



**INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO**



Trabajo Terminal:

**“Ambienta2MX”**  
2014B-073

Presentan:

**Jimenez García Eduardo Gamaliel  
Reséndiz Arteaga Juan Alberto**

**Palabras Clave:** Desarrollo Web, Sistemas distribuidos, Aplicaciones para las comunicaciones en red.

Directores:

**M. en C. Ramírez Morales Mario Augusto, M. en E. Silva Sánchez Carlos**

# Índice general

# Capítulo 1

## Introducción

Ambienta2MX pretende ser una plataforma única en su tipo a nivel nacional, proporcionando la infraestructura lógica, modelo de datos, diagramas y esquemas lógicos necesarios para proveer un servicio de consulta de datos climatológicos e índices de contaminación a través de un portal web y/o servicios expuestos, para fines privados, públicos o sociales en general.

Han existido diversos sistemas que proveen información semejante o han intentado dar solución a la problemática expuesta no solo considerando datos ambientales como la temperatura sino geográficos, geodésicos, topográficos, etcetera; sin embargo, no han tenido el impacto o el apoyo necesario para crecer y proveer la infraestructura lógica y/o física para satisfacer ese problema.

Para llegar a la solución que será descrita a lo largo de éste documento fue necesario considerar un grupo específico de soluciones existentes, que van desde sistemas totalmente orientados a la estandarización, hasta soluciones que toman sólo parte del problema completo que pretende atacar Ambienta2MX.

Problemas de este tipo han sido atacados en otros países, teniendo un impacto favorable en la consulta de información de cierta área en específico, casos como el anterior serán mencionados próximamente.



les o a una región en específico del globo terraqueo. Entre los datos almacenados más importantes se encuentra la georeferenciación (Longitud, Latitud, Altitud), algunos datos relativos a la zona, por ejemplo, códigos postales.

En México, la institución encargada del manejo de datos estadísticos y geográficos es el INEGI. El Instituto Nacional de Estadística, Geografía e Informática (INEGI) es la fusión de varias instituciones gubernamentales que funcionaban de forma independiente hasta el año 1983; éstas se encargaban de el manejo de datos estadísticos, comerciales, económicos y financieros.

Actualmente el INEGI, sede se encuentra ubicada en Aguascalientes, Aguascalientes, es la institución encargada de la captación, procesamiento y difusión de información relativa al territorio nacional. [?]

Los datos geospaciales (Longitud, Latitud, Altitud) que brinda el INEGI se encuentran bajo el estandar ITRF 2008 en época 2010. Sin embargo, aún existen sistemas que se encuentran trabajando bajo el formato ITRF92 época 1988 y NAD27 (formato usado hasta 1998), para ello, se brindará soporte a formatos previos, dando prioridad al último estandar existente.

## 2.2. Sistemas de Monitoreo del medio Ambiente.

### 2.2.1. Proyecto en la ciudad de Salamanca

Uno de los proyectos nacionales que realizo un monitoreo usando un PCFM fue realizado en la ciudad de Salamanca- una de las ciudades más contaminadas en México. Este proyecto es titulado “Análisis de la contaminación del aire usando un Algoritmo PCFM aplicado a una base de datos real.

La intención del estudio es analizar la relación entre contaminación y las variables ambientales. En el análisis de este estudio se involucraron datos desde Enero a Diciembre del 2007. Algunas de las variables que se incluyeron fueron S02, y la concentración de partículas menores a 10m y variables meteorológicas. Velocidad del viento, dirección del viento, temperatura y humedad relativa. A partir de este estudio se instalaron algunos centros de monitoreo en Salamanca.

### 2.2.2. Inventario Nacional de Emisiones de Gases de Efecto Invernadero (INEGEI)

En este proyecto se elabora un informe que comprende las estimaciones de las emisiones por fuente y sumidero. Esto se realiza de acuerdo a lo conforme establecido en los artículos 4 y 12 de la Convención marco de las naciones unidas sobre el cambio climático. Este proyecto es una publicación que nos ayuda a saber que deficiencias tenemos, y que efectos pudiera llegar a tener el exceso de algunos contaminantes, sin embargo no es una plataforma para acceder a estos datos tan importantes.

### 2.2.3. Mapa Digital - INEGI

Este proyecto es un sistema de información geográfica que tiene como objetivo facilitar el estudio de los objetos geográficos a través del conocimiento de su ubicación espacio-temporal, así como de atributos asociados; tales servicios brindan al usuario final la posibilidad de:

- Mostrar en forma gráfica la dimensión de la información contenida por medio de acercamientos, selección de capas de información, localizaciones, mediciones, etc.
- Analizar e interpretar los contenidos geográficos y estadísticos mediante operaciones matemáticas, mapas temáticos, gráficos estadísticos, análisis espacial y estadísticos básicos.
- Integrar información a través de la incorporación de datos vectoriales y raster provenientes de archivos locales, conexiones a servicios WMS y base de datos geospaciales de PostGis.

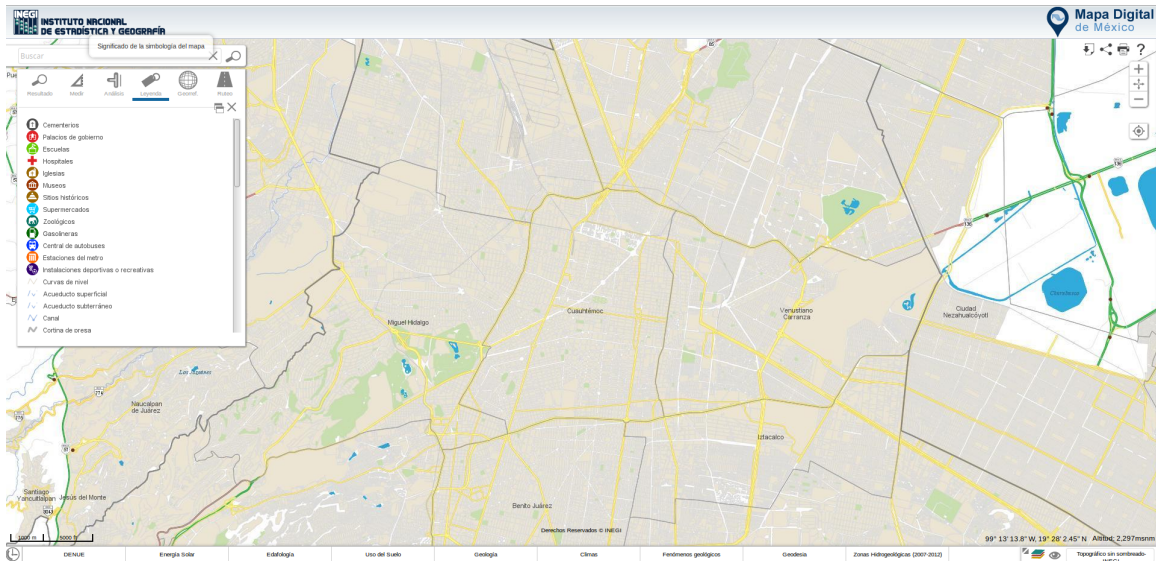


Figura 2.2: Mapa Digital - INEGI.

Toda la información que muestra el mapa digital del INEGI es obtenida por medio de servicios tipo REST, se puede hacer uso de éstos usando Web Scrapping, sin embargo, se encuentra totalmente ligado a la aplicación, trayendo consigo problemas en caso de que se deseara extraer el contenido que ofrecen.

El INEGI y el mapa digital cuentan con algunos puntos de acceso a la información usando como medio principal las peticiones HTTP. Estos servicios suelen ser aislados y requieren de una interacción directa con el humano, es decir, es necesario que éste tenga una interacción mediante escritura o clicks en los formularios de éstos.

## 2.2.4. Base de Datos Estadísticos BADESNIARN

Proyecto de la SEMARNAT que presenta información integrada, revisada y validada con cada una de las fuentes. Además esta estructurada para adecuarse a las necesidades de cada usuario . El usuario en su consulta encontrará el último dato revisado con la fuente, así como la serie histórica disponible en cada caso. Finalmente la plataforma tecnológica detrás le permitirá obtener un archivo electrónico en varios formatos con la info que muestra en pantalla.

The screenshot shows the SEMARNAT website header with the logo and navigation links. The main content area features the title 'BASE DE DATOS ESTADÍSTICOS - BADESNIARN' and a descriptive paragraph. Below this, there is a 'CONSULTA TEMÁTICA' section with a graphic and a description of the thematic search functionality. To the right, a sidebar contains a 'Base de Datos Estadísticos' menu with options for 'Consulta Temática', 'Consulta Dinámica', and 'Contáctanos'. Below the sidebar, there is a 'Publicaciones BADESNIARN' section with a graphic of various reports.

Figura 2.3: Base de datos estadísticos - SEMARNAT.

Tiene un módulo de consultas temática en donde están muy bien delimitados los temas relacionados con el ambiente, y uno dinámica en donde se asocia cada metadato o archivo con palabras clave para que el usuario pueda encontrar el tema de su interés. [?]

## 2.2.5. Forecast IO

Es un atlas del clima que se puede visualizar a través de una página web. En el se pueden consultar datos climatológicos – tales como la temperatura actual, probabilidades de precipitación considerando como modo de visualización base mapas de alta definición.

Este proyecto integra la información de diversas fuentes y también provee un servicio para obtener datos a través de peticiones REST. Una de sus más grandes desventajas es la limitante de peticiones hacia su servicio, ya que después de

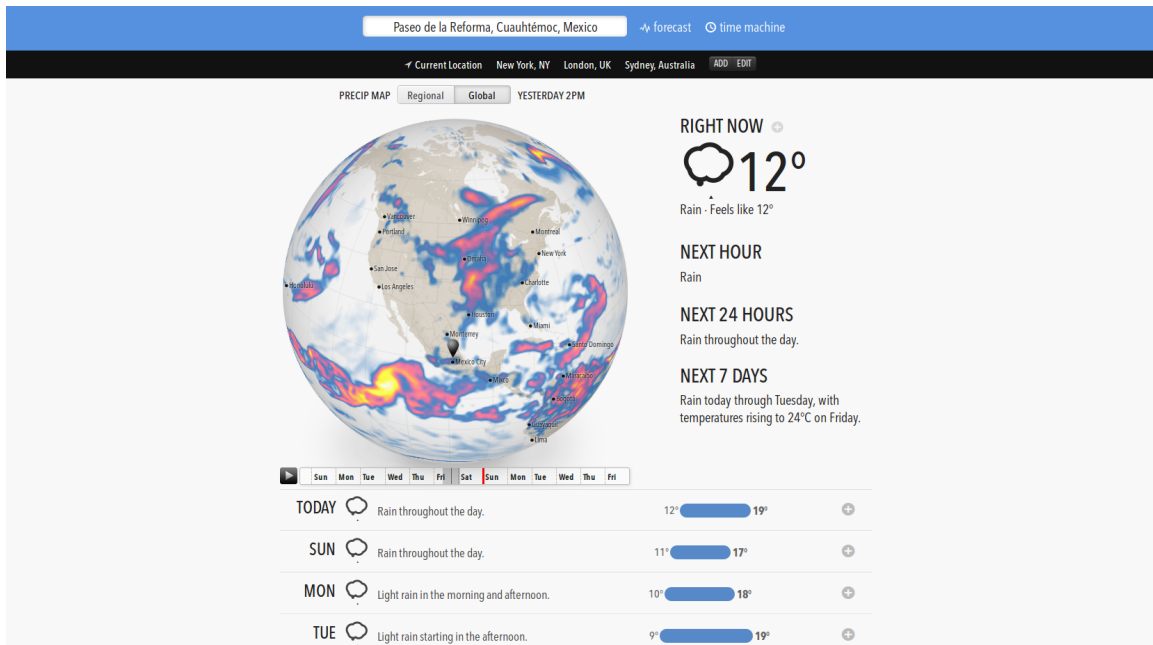


Figura 2.4: Forecast.io

cierta cantidad se aplican cargos a una tarjeta definida al dar de alta la cuenta de desarrollador en la plataforma. [?]

## 2.3. Sistemas de estandarización y unificación de datos.

### 2.3.1. Proyecto INSPIRE.

El Proyecto INSPIRE comenzó a ser desarrollado el 15 de Mayo 2007 y será completamente implementado para el año 2019. Su principal objetivo es es la creación de una infraestructura única de información geoespacial a lo largo de la unión europea.

La infraestructura de datos espaciales asistirá y trabajará a lo largo de todo el territorio Europeo y parte de Asia(Territorio de perteneciente a Rusia), los datos serán diversos considerando temás comunes y técnicos.

El proyecto INSPIRE trabaja bajo algunos principios, por ejemplo:

- Los datos deben ser recabados sólo una vez y mantenerse actualizados de forma efectiva.
- Es posible combinar información espacial de varias fuentes a través del territorio Europeo y compartir sus aplicaciones.
- La información geográfica almacenada en todos los niveles será transparente y compartida.



- Busqueda fácil de de información geográfica que puede ser usada para fines particulares y de uso general.

