configuraciones de datos y servicios. Esta interfaz reduce la necesidad de editar los archivos de configuración a mano.

Para acceder a GeoServer en primer lugar debemos iniciar el servicio.

La url de la interfaz de administración de GeoServer para la tesis es:

<http://132.248.8.125:8085/geoserver>

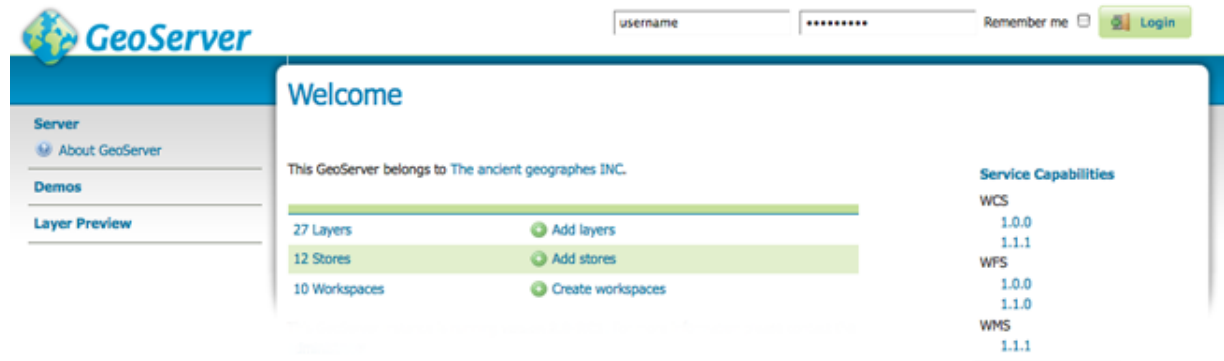


Figura 4.21: Interfaz de administración de GeoServer

Para crear un servicio WMS con GeoServer es necesario identificarse.

#### 4.2.2.5 Publicación de un shapefile con el importador de capas de GeoServer

Para poder servir las capas y crear el servicio WMS (o WFS), el paso esencial fue el de importar los shapefiles a GeoServer. Desde la página de administración de GeoServer, se hizo clic en el enlace Import Data en el lado izquierdo, debajo de Datos.

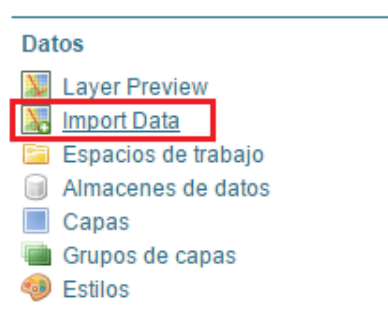


Figura 4.22: Importador de datos de GeoServer

En la página de importación de datos

Se marcó la opción Spatial Files para seleccionar los archivos shapefile (.shp) como tipo de datos, se buscó y seleccionó los archivos que se necesitaron importar o la carpeta en la que se ubicaron los archivos shapefiles. Muy útil para importaciones masivas, Se seleccionó el workspace y el almacén

Se denomina espacio de trabajo o workspace al contenedor usado para agrupar capas similares. Todas las capas y almacenes de datos deben tener un espacio de trabajo asociado (con excepción de los grupos de capas).





El almacén de datos o store es el contenedor de datos geográficos. Un almacén de datos se conecta a una fuente de datos que contiene datos raster o vectoriales. Un origen de datos puede ser un archivo (como un shapefile) o grupo de archivos, una tabla de una base de datos o un directorio.

Cada almacén de datos está asociado a un workspace (sólo a uno).

La figura siguiente muestra un ejemplo del proceso anterior.

## Import Data

### 1. Choose a data source to import from

-  Spatial Files - Files from a directory or archive
-  Mosaic - Raster files from a directory composing a mosaic
-  PostGIS - Tables from PostGIS database
-  Oracle - Tables from Oracle database (Install plugin and drivers.)
-  SQL Server - Tables from Microsoft SQL Server database (Install plugin and drivers.)