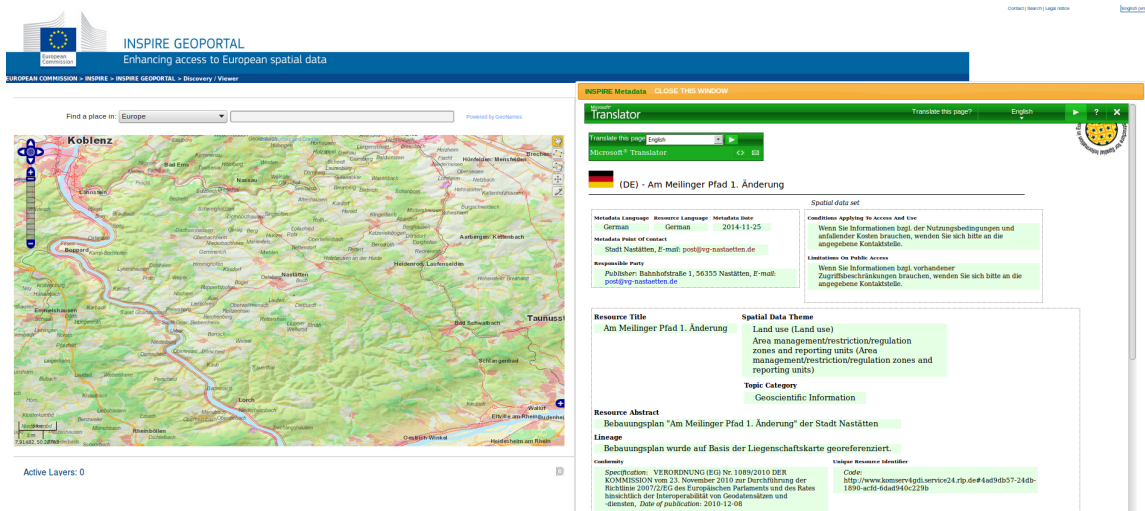


Figura 2.5: Visualización de datos y metadatos en INSPIRE

Actualmente parte de los módulos desarrollados se encuentran en etapas de pruebas. Cuenta con un editor de metadatos que considera la fecha, el nombre de la organización, correo electrónico, palabras clave, ubicación (Latitud, Longitud, Altitud), entre otros. Los metadatos pueden ser también validados utilizando las mismas herramientas que provee la plataforma.

El proyecto INSPIRE cuenta con un proceso de participación de “StakeHolders” para mejorar y analizar los detalles específicos del sistema INSPIRE y todos sus módulos.

### 2.3.2. Geoplatform

Plataforma propuesta por el gobierno de Estados Unidos, principalmente por el Comité Federal de Datos Geográficos (The Federal Geographic Data Committee) [?]. Actualmente funciona como una PaaS (Platform as a Service), cuyos principales objetivos son:

- Unificar y brindar información geoespacial confiable a través de datos y servicios.
- Soporte para toma de decisiones.
- Aplicación para la solución de problemas que pueden ser desarrollados sólo una vez y usados varias veces a través de distintas instituciones federales y otras organizaciones.
- Infraestructura compartida para almacenar datos y aplicaciones.
- Punto focal donde instituciones gubernamentales, académicas, privadas y públicas pueden visualizar información relativa a su región.

## Datasets Published per Month

Data last updated on 05/31/2015 12:50 AM.

Agency Name	Number of Datasets published by month												Total in the Past 12 Months
	Jun '14	Jul '14	Aug '14	Sep '14	Oct '14	Nov '14	Dec '14	Jan '15	Feb '15	Mar '15	Apr '15	May '15	
Department of Agriculture	-	-	22	-	6	4	-	2	11	11	3	4	63
Department of Commerce	38	86	1	368	28165	3580	845	398	394	156	377	193	34601
Department of Defense	1	-	-	-	-	-	-	-	-	-	-	-	1
Department of Homeland Security	-	-	-	-	-	-	-	-	-	-	1	-	1
Department of Housing and Urban Development	12	-	1	-	-	-	-	-	-	-	-	-	13
Department of the Interior	11	134	-	475	404	31	48	22	119	61	161	142	1608
Department of Transportation	-	-	-	-	-	-	-	-	-	-	57	-	57
Environmental Protection Agency	-	-	-	-	-	-	-	-	-	-	1377	248	1625
<b>Total</b>	62	220	24	843	28575	3615	893	422	524	228	1976	587	37969

Figura 2.6: Datasets publicados por Geoplatform

Actualmente usan un estándar de datos abierto conocido como “Common Core Metadata Schema” bajo su versión 1. [?]

La implementación de esta plataforma fue motivada por los principios y espíritu de “Open Government”[?], cuya principal visión es la de enfatizar la comunicación, contabilidad y transparencia entre gobierno-ciudadano, dentro del territorio estadounidense. Todos los datos, aplicaciones y servicios que provee Geoplatform se encuentran bajo licencias libres.

La plataforma fue desarrollada por miembros de la el comité federal de datos geográficos (FGDC por sus siglas en inglés) mediante la colaboración de escuelas y usuarios expertos en el área. La audiencia que toma como fuente incluye agencias federales, estatales y locales, además de sector privado, educativo y al público en general. [?]

### 2.3.3. GeoMap México

GeoMap es una Plataforma Geoespacial Integral, que aprovecha las tecnologías informáticas actuales, implementando metodologías y estándares internacionales. Contiene todas las funcionalidades de un Sistema de Información Geográfica (SIG) y ofrece todos los servicios necesarios para la creación de aplicaciones geoespaciales. Entre sus principales funcionalidades se encuentran:

- Multiplataforma (Versión Escritorio, Web y para Dispositivos Móviles).
- Módulo de Mapas Dinámicos.
- Interacción con diferentes Servidores de Mapas.
- Conexión a distintas Bases de Datos Geoespaciales.

- Importación e Integración de información geográfica de diferentes fuentes y formatos.
- Visualización de Imágenes de Satélite, Ortofotos y de UAV (Vehículo Aéreo No Tripulado).
- Módulo de Administración y Autenticación de Usuarios.



Figura 2.7: Información publicada por GeoMap

La plataforma Informática responde a la necesidad cada vez más frecuente de utilizar mapas y servicios geográficos en aplicaciones Web y para la Nube. Incluye las funcionalidades de integración y administración de información cartográfica, visualización y procesamiento de información geográfica y tabular, conexión a bases de datos geoespaciales e implementación de análisis espacial avanzado. La plataforma se especializa en temas como la geocodificación y geolocalización, áreas de influencia, zonas de cobertura, ubicación de puntos de interés, visualización, búsqueda y análisis de datos geográficos.

## 2.4. Características climáticas y de contaminación en México.

### 2.4.1. ¿Cuándo se iniciaron los problemas de contaminación del aire?

Desde siempre la humanidad ha emitido contaminantes al aire, pero esto se incrementó de manera dramática a partir de la Revolución Industrial iniciada en el Reino Unido a finales del siglo XVII. En esa época, el trabajo manual fue remplazado por maquinaria, básicamente por la introducción de tecnologías que empleaban el vapor y que hacían posible tener altos niveles de producción. El problema fue que con estos avances industriales se incrementó el uso de combustibles, tal como el petróleo y el carbón mineral, ambos indispensables para el funcionamiento de la nueva maquinaria.

Desde entonces el problema de contaminación del aire se ha convertido en una constante en muchas ciudades industriales de todo el mundo, lo que ha ocasionado problemas de salud a su población.

Algunos de los casos más dramáticos y graves son la famosa niebla tóxica londinense de 1952, el deterioro de los bosques europeos por la lluvia ácida en los años cincuenta y sesenta del siglo XX, y la grave situación de la calidad del aire en la Ciudad de México, Tokio y Sao Paulo durante las últimas décadas del siglo anterior.

#### 2.4.2. ¿Cuáles son los contaminantes y qué efectos tienen?

Los contaminantes pueden ser emitidos de forma natural o por actividades relacionadas con el ser humano. Los fenómenos naturales que se producen en la superficie o en el interior de la tierra –como el caso de erupciones volcánicas, que produce emisiones de gases, vapores, polvos y aerosoles-, también contribuyen a la contaminación del aire.

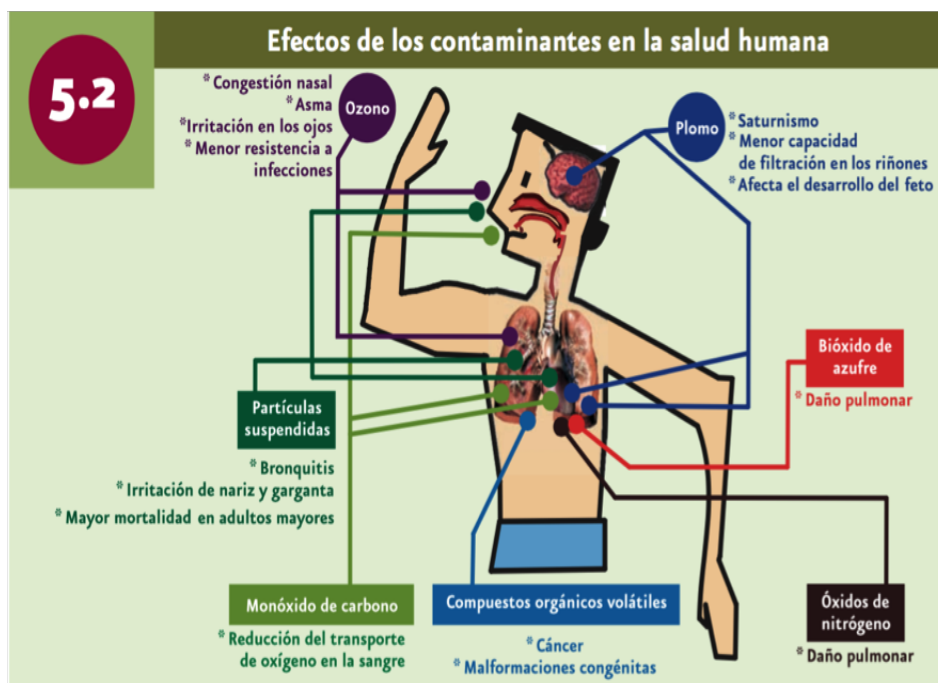


Figura 2.8: Efectos de los contaminantes en el cuerpo.

Los principales contaminantes relacionados con la calidad del aire son el bióxido de azufre (SO<sub>2</sub>), el monóxido de carbono (CO), los óxidos de nitrógeno (NO<sub>x</sub>), las partículas suspendidas, el plomo y el ozono.

Las plantas, animales y otros organismos también resienten los efectos de contaminantes como el ozono. Principalmente con la formación de la lluvia ácida, dicha lluvia es ocasionada con la presencia de ciertos ácidos en la atmósfera

que se precipitan a la tierra con la lluvia. El dióxido de azufre y los óxidos de nitrógeno, resultado de la quema de combustibles fósiles causan lluvia ácida, ya que al combinarse con agua, oxígeno y otros compuestos químicos forman ácidos como el ácido sulfúrico y el nítrico.

Las plantas se ven afectadas por esta lluvia ya que los ácidos pueden obstruir y acidificar los diminutos poros de las hojas por los que las plantas toman el aire que necesitan para realizar la fotosíntesis, además la lluvia ácida degrada los suelos, lo cual afecta las raíces y la nutrición de las plantas.

En el parque nacional Izta-Popo, Zoquiapan y en el Parque Nacional Desierto de los Leones, la lluvia ácida a dañado la vegetación. Estos daños involucran la pérdida de hojas y ramas, crecimiento lento y vulnerabilidad a ataques de plagas y enfermedades.

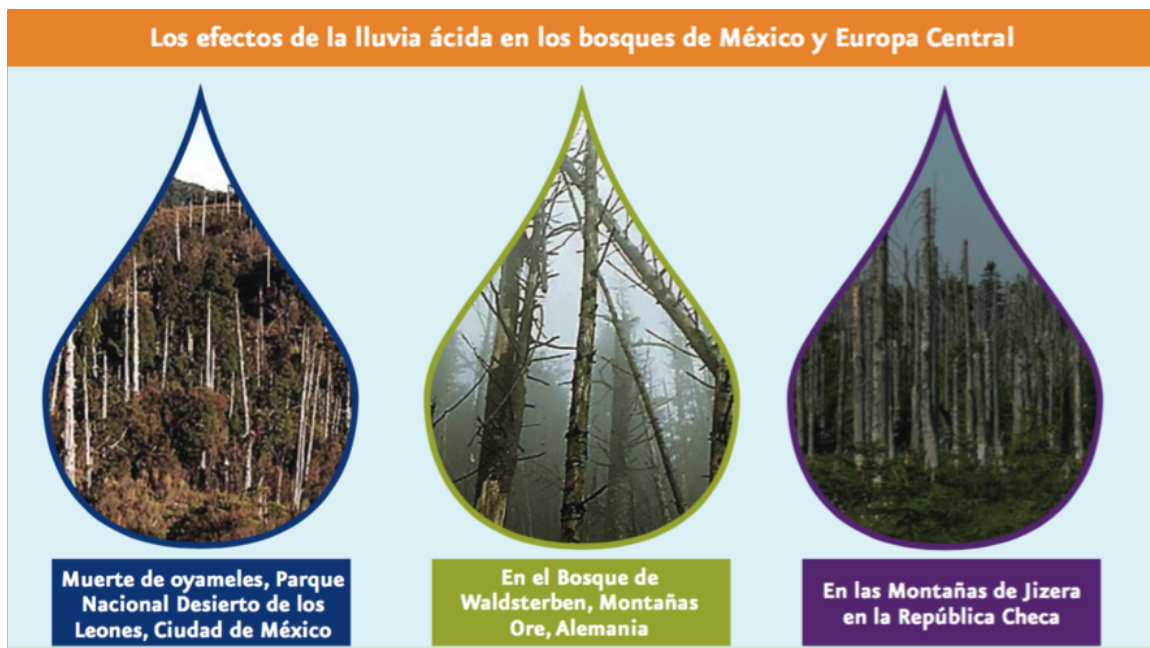


Figura 2.9: Impacto de la lluvia ácida en los bosques de México.

Por otro lado los ríos, lagos y lagunas también pueden hacerse más ácidos por efecto de la lluvia ácida, lo cual pone en serio riesgo a las especies de plantas y animales que los habitan. Algunos ejemplos de estos daños se encuentran en los lagos del norte de Europa, en los que se ha reportado incluso que han quedado sin ninguna forma de vida luego de la contaminación por lluvia ácida.

Por otro lado también los monumentos y edificios sufren deterioros por la lluvia ácida, ya que los ácidos funcionan como agente corrosivo. El laboratorio de restauración del Instituto de Investigaciones Antropológicas de la UNAM indica que en los últimos 25 años el deterioro de los monumentos y edificios históricos de



Figura 2.10: Daños en edificaciones por lluvia ácida.

la ciudad de México se ha acelerado de manera impresionante por el incremento de los niveles de contaminación.

#### 2.4.3. Quiénes generan los contaminantes atmosféricos?

En México al igual que en otros países se han desarrollado inventarios de emisiones que proporcionan información sobre la cantidad de contaminantes que se liberan al aire. En el año de 1999 de acuerdo al inventario de emisiones a nivel nacional se produjeron 40.5 millones de toneladas de las cuales el 58 % correspondieron a fuentes naturales- es decir, el suelo, la vegetación y las actividades volcánicas- y 42 % a la contaminación de origen humano.

A pesar de que aparentemente las fuentes naturales sean las mayores productoras de contaminación, son las fuentes antropogénicas las que están cerca de la población y las que influyen en mayor medida en la calidad del aire que se respira.

Dentro de las fuentes antropogénicas, los vehículos automotores son los mayores productores de contaminantes, después la quema de gas LP y al final las emisiones de plantas generadoras de electricidad.

#### 2.4.4. ¿Qué hemos hecho para resolver el problema?

México lleva tiempo tomando acciones para resolver estos problemas, en 1988 implementó el Sistema Nacional del Inventario de Emisiones de Fuentes Fijas, así como un proyecto para cuantificar las emisiones del Valle de México. A partir del monitoreo de la calidad del aire ha diseñado algunas mejoras como eliminar el plomo en la gasolina, reducción del contenido de azufre.

Actualmente también existe una red de monitoreo atmosférico que abarca 52 ciudades y zonas metropolitanas que mide los niveles de contaminación presentes en el país. La concentración de los contaminantes en el aire se obtiene mediante la toma de muestras de aire que se analizan y procesan.

A partir de estas mediciones nacionales se han detectado cuales son las localidades con mayor índices de contaminación así como los contaminantes que emiten principalmente. Uno de los esfuerzos más notables son los programas para mejorar la calidad del aire (Proaires) que buscan revertir las tendencias de deterioro, ya que incorporan medidas para el control y abatimiento de las emisiones de los contaminantes. También existen centrales eólicas para generar electricidad a partir de la energía del viento, una en la Venta, Oaxaca y la otra en Guerrero Negro, Baja California Sur.

# Capítulo 3

## Metodología

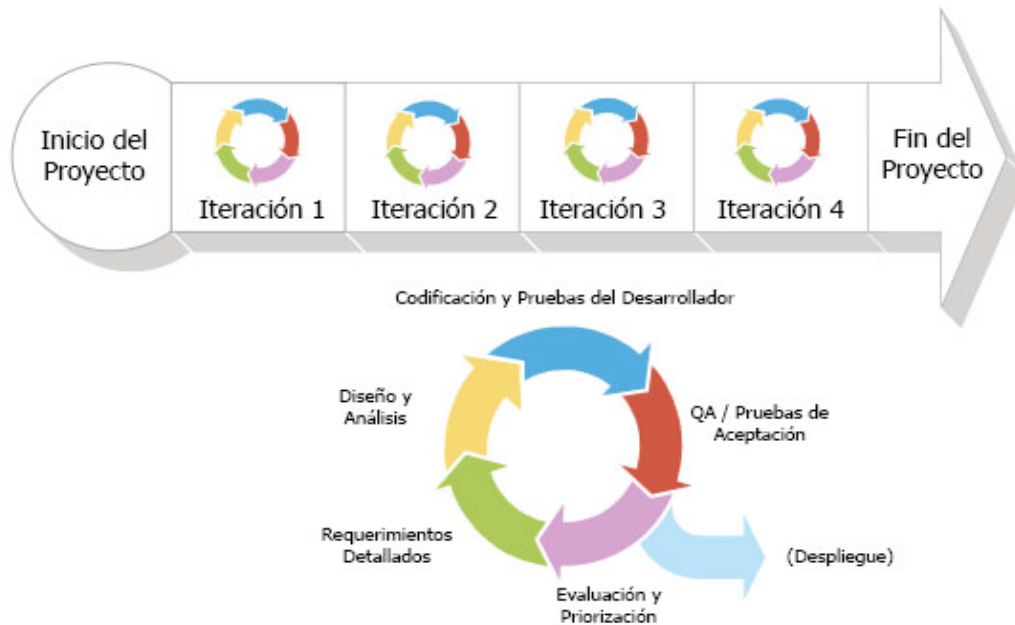
### 3.1. Desarrollo ágil de software

El desarrollo ágil propone una alternativa al desarrollo de software tradicional. Los enfoques del desarrollo ágil son principalmente usados en el desarrollo de software para ayudar a las compañías a responder fácilmente al cambio. [?]

#### 3.1.1. Scrum

Scrum es un marco de gestión para el desarrollo incremental de un producto que proporciona una estructura de roles, reuniones, reglas y artefactos.

Scrum utiliza iteraciones de longitud fija denominados Sprints, que son típicamente de dos semanas o 30 días de duración. Los equipos de Scrum intentan generar un incremento de producto potencialmente entregable (debidamente probado) en cada iteración. [?]





### 3.1.2. Programación Extrema (XP)

Es un enfoque disciplinado para entregar software de alta calidad rápida y continuamente. Promueve una alta participación del cliente, retroalimentación rápida, pruebas continuas, planeación continua y entrega de software funcional en intervalos muy frecuentes que van de 1 a 3 semanas. [?]

## 3.2. Técnicas de desarrollo ágil

### 3.2.1. Desarrollo guiado por pruebas (Test Driven Development)

El desarrollo guiado por pruebas es una técnica avanzada que hace uso de pruebas unitarias para guiar el diseño del software y forzar el desacoplamiento de dependencias.

El resultado de usar esta práctica es una comprensiva suite de pruebas que pueden ser ejecutadas en cualquier momento para proporcionar información de que el software está aún funcionando. [?]

### 3.2.2. Refactor

Es una técnica disciplinada para la reestructuración de un cuerpo de código existente, alterando su estructura interna sin cambiar su comportamiento exterior.

Cada transformación es pequeña, pero una secuencia de transformaciones pueden producir una importante reestructuración. [?]

### 3.2.3. Integración continua

Es una práctica de desarrollo que requiere a los desarrolladores la integración de código en un repositorio compartido varias veces al día.

Cada registro de entrada es entonces verificado por un proceso de construcción automatizado permitiendo a los equipos detectar problemas a tiempo.

La integración continua trae muchos beneficios, entre ellos la detección de errores rápidamente y reducir los problemas de integración lo que permite entrega de software más rápidamente. [?]

## 3.3. Justificación

Para el desarrollo del sistema pondremos en práctica las técnicas mencionadas anteriormente que son características del marco de trabajo Scrum y de la metodología Extreme Programming.

**El proceso de desarrollo del proyecto se llevará a cabo de la siguiente manera:**

- Planteamiento de las historias de usuario.
- Definición del plazo de tiempo para cada sprint y el conjunto de funcionalidades relacionadas a cada historia de usuario que serán desarrolladas dentro del mismo.
- Implementación de pruebas unitarias en conjunto con el desarrollo de cada módulo de la aplicación.
- Implementación de pruebas de integración para verificar la comunicación entre dos o más módulos.
- Despliegue de la aplicación en un ambiente productivo una vez que todas las pruebas hayan pasado.
- Implementación de pruebas funcionales para verificar que el flujo descrito en las historias de usuario se lleva a cabo correctamente.

### **3.4. Modelo por prototipos**

**Es un modelo para el desarrollo de sistemas en el cual un prototipo es construido, probado y finalmente reconstruido las veces que sea necesario hasta que un prototipo aceptable es finalmente alcanzado del cual el sistema completo o producto puede ser totalmente desarrollado.**

**Este modelo funciona bien en escenarios donde no se conocen por completo los requerimientos.**

#### **3.4.1. ¿Qué es un prototipo de software?**

**Es un modelo de software con una funcionalidad limitada que permite al usuario evaluar los propósitos del desarrollador y probarlos antes de su implementación.**

**También ayuda a entender los requerimientos específicos del usuario y que no pudieron haber sido considerados por el desarrollador durante el diseño del sistema.**

#### **3.4.2. Implementación del modelo de prototipos**

- Los nuevos requerimientos del sistema se definen con el mayor detalle posible.
- Un diseño preliminar es creado para el nuevo sistema.
- Un primer prototipo del nuevo sistema es construido para el diseño preliminar que representa una aproximación a las características del producto final.
- El usuario evalúa el primer prototipo, notando sus fortalezas y debilidades, qué necesita agregarse y que debería removerse. El usuario recoge las observaciones de los usuarios.
- El primer prototipo es modificado, basado en los comentarios hechos por el usuario, y un segundo prototipo del nuevo sistema es construido.

- El segundo prototipo es evaluado del mismo modo que el primero.
- Los pasos anteriores son iterados tantas veces como sea necesario, hasta que los usuarios consideren que el prototipo representa al producto final deseado.
- El sistema final es construido, basado en el prototipo final.
- El sistema final es evaluado y probado a fondo. Existe una rutina de mantenimiento continuo para prevenir errores a gran escala.

### **3.4.3. Integración del modelo orientado a prototipos, el framework Scrum y la metodología Extreme Programming**

Al desarrollar el proyecto utilizando el modelo orientado a prototipos nos fue posible incorporar varias características del framework Scrum y la metodología XP.

Nos dimos cuenta de que el planteamiento de las historias de usuario al inicio del desarrollo de un prototipo nos permitía describir rápidamente los requerimientos principales y definir criterios de aceptación para considerar cuando una funcionalidad o flujo de la aplicación estaba completo.

Otra de las características que tomamos del framework Scrum fueron los Sprints; los cuales implementamos para enfocarnos en el desarrollo de cierta funcionalidad que agrupamos en un periodo de tiempo. La duración de cada Sprint fue de dos semanas.

De igual modo, hicimos uso de algunas técnicas características de la metodología XP como el desarrollo guiado por pruebas al construir nuestro primer prototipo; en el cual, primero se definen un conjunto de pruebas de unidad y en base a ellas se va desarrollando la funcionalidad para que las pruebas pasen sin problemas. Al hacer esto se asegura que la funcionalidad de un módulo pueda ser extendida fácilmente.





## Capítulo 4

# Tecnologías

Para el desarrollo del proyecto decidimos utilizar un lenguaje de programación dinámico que corre sobre la JVM (Groovy) y el conjunto de herramientas que existen en su ecosistema. Además, para la parte de la aplicación web será necesario integrar tecnologías JavaScript que nos ayuden en la interacción con la aplicación que expone datos para mostrar la información que se requiera.

A continuación se describen las ventajas de las tecnologías que hemos decidido utilizar para el desarrollo del proyecto

Nombre	Ventajas
	<ul style="list-style-type: none"> <li>▪ Implementa funciones espaciales para encontrar información relevante de ubicaciones específicas.</li> <li>▪ Útil para el procesamiento de grandes cantidades de información.</li> </ul>
	<ul style="list-style-type: none"> <li>▪ Lenguaje expresivo que incrementa la productividad. Se reduce el azucar sintactico en comparación con lenguajes como C++ y Java y la curva de aprendizaje es muy pequeña si ya se conoce Java.</li> <li>▪ Se pueden integrar fácilmente todas las bibliotecas de Java ya que corre sobre la JVM.</li> <li>▪ Closures</li> </ul>
	<ul style="list-style-type: none"> <li>▪ Sigue un enfoque de construcción por convención.</li> <li>▪ A diferencia de otras herramientas para la construcción de proyectos como Maven o Ant que utilizan XML, Gradle hace uso de un poderoso lenguaje específico de dominio (Groovy) para la definición de las tareas que deben ejecutarse.</li> <li>▪ Cuenta con un administrador de dependencias que se encarga de descargarlas y las deja disponibles para su uso en la aplicación.</li> </ul>
	<ul style="list-style-type: none"> <li>▪ Genera la estructura lógica para aplicar el patrón de arquitectura Model Vista Controlador, ya que se encuentra construido sobre Spring MVC.</li> <li>▪ Buen soporte de pruebas unitarias, de integración y funcionales.</li> <li>▪ Capa de Mapeo Objeto Relacional que trabaja sobre Hibernate para las operaciones transaccionales.</li> </ul>
	<ul style="list-style-type: none"> <li>▪ Gran escalabilidad</li> <li>▪ Canal distribuido de mensajes como medio de comunicación.</li> <li>▪ Herramienta políglota.</li> </ul>

	<ul style="list-style-type: none"> <li>▪ Permite la automatización y ejecución de tareas para la construcción de un proyecto JavaScript.</li> </ul>
	<ul style="list-style-type: none"> <li>▪ Define la estructura de directorios y archivos para un proyecto JavaScript.</li> <li>▪ Cuenta con varios generadores para crear el código repetitivo que se necesita para iniciar un proyecto.</li> <li>▪ Define tareas para que el desarrollador se concentre sólo en realizar la funcionalidad de la aplicación.</li> </ul>
	<ul style="list-style-type: none"> <li>▪ Incrementa la productividad al desarrollar aplicaciones single page.</li> <li>▪ Permite la escalabilidad de una aplicación JavaScript.</li> <li>▪ Un modelo de objetos rico y plantillas que se enlazan a los datos.</li> <li>▪ Cuenta con propiedades calculadas de un modelo de datos.</li> </ul>
	<ul style="list-style-type: none"> <li>▪ Automatización del navegador para realizar pruebas funcionales.</li> <li>▪ Opción para hacer screenshots de algún flujo de la aplicación.</li> </ul>

Cuadro 4.1: Descripción de las tecnologías

# Capítulo 5

## Modelo de Datos

### 5.1. Justificación

Se hará uso de un modelo de base de datos orientado a documentos. También se contará con un esquema no relacional, es decir, se carece de una normalización definida, haciendo uso de MongoDB, que se encargará de persistir y manejar las estructuras de datos en documentos JSON. Esta base de datos pertenece a la categoría NoSQL.