### 2. Configure the data source

Choose a file or directory

[Browse...](#)

### 3. Specify the target for the import

Workspace

Store

### Recent Imports

<input type="checkbox"/>	Import	Status	Last Updated
<input type="button" value="&lt;&lt;"/> <input type="button" value="&lt;"/> <input type="button" value="&gt;"/> <input type="button" value="&gt;&gt;"/> Resultados 0 a 0 (de un total de 0 ítems)			

Figura 4.23: Proceso para importar datos con GeoServer

Se hizo clic en Next y en la siguiente página se marcó las casillas de las capas que se necesitaban importar

Status	Created	Last Updated
PENDING	hace instantes	hace instantes

home/angel/datos/division\_repmex.shp Select: All | None | Ready

Layer	Status	Actions
<input checked="" type="checkbox"/> division_repmex	➔ Ready	Advanced...

<< < | I | > >> Resultados 1 a 1 (de un total de 1 ítems)

Figura 4.24: Capas importadas con GeoServer

Una vez completado el proceso, se pueden visualizar las capas. Una capa vectorial almacenada en GeoServer, ya tiene habilitados por defecto los servicios WMS, WFS y el acceso a WPS. La visualización de las capas en este punto no es más que una petición GetMap, en la que se especifica como parámetro que la aplicación que se utiliza para mostrar el WMS va a ser OpenLayers:

Status	Created	Last Updated
COMPLETE	hace instantes	hace instantes

home/angel/datos/division\_repmex.shp Select: All | None | Ready

Layer	Status	Actions
<input checked="" type="checkbox"/> division_repmex	✔ Import successful.	View in Layer Preview ▾ Go

<< < | I | > >> Resultados 1 a 1 (de un total de 1 ítems)

Figura 4.25: Despliegue de una capa con GeoServer

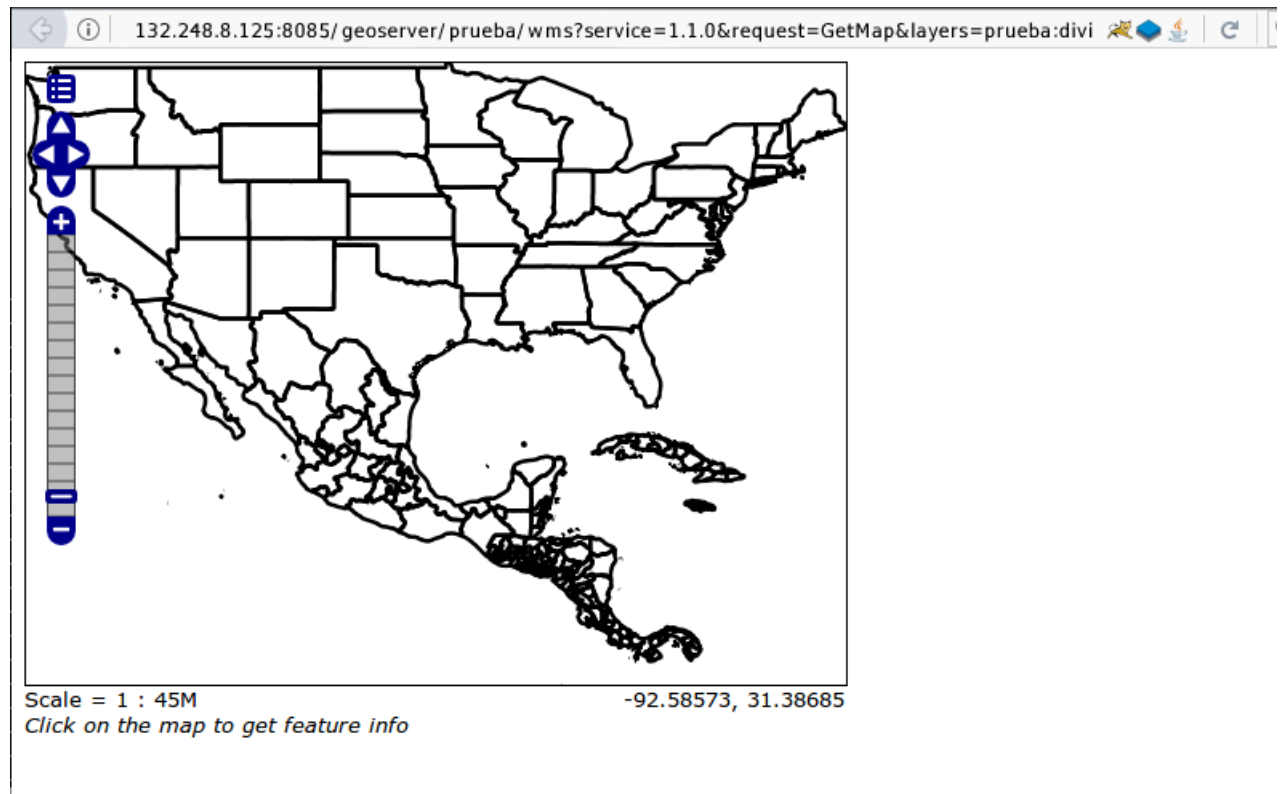


Figura 4.26: Visualización de una capa con GeoServer usando OPenLayers

Con esto, se tiene un servicio WMS habilitado en GeoServer listo para que la aplicación web mapping haga uso de él.