Las características de una base de datos NoSQL son las siguientes:

- Modelo de datos flexible.
- Buen rendimiento en clusters.
- Sin esquemas.

Este tipo de bases resultan útiles debido a que se se pretende procesar grandes volúmenes de información para mostrar un historial de la información climática que se vaya persistiendo.

Las tecnologías para grandes volúmenes de información son relativamente nuevas; estas surgieron debido a la necesidad que tenían empresas grandes como Google y Amazon.

La información fue convertida de un modelo relacional a un modelo basado en documentos, jerárquico o basado en columnas usando proceso de denormalización, ésto con la finalidad de dar un orden a la información considerando simplemente consultas de cierto tipo evitando así operaciones típicas del álgebra relacional cómo el Producto Cartesiano, que implicaba el uso de grandes recursos.

Las tecnologías de tipo NoSQL se orientan a consultas y no a transacción. Lo que brinda un fácil acceso a la información, una estructura de los datos orientada a su posterior análisis, respuestas en tiempo real y una gran capacidad de escalar y replicar. [?]

Empresas como Foursquare, Google, Amazon, Uber o Twitter, han optado por complementar la información que persisten de forma relacional con modelos orientados a documentos usando tecnologías como Hbase, MongoDB, BigTable, DynamoDB, por citar algunos ejemplos. [?] [?]

## 5.2. Descripción

Considerando la problemática y solución que plantea el equipo de Ambienta2MX, se ha optado por orientar el sistema a consultas usando documentos como modelo de datos base.

La información que será guardada y posteriormente consultada por otros módulos del ecosistema Ambienta2MX seguirá un esquema propuesto por el equipo de trabajo, éste recopila la estructura de sistemas que proveen información climática como lo son weather.gov, forecast.io, Weather Underground, proyecto INSPIRE (Unión Europea), Servicio Meteorológico Nacional, entre otros. [?] [?]

Para la persistencia de la información se han definido los siguientes documentos:

- Places
- Pollution
- Weather

Los documentos serán almacenados en MongoDB considerando los datos definidos en la especificación JSON Data Interchange Format (en su versión 2013). [?]

El equipo de Ambienta2MX ha considerado el uso de el formato antes mencionado debido a su facilidad de lectura e integración con otro tipo de plataformas y lenguajes de programación, actualmente es el formato que rige el manejo de API de tipo REST.

Dentro de los modelos existe información relativa a los proveedores o fuentes de información, ésta información será utilizada para fines de consulta y contar con el control del origen de los datos.



### 5.2.1. Places

```
{
  "location": { // Using GeoJson Spec
    "type": "Point",
    "coordinates": [Number]
  },
  "sexagesimal_coordinates": [Number], // Original coordinates from INEGI
  "source": String,
  "itrf_coordinates": [Number], // Converted coordinates to itrf2008 format.
  "nad27_coordinates": [Number], // Previous coordinates
  "height": Number, // height of the place
  "town": String, // name of the town
  "state": String, // name of the state
  "city": String, // name of the city
  "fullName": String, // Place full name,(Search)
  "zipCode": String, // Optional information
  "extraInfo": [String], // Extra information
  "provider": [Object], // Information about the providers (INEGI, Google,
    Geohack, etc) as provider:{information:content,information:url, ...}
  "lastUpdated": Date, // Grails generated information
  "dateCreated": Date, // Grails generated information
}
```

## 5.2.2. Pollution

```
{
  "location": { // Using GeoJson Spec
    "type": "Point",
    "coordinates": [Number],
  },
  "height": Number, // height of the place
  "airQuality": Number, // Air Quality (Percentage)
  "ozone": Number, // Ozone index (Percentage)
  "sulphurDioxide": Number, // Sulphur Dioxide index (Percentage)
  "nitrogenDioxide": Number, // Nitrogen Dioxide index (Percentage)
  "carbonMonoxide": Number, // Carbon Monoxide index (Percentage)
  "UV": Number, // Ultraviolet index
  "provider": [Object], // Providers list as provider:{information:content,
    information:url, ...}
  "sampleDate" : Date, // Sample date
  "lastUpdated": Date, // Grails generated information
  "dateCreated": Date // Grails generated information
}
```

### 5.2.3. Weather

```
{
  "location":{ // Using GeoJson Spec
    "type":"Point",
    "coordinates":[Number],
  },
  "height":Number, // height of the place
  "description": String, // Description about the weather
  "rainfallIntensity": Number, // Rain Intensity (Percentage)
  "rainfallProbability": Number, // Rain probability (Percentage)
  "temperature": Number, // Celsius (desired)
  "apparentTemperature": Number, // Apparent Temperature (Some sources could
    provide this information)
  "dewPoint": Number, // Dew Point
  "humidity": Number, // Humidity index (Percentage)
  "windSpeed": Number, // Wind speed (Km/h)
  "windDirection": Number, // Wind direction, considering eight cardinal
    points
  "windBearing": Number, // Wind direction (Consdiering 360 degrees)
  "visibility": Number, // Visibility (Percentage)
  "cloudCover": Number, // Clouds on the sky (Percentage)
  "pressure": Number, // Pressure (mmHg)
  "provider": [Object], // Providers list as provider:{information:content,
    information:url, ...}
  "sampleDate": Date, // Sample date
  "lastUpDated": Date, // Grails generated information
  "dateCreated": Date // Grails generated information
}
```

## Capítulo 6

# Definición temática de Ambienta2MX

### 6.1. Objetivo general

Considerando el problema de información y estructura de datos que tiene México en la actualidad, además de seguir la tendencia y aprovechar la brecha que ha disminuido el uso de datos abiertos, el equipo de Ambienta2MX decidió afrontar la tarea de desarrollar una herramienta que permita la conceptualización de la información de variables ambientales e índices de calidad y contaminación del aire que el INEGI y otras instituciones públicas o privadas almacenan y/o exponen para fines educativos, informativos o de uso particular.

### 6.2. Justificación

Existe la necesidad de una fuente de información que presente datos climáticos y de contaminación del aire que ofrecen diversas instituciones como el Instituto Nacional de Estadística y Geografía (INEGI), el Servicio Meteorológico Nacional y la Comisión Nacional del Agua de manera estandarizada. Esto con la finalidad de ordenar y categorizar un conjunto de metadatos.

## Capítulo 7

# Descripción y Módulos de Ambienta2MX

### 7.1. ¿Qué y para qué es Ambienta2MX?

Ambienta2MX es el nombre de la plataforma que pretende formar parte de una macro solución orientada a la estandarización de datos geoespaciales que el INEGI y otras instituciones públicas tienen en su haber.

Actualmente no existe un estándar de datos geográficos a nivel nacional. Han existido aproximaciones mediante concursos que instituciones públicas como el INEGI ha publicado, o simplemente han existido propuestas que han brindado una solución incompleta a la unión y manejo de información geográfica, geodésica, hidrográfica, climática, topográfica, etc.

Ambienta2MX toma parte de todo el problema y propone una infraestructura lógica para afrontar la estandarización de variables ambientales y algunos índices de contaminación. Esta información actualmente se encuentra en formatos muy rudimentarios como textos planos sin algún protocolo o definición para su interpretación.

Sistemas semejantes, por ejemplo, el Servicio Meteorológico Nacional carece de algún recurso del cual se puedan realizar consultas que no sea mediante su portal web, esto trae problemas directos de compatibilidad con otros sistemas.

Un caso semejante tenemos con la información que la Conagua maneja en sus centrales meteorológicas a lo largo del país, los datos que brindan se actualizan de forma periódica y el único medio de acceso es a través de una página de internet que devuelve archivos en formato de texto u hojas de cálculo.

Los impedimentos antes mencionados conllevan a situaciones tan triviales como la consulta de datos para alguna región o punto específico del territorio nacional, al existir diversas fuentes no es posible tener un compendio del cual tomar la información que más nos convenga.

Si a este problema se le añade que los datos carecen de un estandar, llegamos al punto en el que intentar manipular o tratar los datos se vuelve una tarea complicada y en exceso tediosa.

Considerando dichos problemas Ambienta2MX, propone un estandar de datos climáticos tomando como referencia diversas fuentes y adaptando los tipos de datos a tecnologías y tendencias actuales, brindando así una mayor portabilidad y simplicidad en la consulta de información.

## 7.2. Diagrama de Ambienta2MX

Ambienta2MX constará de varios módulos que trabajarán de forma conjunta para satisfacer la necesidad de tener un estandar y un repositorio de datos climáticos a nivel nacional.

Al brindar un sistema modularizado, se genera de forma directa un impacto en el proceso de análisis, desarrollo e integración. Éste tipo de modelo describe de una forma sencilla los componentes necesarios para solventar la demanda a la que se encontrará sometida la plataforma.

A continuación se muestra el diagrama a bloques de Ambienta2MX, todos los módulos, recursos y bases de datos serán descritos de forma posterior.

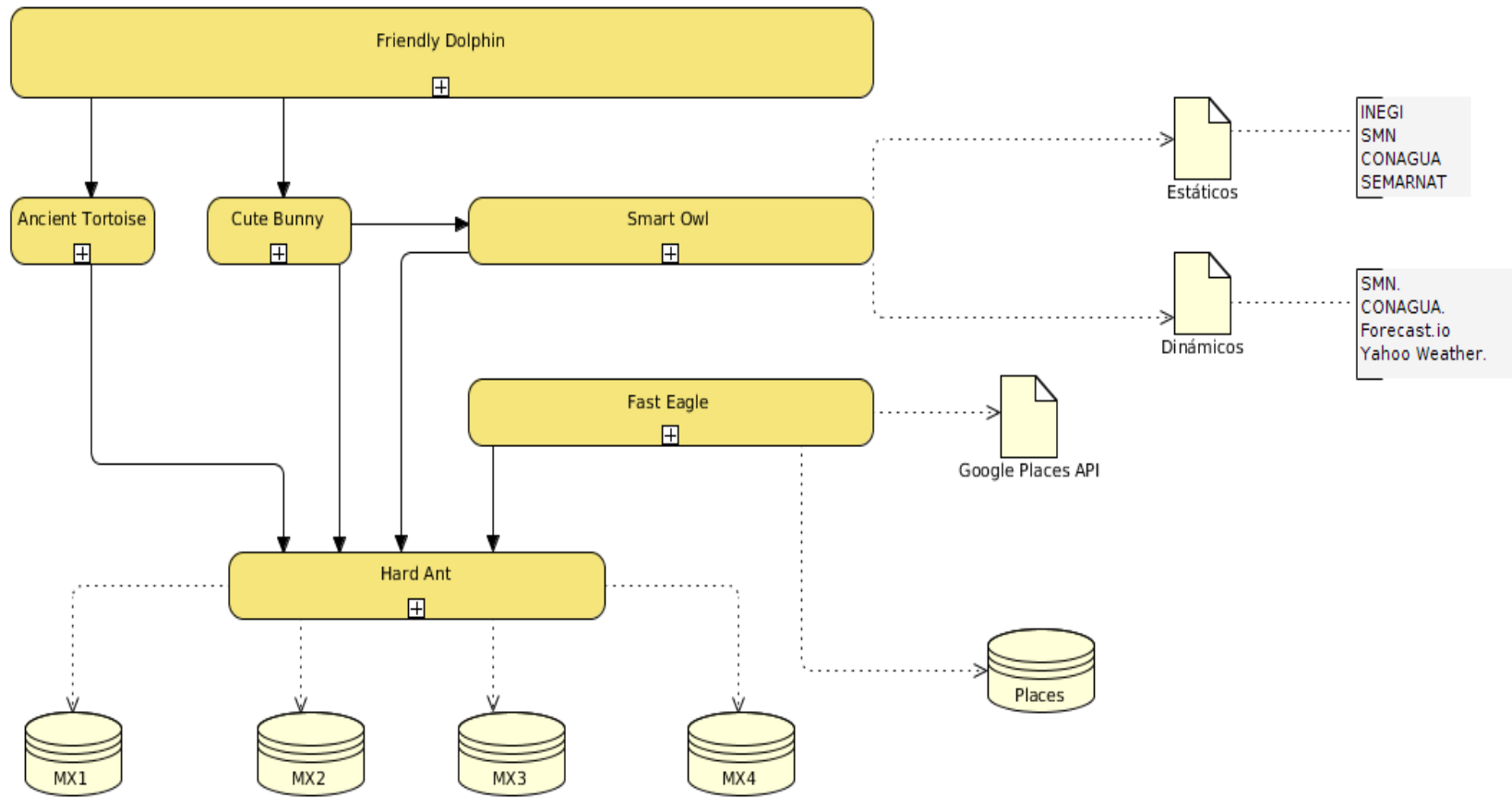


Figura 7.1: Módulos y estructura de Ambienta2MX



Cómo se apreciar en el diagrama, además de los gestores de bases de datos, Ambienta2MX se encuentra dividido en siete módulos básicos:

- Friendly Dolphin.
- Ancient Tortoise.
- Cute Bunny.
- Smart Owl.
- Fast Eagle.
- Hard Ant.

En el mismo diagrama se pueden observar las fuentes que proporcionarán la información ya sea a un nivel estático, por ejemplo, carta climática anual de algún municipio del territorio nacional; o bien, recursos que se actualizan de forma periodica como son los datos que provee el Servicio Meteorológico Nacional.

Se condieran cinco bases de datos, *MX1*, *MX2*, *MX3*, *MX4*, *Places*. Todas las bases del tipo MX contarán con la información de variables ambientales así también de los índices de contaminación de las zonas que conforman al territorio nacional.

Para el caso de *Places*, la base será usada como un macro índice cartográfico del territorio nacional, es decir, esta base será la referencia a nivel latitud, longitud y altitud para ubicar los datos que requieran ser procesados.

Todas las bases se encontrarán funcionando bajo un modelo de base de datos documental teniendo una alimentación bajo demanda, es decir, el contenido gestionado irá aumentando conforme las éstos vayan siendo solicitados.

## 7.3. Módulos de Ambienta2MX

### 7.3.1. Friendly Dolphin

#### Definición

Éste módulo es el encargado de brindar la información procesada al usuario a través de una página de internet. Es el módo visual que los usuarios finales tendrán para poder interactuar con el ecosistema Ambienta2MX.

Se presenta cómo un módulo web que consumirá la información procesada y almacenada por las cuatro bases (*MX1*,*MX2*,*MX3*,*MX4*) y la base de soporte (*Places*).

La principal función es la de consulta y visualización de datos. Es la capa más expuesta y menos técnica de Ambienta2MX ya que es la que tendrá interacción directa con usuarios no técnicos, sin embargo, contará con los procesos necesarios para poder extraer información de las demás plataformas en formatos convencionales cómo JSON o XML para uso posterior del usuario.

Interactúa de forma directa con los bloques *Ancient Tortoise* y *Cute Bunny*, que forman parte de la segunda capa de exposición de datos de Ambienta2MX. Se comunica con los demás módulos mediante servicios de tipo REST que funcionan bajo el patrón de convención sobre configuración[?], brindando así una gran compatibilidad con éstos además de disminuir el tiempo de desarrollo debido a que no es necesario generar código único y se opta por la reutilización de éste y bibliotecas que siguen el mismo método de trabajo.

Friendly Dolpin contará con varios procesos y módulos a ser desarrollados. Éste módulo se desarrollará usando tecnologías cómo HTML, Javascript y CSS, además de contar con un ciclo continuo de desarrollo usando herramientas de apoyo cómo Yeoman, Gulp para el maquetado y gestión de tareas comunes en proyectos de tipo web.

Se hará uso del servidor interno que ofrece Gulp junto con las tareas y gestión de bibliotecas de terceros. En cuanto al desarrollo de los estilos necesarios para las vistas se implementará Bootstrap cómo maquetado CSS y finalmente el manejo de vistas, peticiones y lógica dentro del navegador de los clientes se implementará usando EmberJs.

### Diagrama por bloques

A continuación se muestra el diagrama básico de la aplicación:

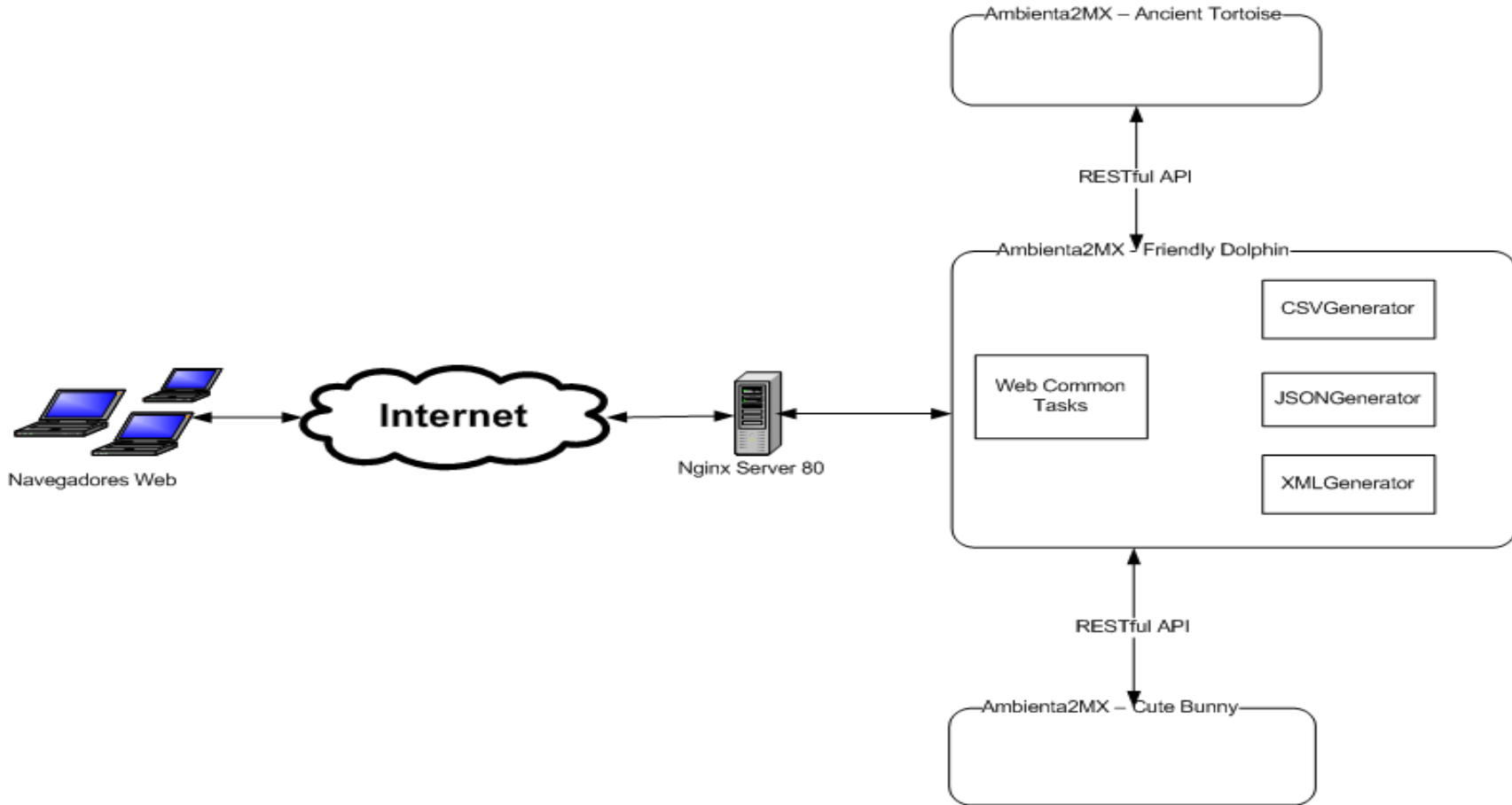


Figura 7.2: Diagrama General de Friendly Dolphin

### 7.3.2. Cute Bunny

#### Definición

El módulo *Cute Bunny* es parte medular para la comunicación con otros sistemas. Cute Bunny es el encargado de proporcionar los respectivos servicios de tipo REST a otros sistemas o módulos que deseen consultar la información almacenada en las respectivas bases de tipo MX.

La información de variables ambientales e índices de contaminación y calidad del aire, será expuesta por medio de peticiones HTTP que siguen un patrón “Conv over Conf”, es decir, sigue la misma estructura que otros sistemas y frameworks de desarrollo web, esto facilitará que aquellos desarrolladores y analistas que deseen acceder de forma programática no tengan que realizar demasiadas configuraciones y se adapten de forma orgánica al sistema.

HTTP Method	URI	Controller Action
GET	/books	index
GET	/books/create	create
POST	/books	save
GET	/books/\${id}	show
GET	/books/\${id}/edit	edit
PUT	/books/\${id}	update
DELETE	/books/\${id}	delete

Figura 7.3: Tabla de peticiones a Cute Bunny

Considerando la naturaleza del sistema, que es primordialmente de consultas y extracción de información de servicios externos, no es necesario contar con una extensa lógica de negocio. El núcleo de la aplicación será la exposición de datos y seguir la convención de servicios REST, considerando algunos ejemplos como los que muestra el siguiente diagrama.

Cute Bunny interactúa con otros dos módulos de Ambienta2MX, Smart Owl y Hard Ant. La interacción con estos surge debido a que Hard Ant es el encargado de gestionar el acceso a las bases de datos de tipo MX y Smart Owl brindará y dará solución a las búsquedas que no se encuentren en las bases de tipo MX, es decir, tratará de encontrar la información que Cute Bunny le solicitó para guardarla en algunas de las bases y posteriormente regresar el resultado al solicitante.

### Diagrama por bloques

A continuación se mostrará el diagrama por bloques que define la estructura de Cute Bunny.

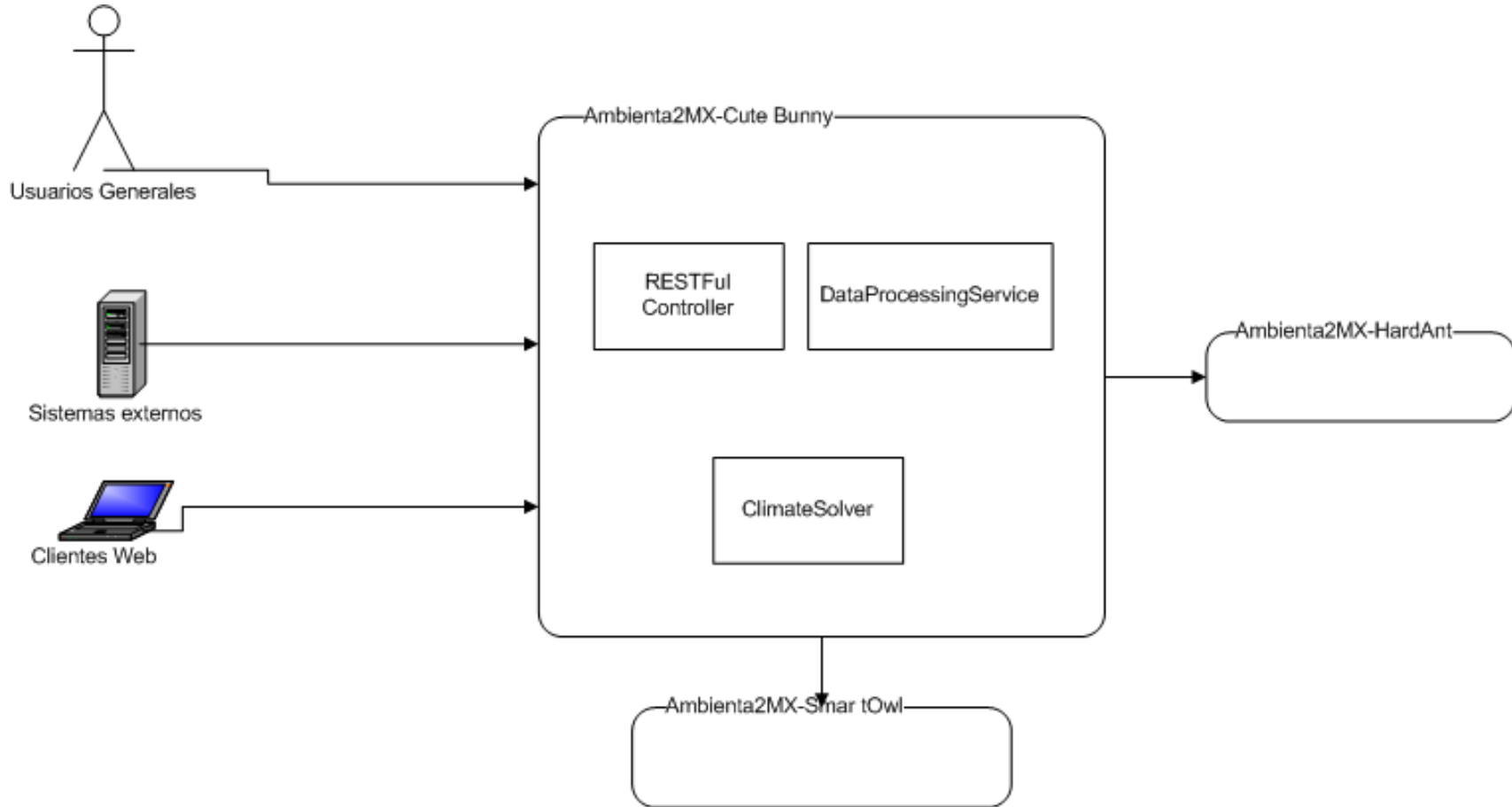


Figura 7.4: Diagrama General de Cute Bunny

### 7.3.3. Ancient Tortoise

#### Definición

El principal objetivo de este módulo es procesar y mostrar la información histórica de las variables almacenadas en las bases de datos de tipo MX. Ancient Tortoise tomará calculará y generará por medio de una regresión lineal una predicción, a lo más de un mes, de las variables ambientales y de contaminación de un estado o localidad en específico. [?]