#### 4.2.2.6 La interfaz gráfica de usuario principal

La aplicación *web mapping* se inicia mediante una *url* a su sitio web. Cuando la aplicación termina de cargarse, se presenta al usuario la interfaz gráfica principal (véase la figura: 4.27). Esta interfaz de visualización se compone principalmente de un mapa, con una barra de búsqueda en la parte superior izquierda, un sistema de navegación en la parte superior derecha y un panel para intercambiar las capas base en la parte inferior derecha, así como los respectivos logos institucionales.

Debido al propósito y naturaleza de la aplicación, el mapa tiene que ser lo más grande posible, pero todavía deja espacio para los controles mencionados anteriormente con controles html fáciles de utilizar y con iconos que dan una clara indicación de su funcionalidad. Además

del mapa base que se presenta, por encima de él existen dos imágenes vectoriales que dibujan la división política de la República Mexicana y los puntos o marcadores que representan las entidades federativas del país.

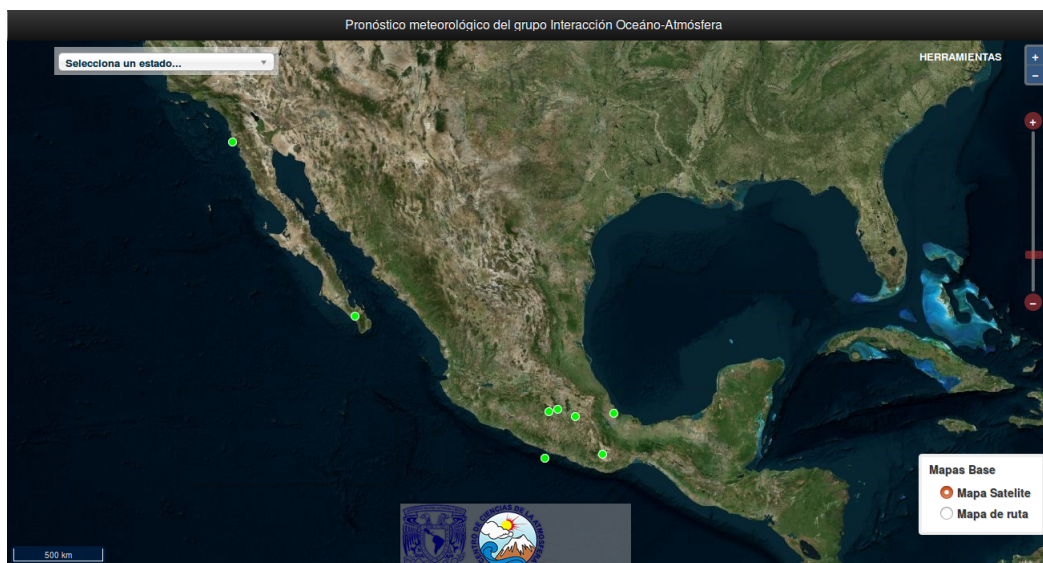


Figura 4.27: Interfaz gráfica de usuario

Se puede ir a los marcadores (color verde) que representan las entidades federativas del país mediante la barra de selección de estado o directamente haciendo clic en el marcador deseado. Sea cual sea el método elegido, se accede a la información meteorológica asociada a las entidades federativas, desplegando un *popup* con la temperatura para ese estado, así como a un vínculo que permite acceder al pronóstico detallado para la entidad federativa seleccionada. Las figuras 4.28 y 4.29 muestran este proceso.