La información obtenida a demanda por el módulo Smart Owl será la fuente de información básica para generar los modelos de predicción necesarios. Éste no modificará la información existente en la base de datos, sólo obtendrá la información y calculará los datos entre los rangos de fechas definidos por el usuario desde la interfaz generada en Friendly Dolphin.

El usuario final tendrá interacción con estas operaciones ya sea a través de un servicio REST o usando la interfaz web que proporciona Friendly Dolphin. El diagrama por bloques del módulo aísla el módulo de cálculos matemáticos dejando sólo expuesto el servicio para consultas dado una localidad o estado de la República Mexicana considerando como límites una fecha inicial y final. El módulo proveerá la opción de exportar la información generada por éste en los formatos JSON y XML.

Diagrama por bloques.

A continuación se mostrará el diagrama por bloques que define la estructura de Ancient Tortoise.

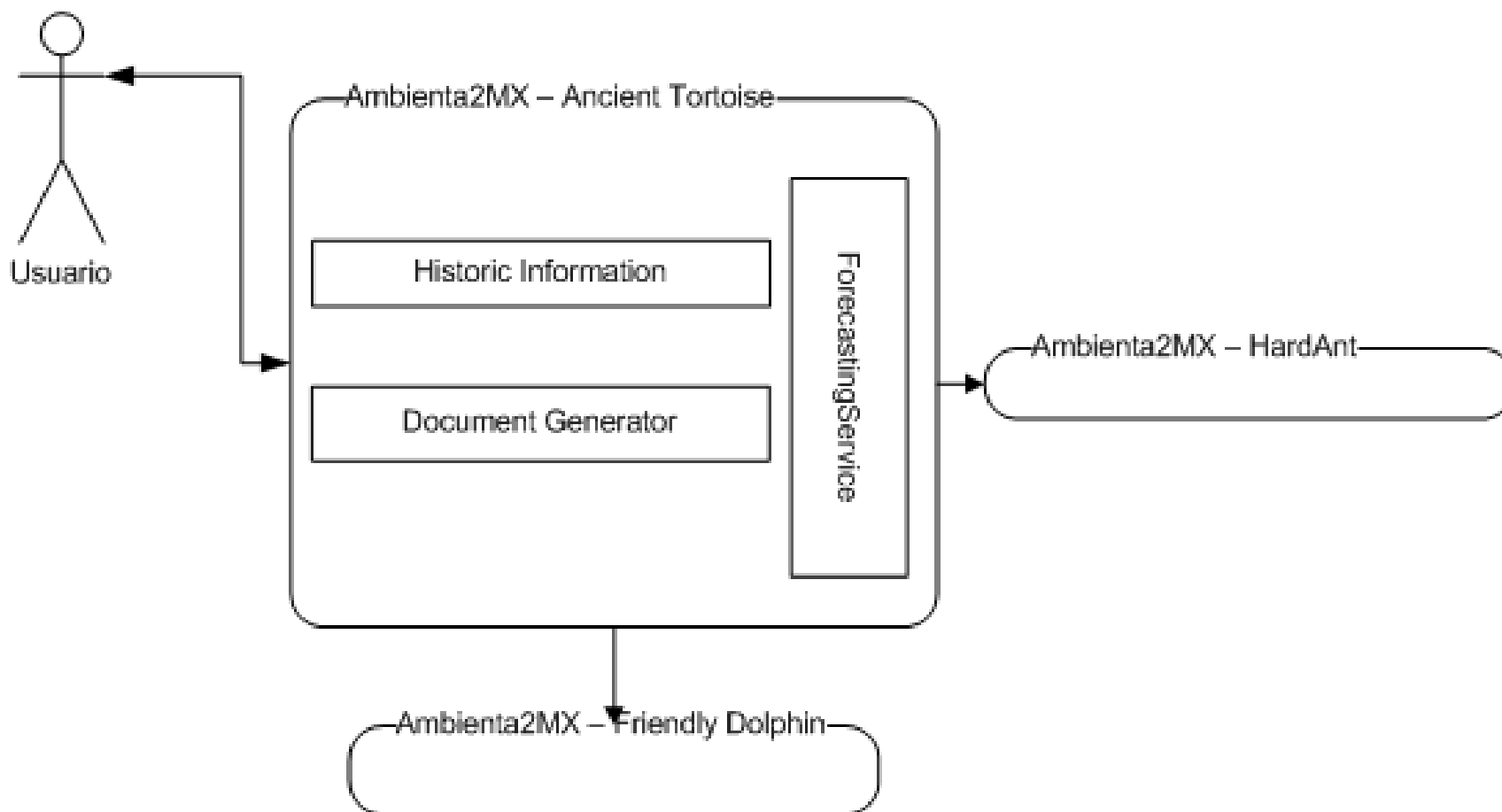


Figura 7.5: Diagrama General de Ancient Tortoise

#### 7.3.4. Smart Owl

##### Definición

La función principal del Smart Owl es la obtención de la información de diversas fuentes, tanto estáticas y dinámicas.

Toda la información que se encontrará en las bases de datos de tipo MX será obtenida a través de Smart Owl, muchas de las fuentes no cuentan con los datos climatológicos completos, principalmente las gubernamentales como: CONAGUA, INEGI o SMN, por citar algunas.

Considerando esa problemática, Smart Owl buscará y tratará de resolver la información de los campos faltantes tomando como base distintas fuentes de datos, algunas establecidas y otras de tipo gubernamental.

También se tomará información de fuentes que tipo dinámica, es decir, cuya información suele ser actualizada en entre periodos de una o dos horas. Estas fuentes suelen contar con RESTful API's para consumo de forma programática.

El modelo de datos final podrá ser entonces persistido después de que haya sido resuelto completa o parcialmente. Se llevará el control de los metadatos considerando su origen, su fecha y algunos tags relacionados los que proveen la información.



Diagrama por bloques.

A continuación se mostrará el diagrama por bloques que define la estructura de Smart Owl.

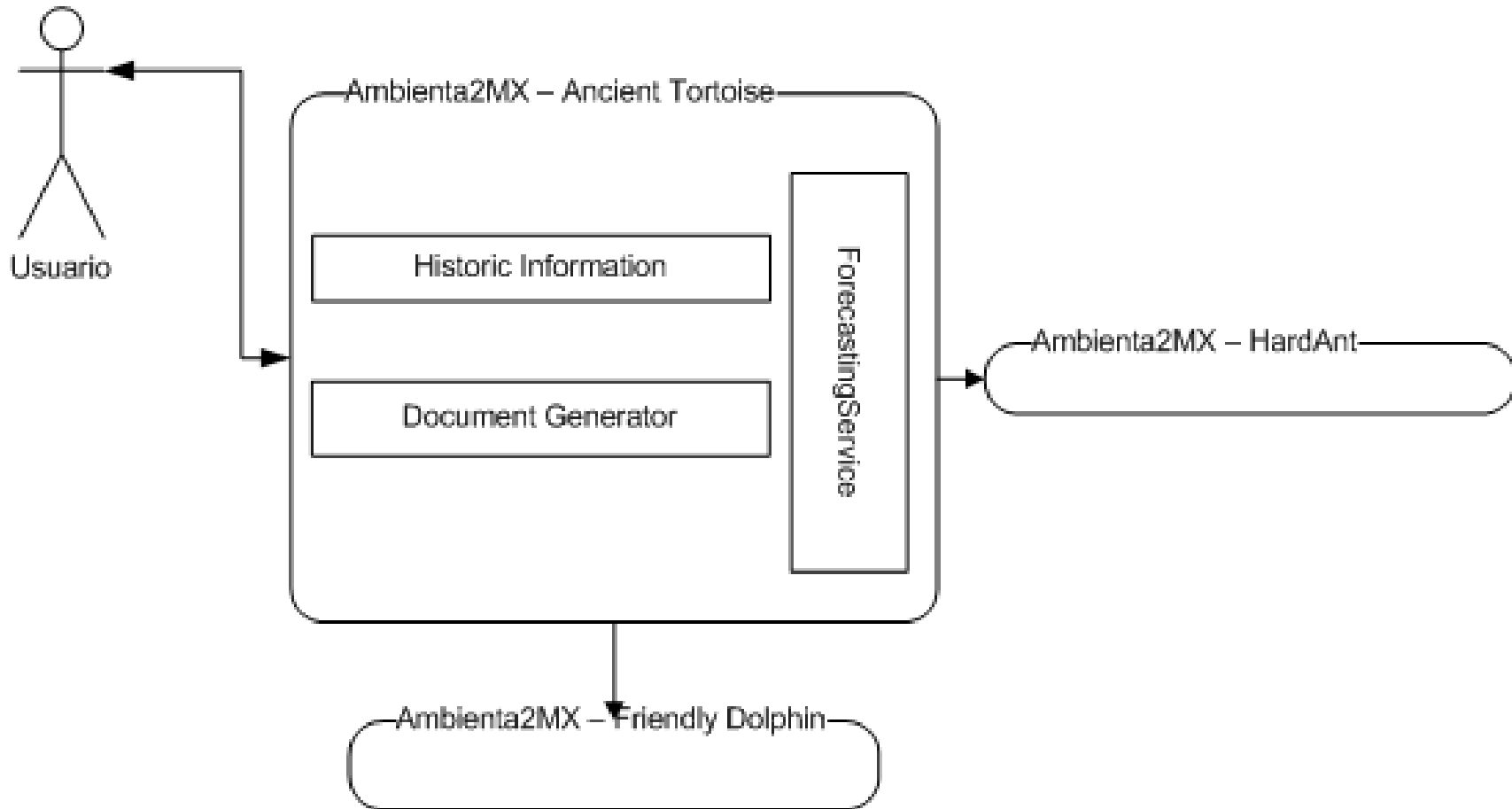


Figura 7.6: Diagrama General de Smart Owl

A continuación se muestran los diagramas de secuencia planteados para el funcionamiento del módulo mencionado.

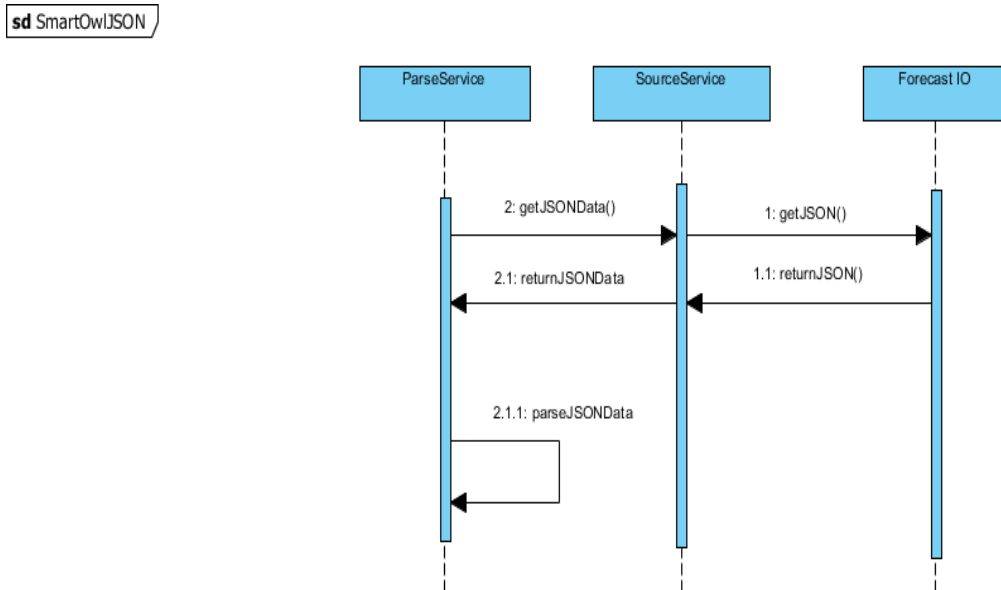
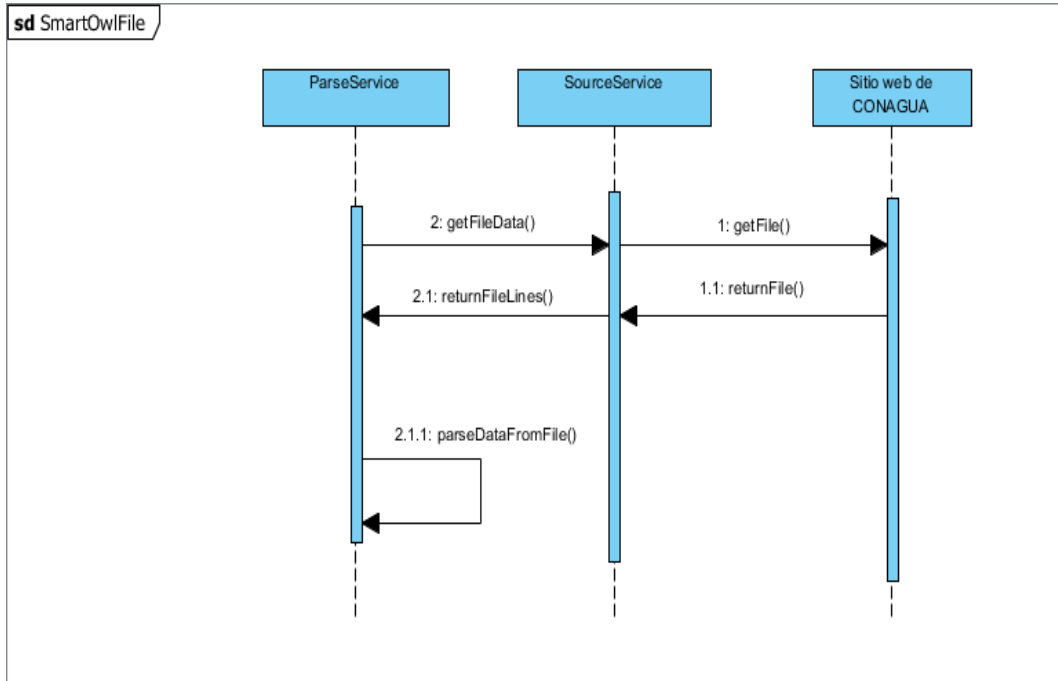


Diagrama por bloques

### 7.3.5. Fast Eagle

#### Definición

El módulo Fast Eagle formará parte de la arquitectura final de Ambienta2MX. El propósito principal de este módulo es brindar la información cartográfica de México por medio de un servicio expuesto, considerando latitud, longitud o nombre de la localidad deseada.

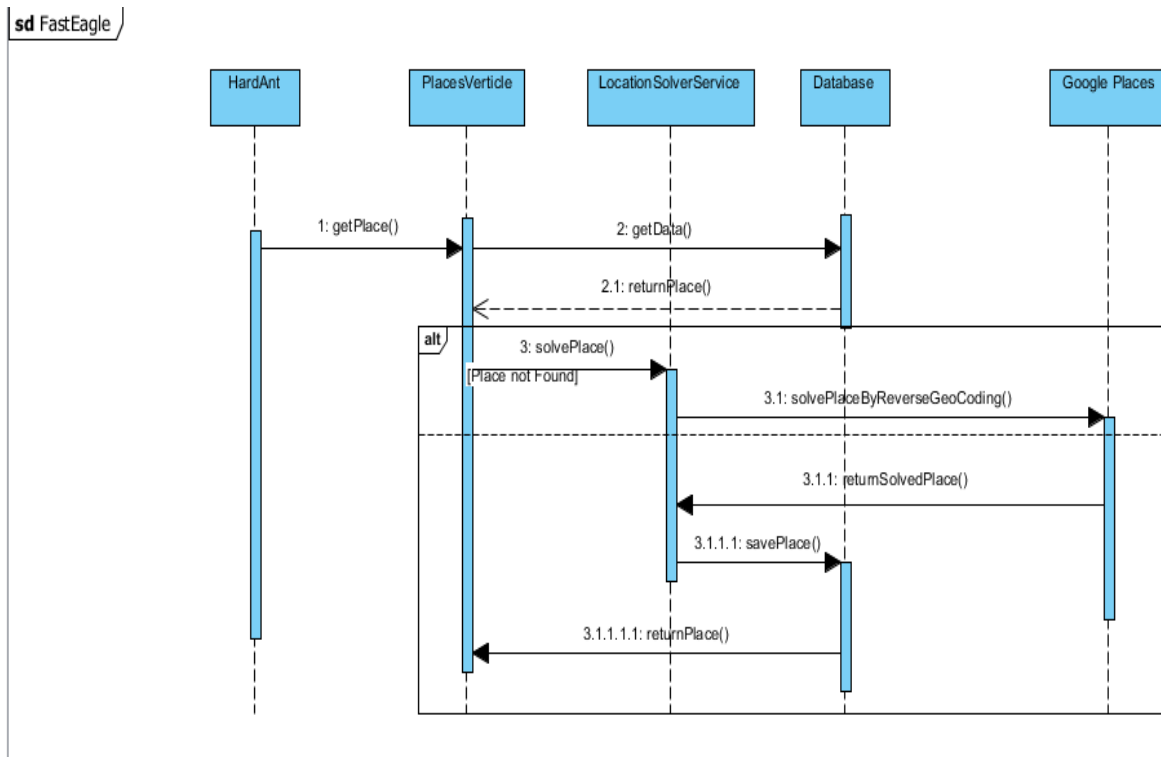
Fast Eagle se encargará de la lectura, carga, verificación y resolución de los archivos brindados por el (Instituto Nacional de Estadística y Geografía) INEGI en la información pública que tiene de la cartografía del territorio nacional.

El proceso de toma de datos del INEGI será ejecutado sólo una vez al inicio o primer despliegue del sistema. Esto debido a que la cartografía no deberá ser procesada de forma dinámica. Se usará un modelo de datos persistido Mongo para sus posteriores consultas.

La información que proporciona el INEGI carece de campos esenciales para la estandarización de los información cartográfica, para dar solución a ese contra-tiempo se hará uso de servicios externos que ya cuentan con información definida, es decir, que su información ha pasado bajo un cierto proceso de limpieza y regulación, por ejemplo, los servicios de Google Places API.

En caso de que no exista la información deseada por el usuario o algún otro módulo del sistema en la cartografía (base de datos llamada Places), Fast Eagle tratará de resolver la información en fuentes externas, persistiendo el modelo resuelto y regresando esa información al solicitante.

## Diagrama de secuencias



## Diagrama por bloques

Fast Eagle contará con varios procesos a ser desarrollados, la integración de cada proceso y su respectiva integración dará solución a un problema de estandarización, resolución y consulta de datos geográficos vía Latitud, Longitud y Ubicación.

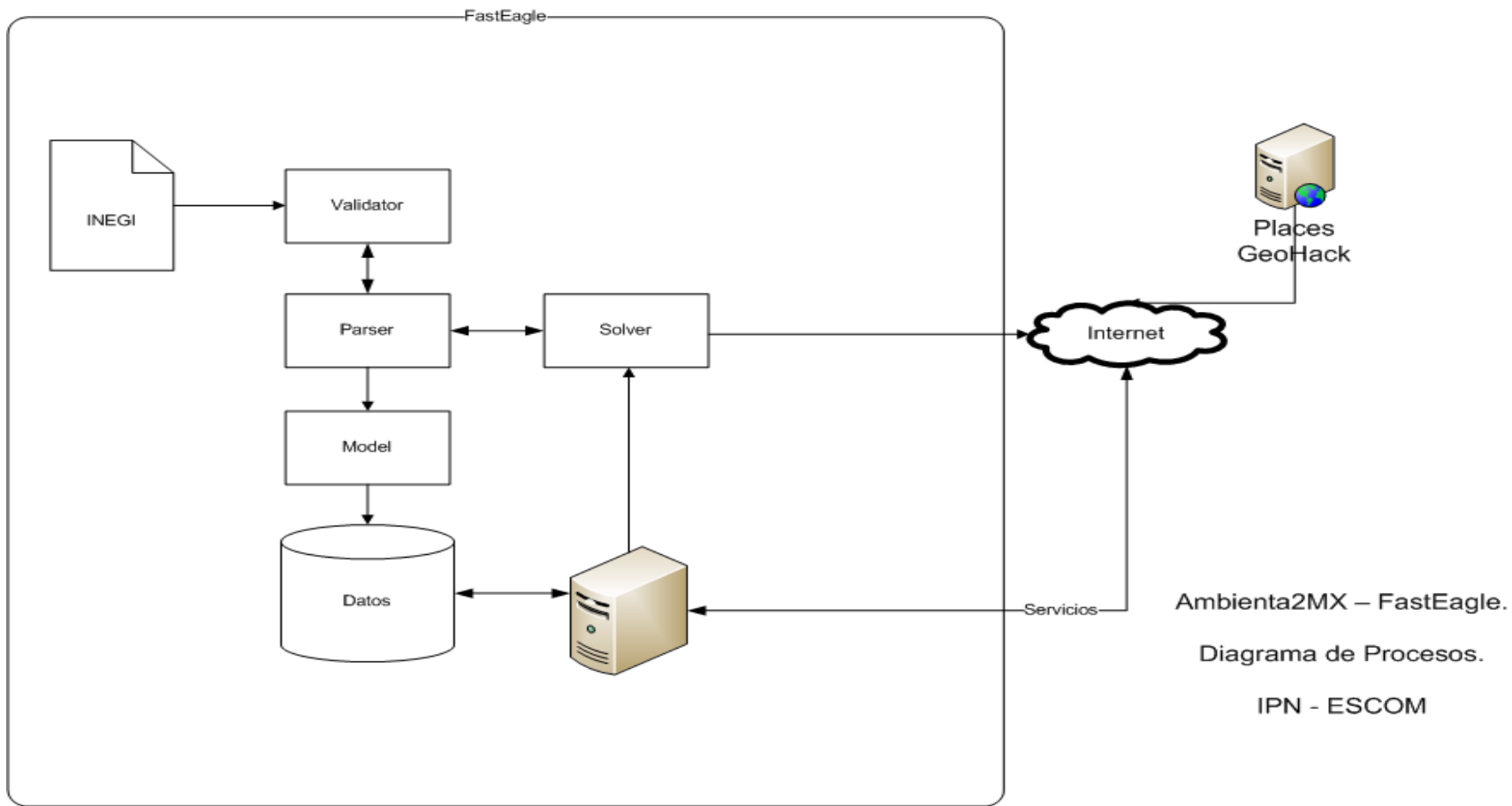


Figura 7.7: Diagrama por bloques de Fast Eagle

En el diagrama se muestran cuatro módulos básicos, estos forman parte del núcleo de Fast Eagle, también podemos observar que se cuenta con la interacción de servicios de terceros como Google Places, también se cuenta con la exposición de los servicios a través de un servidor web.

El módulo de Validación *Validator*, será el encargado de tomar las fuentes que el INEGI brinda al público en general en forma de archivos CSV, y realizar un proceso de validación a los datos que éstos tienen.

El módulo indicará que datos necesitan una resolución y cuáles pueden ser estandarizados y posteriormente almacenados en la base de datos.

*Parser* tomará los datos que el proceso de validación le arroje para transformar al estado propuesto por el equipo de trabajo (Véase modelo de datos). Considerando un proceso de resolución en caso de que la información proporcionada por el INEGI se encuentre incompleta no sea válida.

Para toda la información que carezca de datos correctos *Solver* buscará una resolución en servicios de terceros, después de la resolución, los datos serán guardados en el gestor de bases de datos bajo el formato propuesto por el equipo de trabajo.

*Model* es la capa de interacción con la base de datos, ésta se encarga de las operaciones mejor conocidas como CRUD (Create, Read, Update and Delete), persistiendo la información en MongoDB.

Para poder exponer los datos, se hará uso de un servidor web minimalista orientado a micro servicios, éste será un servicio público que formará parte de la infraestructura final de Ambienta2MX.

El servicio expuesto se encargará de las búsquedas a nivel base de datos y en caso de no encontrar la información buscará en terceros para poder agregarla a la base de datos y así ir mejorando el contenido de nuestro índice cartográfico.

### 7.3.6. Hard Ant

#### Definición

Hard Ant es uno de los bloques funcionales base de Ambienta2MX, su función principal es la de enrutar las peticiones a las bases de tipo MX además de brindar la solución cartográfica (a nivel ubicación) interactuando con Fast Eagle.

Como complemento a la arquitectura, también contará con el proceso del registro masivo de información, exponiendo servicios que el módulo Smart Owl usará de forma constante para persistir la información estandarizada de diversas fuentes de información.

Hard Ant brindará los canales de acceso a las bases MX definidas en el diagrama general mediante servicios HTTP, estos servicios brindarán información para el proceso de inserción y extracción de la información.

Este módulo es lo que sería considerado la capa del modelo de datos en un patrón MVC, ya que es la que tiene contacto de forma directa con los datos almacenados en las bases de datos orientadas a documentos gestionadas por Mongo.

Se utilizará un pool de conexiones a la base para garantizar el acceso o escritura a los datos además de brindar la posibilidad de respuestas asincronas y no bloqueantes entre las consultas realizadas.

Considerando trabajos más pesados (Obtención de datos históricos) se utilizarán procesos en segundo plano, esto es posible gracias a la implementación de “Verticles” nativas de Vert.x, tecnología que será usada para el desarrollo y despliegue final de éste módulo.

Diagrama por bloques

A continuación se mostrará el diagrama por bloques que define la estructura de Hard Ant.