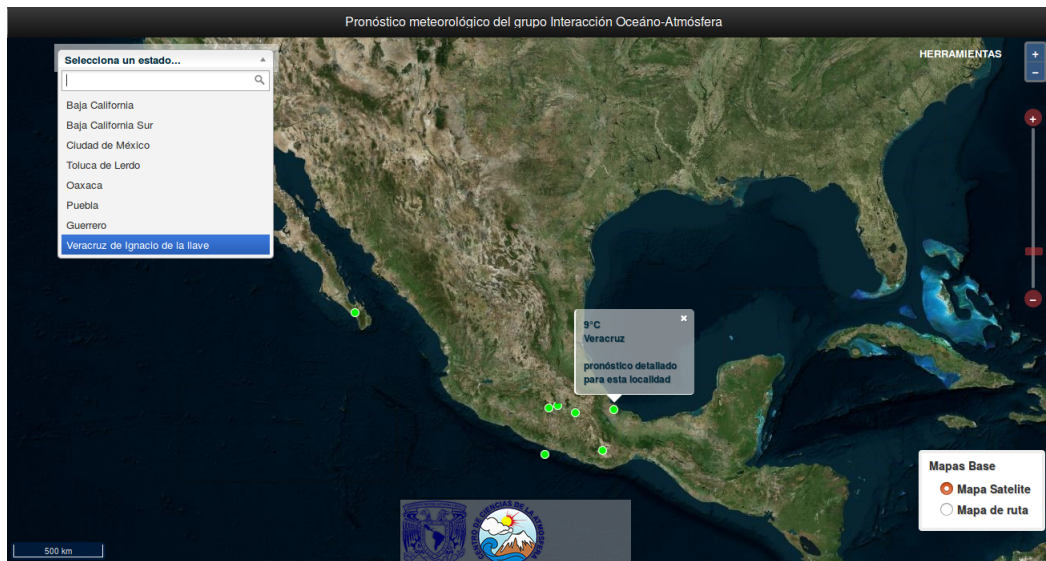


Figura 4.28: Selección de una entidad federativa

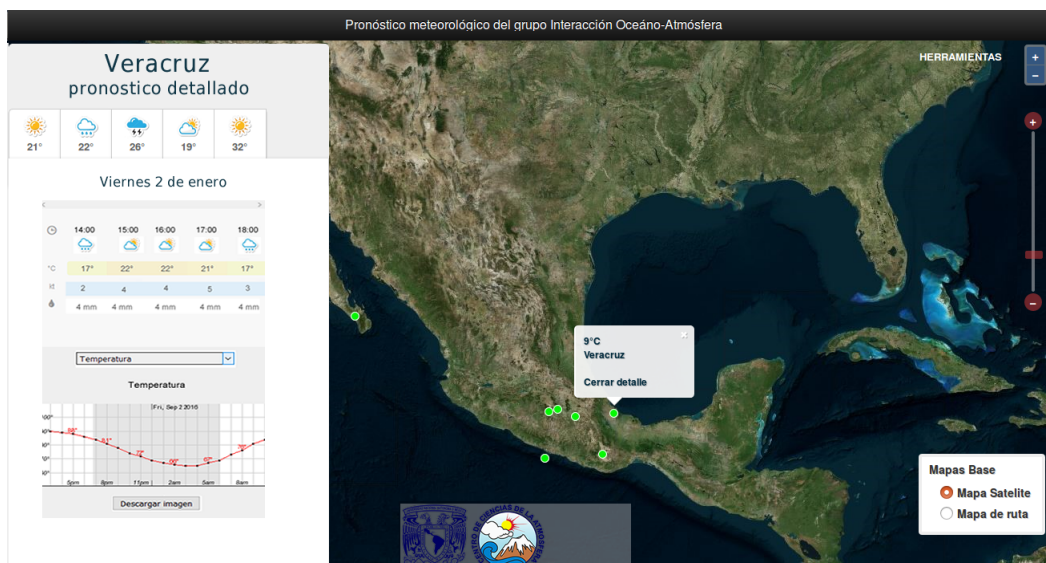


Figura 4.29: Despliegue del pronóstico detallado

#### 4.2.2.7 Compatibilidad de hardware

La aplicación está diseñada para el uso de tabletas. Se probó en el iPad de Apple iOS; pero deberá funcionar de la misma manera en otras tablets y sistemas operativos. Se ha ejecutado en un Samsung Galaxy Tab 10.1, pero para este caso, sólo debe ejecutarse en modo *landscape*. La aplicación también se ejecuta en computadoras de escritorio en los navegadores web *Firefox*,



*Chrome e Internet Explorer*. Si la aplicación se ejecuta en un smartphone, se proporciona una versión más sencilla de la misma, en la cual se disponen de controles más básicos. Esto se maneja automáticamente.

#### 4.2.2.8 El mapa

El mapa muestra dos capas vectoriales; una que representa la división política del país y la otra sus entidades federativas. Admite gestos de navegación táctil. Esto significa que, en los dispositivos móviles se puede tocar, arrastrar, hacer acercamientos o alejamientos con los dedos, similar a las funciones que se hacen con el ratón.

#### 4.2.2.9 El menú de estados

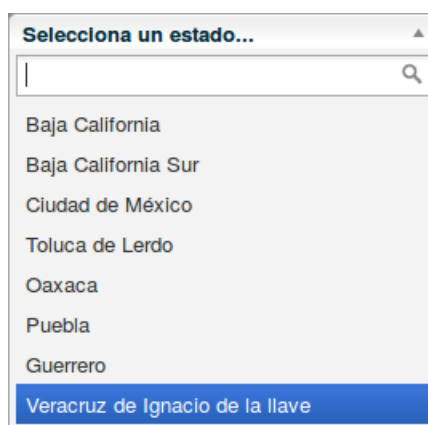


Figura 4.30: Menú de estados

El menú de estados es un control web que permite a un usuario seleccionar de manera directa los estados o entidades federativas colocadas en el mapa, o buscar alguna de las proporcionadas mediante un buscador integrado.

#### 4.2.2.10 El panel del pronóstico detallado

Este panel se activa al hacer clic en el vínculo que proporciona una entidad federativa seleccionada y en él se muestra la información detallada del pronóstico meteorológico para esa entidad. Información obtenida por el modelo WRF y entregada a la aplicación *webmapping* mediante un servicio web.

El panel esta dividido en el pronóstico a cinco días para las variables temperatura, precipitación y viento en superficie y un control web que permite al usuario graficar las variables pronosticadas.



Figura 4.31: Pronóstico detallado

#### 4.2.2.11 El selector de capas

Como su nombre lo indica, este control web permite al usuario cambiar las capas base de la aplicación. Por el momento, las capas base son proporcionadas por BingMaps

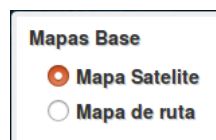


Figura 4.32: Selector de capas

## Resultados

### 5.1 Conclusiones

Con base en los resultados presentados se puede decir que se han cumplido con los objetivos del proyecto. Se ha creado una aplicación web con tecnologías open source que permite difundir los productos meteorológicos generados por el grupo.

Entre las principales contribuciones del presente trabajo es el uso de estándares para el intercambio de información web para la implementación de servicios REST . Los servicios creados podrán ser consumidos no sólo por la aplicación web mapping desarrollada en la tesis sino por cualquier desarrollador de plataformas web meteorológicas.

### 5.2 Trabajo futuro

- Cambiar los archivos por base de datos.
- Crear un servicio web que obtenga información meteorológicas de otras variables del WRF.
- Aumentar la capacidad del servicio web para que regrese datos regionales especificados por un polígono en vez de realizar consultas por datos puntuales.





## Conceptos básicos de Cartografía

La cartografía es la ciencia que se encarga de reunir y analizar medidas y datos de regiones de la Tierra, para representarlas gráficamente a diferentes dimensiones lineales. Al ser la tierra esférica, o más bien geoide, lo cual es una derivación del término “esférico” , ha de valerse de un sistema de proyecciones para pasar de la esfera al plano. El problema es aún mayor, pues en realidad la forma de la Tierra no es exactamente esférica, su forma es más achatada en los polos, que en la zona ecuatorial. A esta figura se le denomina Elipsoide. Pero además de representar los contornos de las cosas, las superficies y los ángulos, se ocupa también de representar la información que aparece sobre el mapa, según se considere qué es relevante y qué no. Esto, normalmente, depende de lo que se quiera representar en el mapa y de la escala.

### A.1 Sistemas de Coordenadas

El sistema de coordenadas geográficas es un sistema que referencia cualquier punto de la superficie terrestre y que utiliza para ello dos coordenadas angulares, latitud (norte o sur) y longitud (este u oeste), para determinar los ángulos laterales de la superficie terrestre con respecto al centro de la Tierra y alineadas con su eje de rotación.

### A.1.1 Latitud y longitud

Cualquier lugar de la superficie terrestre puede ser referenciado por la intersección de un paralelo y un meridiano, es a partir de estos conceptos de donde surgen las coordenadas de latitud y longitud.

La latitud proporciona la localización de un lugar, en dirección Norte o Sur desde el ecuador y se expresa en medidas angulares que varían desde los  $0^\circ$  del Ecuador hasta los  $90^\circ\text{N}$  ( $+90^\circ$ ) del polo Norte o los  $90^\circ\text{S}$  ( $-90^\circ$ ) del polo Sur. Como podemos ver en la imagen, si trazamos una recta que vaya desde el punto P hasta el centro de la esfera O, el ángulo que forma esa recta con el plano ecuatorial expresa la latitud de dicho punto.

La longitud proporciona la localización de un lugar, en dirección Este u Oeste desde el meridiano de referencia  $0^\circ$ , o meridiano de Greenwich, expresándose en medidas angulares comprendidas desde los  $0^\circ$  hasta  $180^\circ\text{E}$  ( $+180^\circ$ ) y  $180^\circ\text{W}$  ( $-180^\circ$ ). Se puede ver en la imagen que el ángulo b mide la distancia angular del meridiano del lugar P con el meridiano  $0^\circ$  (meridiano de Greenwich). Es lo mismo medir este ángulo sobre el círculo del ecuador que sobre el círculo del paralelo que pasa por el punto P, el valor angular de b es igual en ambos casos. En el ejemplo de la figura A.1, la longitud es Oeste (W) puesto que el meridiano del punto P está al Oeste del meridiano de Greenwich. Mientras que un grado de latitud corresponde a una distancia casi idéntica como se ha comentado anteriormente, no sucede lo mismo con un grado de longitud dado que los círculos sobre los cuales se miden convergen hacia los polos. En el ecuador, un grado de longitud equivale a 111,32 kms que es el resultado de dividir la circunferencia ecuatorial entre  $360^\circ$ .

### A.1.2 Proyecciones

Cómo ya se dijo, ningún mapa en la Tierra son representaciones verdaderamente perfectas; todos los mapas tienen algo de distorsión. La razón de esto, es debido a que se está tratando de representar un objeto 3D (un elipsoide: la Tierra) en dos dimensiones (un plano: el propio mapa).

Una proyección es una representación de la totalidad o partes de una superficie de una esfera 3D (o más precisamente, un elipsoide) en un plano 2D (u otros tipos de geometría). Cada

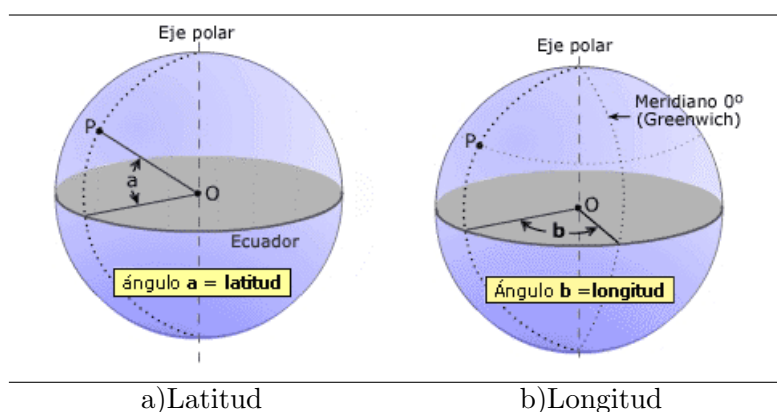


Figura A.1: Coordenadas geográficas

mapa tiene algún tipo de proyección. Esto es un atributo inherente de los mapas. Imaginemos que pelamos una naranja y luego intentamos aplanar la cáscara. Es en ese momento cuando algún tipo de distorsión sucederá, y si tratáramos de encajar completamente la cáscara en un cuadrado o rectángulo (como un mapa plano, de dos dimensiones) se tendrían muchos problemas para encajar. Para conseguir que la cáscara pudiese encajar perfectamente en un plano cuadrado o en un rectángulo, se puede tratar de estirar o cortar algunas partes de la cáscara para tratar de reorganizarlas y que de esta forma pudiera encajar. El mismo tipo de idea se aplica al intentar crear un mapa. Las proyecciones vienen a solventar este problema, estudiando las distintas opciones para obtener la forma más exacta posible en el mapa.

Las proyecciones se proyectan sobre una superficie geométrica, tres de las más comunes son: proyecciones planas, cónicas o cilíndricas. Imaginemos un cilindro que se envuelve alrededor de la Tierra, con el centro de la circunferencia del cilindro tocando el ecuador. Ahora, la Tierra es proyectada sobre la superficie de este cilindro, y si se corta el cilindro de arriba abajo verticalmente y se desenvuelve y se coloca plano, se tendría una proyección cilíndrica regular.

La proyección Spherical Mercator es la proyección que se usa en este trabajo de investigación. Esta proyección presenta una buena aproximación en su zona central, pero las zonas superior e inferior correspondientes a norte y sur presentan grandes deformaciones. Google Maps y Virtua Earth 2d, actualmente usan la proyección Mercator. A pesar de sus relativas distorsiones de escala, esta proyección está muy indicada para mapas interactivos en que se hacen desplazamientos y acercamientos de regiones pequeñas, donde las formas se distorsionan

relativamente poco.



## Sistemas de información geográfica

Un Sistema de Información Geográfica (también conocido con los acrónimos SIG en español o GIS en inglés) es un conjunto de herramientas que integra y relaciona diversos componentes (usuarios, hardware, software, procesos) que permiten la organización, almacenamiento, manipulación, análisis y modelización de grandes cantidades de datos procedentes del mundo real que están vinculados a una referencia espacial, facilitando la incorporación de aspectos sociales-culturales, económicos y ambientales que conducen a la toma de decisiones de una manera más eficaz.

En cuestiones de hardware los SIG están conformados de un ordenador o grupo de ordenadores y periféricos de entrada y salida que constituyen el soporte físico del SIG. Los productos de software para los SIG proveen las herramientas y funcionalidades necesarias para almacenar, analizar y mostrar información geográfica, los componentes principales son:

- Sistemas para administrar bases de datos.
- Una interfaz gráfica de usuarios para el fácil acceso a las herramientas.
- Herramientas para captura y gestión de la información.
- Herramientas para soporte de consultas, análisis y visualización de los datos espaciales.

Actualmente, con el mayor acceso a Internet y a la demanda de datos geográficos distribuidos, el software SIG ha cambiado gradualmente su perspectiva hacia la distribución de datos a través de la red. Los SIG que en la actualidad se comercializan son combinaciones de varias aplicaciones interoperables y APIs.

Datos geográficos: Es el componente más importante. Pueden obtenerse por recursos propios o a través de proveedores de datos. A su vez es el elemento diferenciador de un SIG frente a un Sistema de Información, ya que este tipo de datos contiene dos vertientes: la espacial y la temática. Procedimientos: Para que un SIG tenga éxito, este debe operar de acuerdo a un plan bien diseñado y estructurado y acorde a las reglas de la empresa, que son los modelos de las actividades propias de cada organización. Equipo humano: Hay distintas formas de interactuar con este tipo de programas, dependiendo del tipo de usuario.

## Bibliografía

- [1] *OpenLayers 3 Beginners Guide*, Packt Publishing Ltd, 2nd ed., Enero 2015.
- [2] *Learning JavaScript Data Structures and Algorithms*, Packt Publishing Ltd, 2nd ed., Junio 2016.
- [3] *Mastering OpenLayers 3*, Packt Publishing Ltd, 2nd ed., Enero 2016.
- [4] C. P. MUKI HAKLAY, ALEX SINGLETON, *Web mapping 2.0.*; Geography Compass, 2 (2008), p. 2011 2039.
- [5] S. MUMBAIKAR, P. PADIYA, AND D. O. C. ENGINEERING, *Web services based on soap and rest principles*, 2013.
- [6] OGC, *Standards and supporting documents*.  
Recuperado de <http://www.opengeospatial.org/standards>.
- [7] D. RETHANS, *Geolocation and maps with php*.  
Recuperado de <https://derickrethans.nl/talks/maps-tek11.pdf>.
- [8] P. P. SNEHAL MUMBAIKAR, *Web services based on soap and rest principles*, International Journal of Scientific and Research Publications, 3 (2013), pp. 2250–3153.
- [9] M. TEAM, *Web mapping*.  
Recuperado de <https://sites.google.com/site/mapmakerpedia/maps-101/web-mapping>.