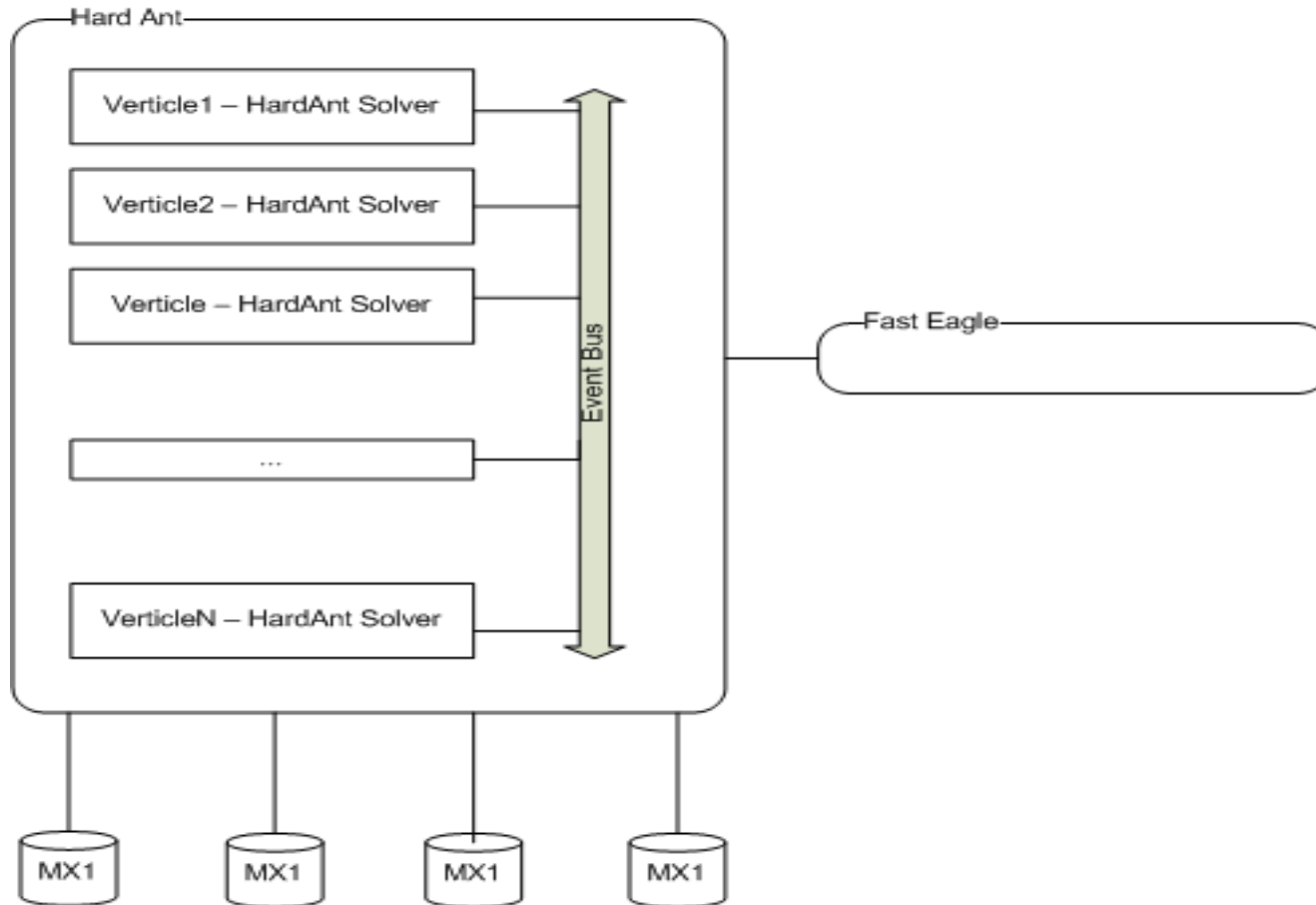


Figura 7.8: Diagrama General de Hard Ant



En el diagrama se puede apreciar la replicación de “Verticles” que interactúan a través del mismo canal de información, esto brinda una alta disponibilidad del servicio ya que las peticiones son atendidas y procesadas no sólo por un elemento existente si no varios. Otros módulos de Ambienta2MX pueden conectarse a este canal de información ya sea de forma local o distribuida.

### 7.3.7. Otros.

#### Módulo de WebScrapping - Wicked Fox

Este módulo será el encargado de tomar información de plataformas que carecen de un servicio web definido, por ejemplo el Sistema Meteorológico Nacional. Toda la información que exponen de forma diaria, semana, mensual y anual. Se encuentra bajo un formato HTML que puede ser visualizado a través de algún navegador web al acceder a su sitio.

Problemas como los anteriores suelen ser comunes a nivel nacional (principalmente plataformas gubernamentales), para poder hacer uso de esos recursos es necesario generar un módulo que de forma programática pueda extraer limpiar, extraer y adecuar la información.

Al proceso antes descrito se le conoce como *Web Scrapping*?, que se define como la extracción de información de sitios web a través de medios programáticos o programas definidos.

Este módulo se encuentra implementado bajo un script escrito en Javascript usando PhantomJs como medio lógico y programático para la implementación del Scrapping.

La definición básica de *Wicked Fox* es tomar el recurso que brinda el SMN, extraer la información contenida en tablas y posteriormente mandar esta información al módulo *Smart Owl* que se encargará de adecuar y después delegar el trabajo al módulo de registro *Hard Ant* donde al final la información climatológica de cierta localidad será guardada en alguna de las bases de datos antes mencionadas.

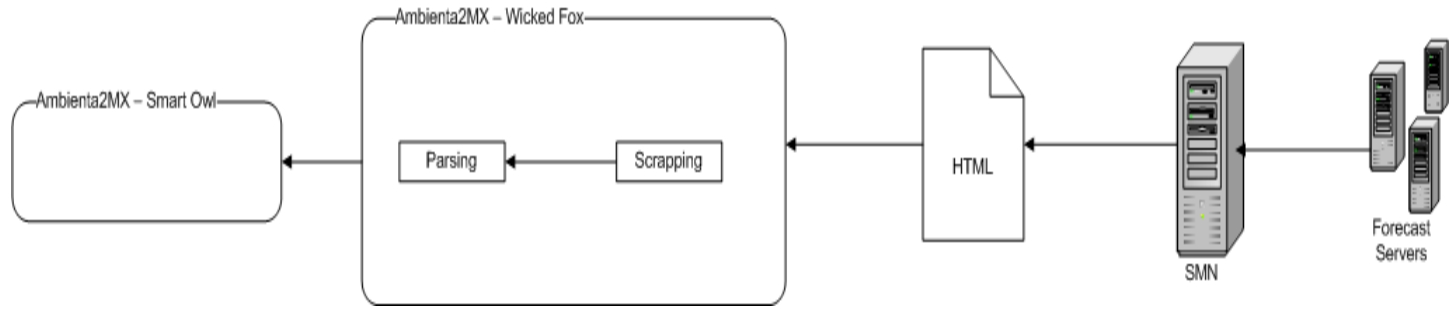


Figura 7.9: Diagrama General de Wicked Fox

Éste módulo presenta una gran importancia debido a que actualmente se carece de un servicio a nivel nacional que brinde información meteorológica con cierta veracidad, si bien es cierto que otros sistemas extraen información de datos tomados por centrales mexicanas, el costo de éstos suele ser elevado.

La extracción de la información se ejecutará *al vuelo*, es decir, éste proceso se ejecutará cada que se pidan datos con un margen no mayor a una hora de cierta localidad o ubicación del territorio nacional.

Para poder hacer uso de éste módulo es necesario brindar la información de la localidad via latitud/longitud o bien, a través de la localidad descrita, éstos datos serán proporcionados y resueltos por el módulo *Fast Eagle* descrito anteriormente.

# Capítulo 8

## Análisis y gestión de riesgos

### 8.1. Definición y clasificación

Los objetivos de la gestión de riesgos son identificar, controlar y eliminar las fuentes de riesgo antes de que empiecen a afectar el cumplimiento de los objetivos del proyecto.

El riesgo siempre implica una incertidumbre y una pérdida potencial, es necesario llevar una cuantificación de estos parametros, por lo que suelen ser clasificados en diferentes categorías por ejemplo:

- Riesgos del proyecto
- Riesgos técnicos
- Riesgos de negocio

Es necesario llevar un proceso de administración de riesgos y planes de contingencia para poder controlar esos inesperados eventos. Primero se tienen que identificar los riesgos, considerando principalmente los más potenciales, posteriormente analizarlos y dandoles una prioridad, generar planes de contingencia para finalmente supervizarlos y actuar conforme a lo acordado.

Los riesgos pueden ser clasificados respecto a estudios previos o con relación a la experiencia de trabajo que tengan los desarrolladores y analistas del proyecto, sin embargo, generalmente suelen asignarseles ciertas ponderaciones por convención, como se muestra a continuación:

- Muy bajo ( <10 % )
- Bajo ( 10 - 25 % )
- Moderado (25 - 50 % )
- Alto (50 - 75 % )
- Muy Alto ( >75 % )

Todos los riesgos deben encontrarse también clasificados entre cualquiera de las cuatro valoraciones: Insignificante, Tolerable, Serio, Catastrófico; los planes de contingencia suelen ser desarrollados para aquellos riesgos con probabilidad de moderada a muy alta considerando, tomando en cuenta un impacto serio o catastrófico.

A lo largo del desarrollo de Ambienta2MX surgieron y surgirán eventos y sucesos inesperados que serán considerados como riesgos potenciales dependiendo el impacto que estos lleguen a tener.

Dentro del proceso de planeación del proyecto se decidió delegar o depender de los servicios de terceros en cuanto a infraestructura física, esto implica que se disminuyen riesgos de instalación o mantenimiento a equipos de cómputo o redes computacionales.

Sin embargo, esto nos deja a merced de nuestros proveedores de servicios. La siguiente tabla muestra los riesgos más significativos considerados a lo largo del desarrollo del proyecto y sus respectivos planes de acción.

Tabla de Riesgos				
Descripción	Tipo de Riesgo	Valoración	Porcentaje	Plan de acción
No disponibilidad de los servidores de Amazon	Técnico	Catastrófico	1 %	Migración parcial a servicios de hosting gratuitos como Heroku u Openshift.
Falta de presupuesto	Proyecto	Serio	25 %	Buscar un proceso de incubación en empresas como Apache, Eclipse y migrar la plataforma a servicios de hosting gratuitos como Heroku u Openshift.
Falta por razones personales de miembros del equipo	Proyecto	Serio	25 %	Rediseñar y adaptar las tareas con nuevos integrantes y habilitar forma de trabajo remota considerando fines de semana.
Extinción de recursos de información	Proyecto, Técnico	Catastrófico	10 %	Cambiar las reglas de negocio y buscar nuevas fuentes de información como SaaS y desplegar los módulos de estandarización nuevamente.
Necesidad de escalabilidad de la plataforma	Técnico	Serio	10 %	Definir el tipo de escalabilidad a usar dependiendo los recursos monetarios existentes.
Ataques de Denegación de Servicios (DoS)	Técnico	Tolerable	10 %	Cambiar el tipo de dirección en los servidores de Amazon y agregar caché al sistema.
Adaptación a nuevos estándares	Técnico	Tolerable	15 %	Rediseñar y extender la funcionalidad de la aplicación, dando soporte a ambos modelos de intercambio de datos.

Cuadro 8.1: Analisis de Riesgos

# Pruebas unitarias

## Smart Owl

### Package mx.ipn.ambienta2mx.smartOwl

all > mx.ipn.ambienta2mx.smartOwl

11 tests	0 failures	0 ignored	0.402s duration	<b>100%</b> successful
-------------	---------------	--------------	--------------------	---------------------------

#### Classes

Class	Tests	Failures	Ignored	Duration	Success rate
<a href="#">ParseServiceSpec</a>	4	0	0	0s	100%
<a href="#">SourceServiceSpec</a>	7	0	0	0.402s	100%

### Class mx.ipn.ambienta2mx.smartOwl.ParseServiceSpec

all > mx.ipn.ambienta2mx.smartOwl > ParseServiceSpec

4 tests	0 failures	0 ignored	0s duration	<b>100%</b> successful
------------	---------------	--------------	----------------	---------------------------

#### Tests

Test	Duration	Result
Should get the longitude 19°18'36" and altitude 2348 given the url: http://smn.cna.gob.mx/emas/txt/DF15_10M.TXT	0s	passed
Should get the longitude 99°09'29" and altitude 2281 given the url: http://smn.cna.gob.mx/emas/txt/DF09_10M.TXT	0s	passed
Should get the longitude 99°09'44" and altitude 2946 given the url: http://smn.cna.gob.mx/emas/txt/DF08_10M.TXT	0s	passed
Should get the longitude 99°12'14" and altitude 2200 given the url: http://smn.cna.gob.mx/emas/txt/DF06_10M.TXT	0s	passed

### Class mx.ipn.ambienta2mx.smartOwl.SourceServiceSpec

all > mx.ipn.ambienta2mx.smartOwl > SourceServiceSpec

7 tests	0 failures	0 ignored	0.402s duration	<b>100%</b> successful
------------	---------------	--------------	--------------------	---------------------------

#### Tests

Test	Duration	Result
Should return the JSON when https://api.forecast.io/forecast/ is consulted given the latitude 19.029 and longitude -98.206	0s	passed
Should return the JSON when https://api.forecast.io/forecast/ is consulted given the latitude 19.395 and longitude -99.025	0s	passed
Should return the JSON when https://api.forecast.io/forecast/ is consulted given the latitude 23.543 and longitude -103.025	0.001s	passed
Should return the file lines when http://smn.cna.gob.mx/emas/txt/DF06_10M.TXT is consulted	0.072s	passed
Should return the file lines when http://smn.cna.gob.mx/emas/txt/DF08_10M.TXT is consulted	0.060s	passed
Should return the file lines when http://smn.cna.gob.mx/emas/txt/DF09_10M.TXT is consulted	0.062s	passed
Should return the file lines when http://smn.cna.gob.mx/emas/txt/DF15_10M.TXT is consulted	0.207s	passed

# Historias de Usuario

## Friendly Dolphin

- US1 Reporte de información climática actual
  - **Como** usuario de Friendly Dolphin
  - **Quiero** consultar la información climática actual
  - **De tal manera** que pueda generar un reporte con la información estandarizada.
- Criterios de Aceptación
  - Seleccionar una región del país.
  - Se deberá mostrar la información en texto plano.
  - Opción para exportar la información en formato JSON o XML.
- US2 Historial de información climática
  - **Como** usuario de Friendly Dolphin
  - **Quiero** consultar el historial de los datos climáticos.
  - **De tal manera** que pueda visualizar de manera gráfica el cambio de los valores de las variables climáticas a través de un periodo de tiempo
- Criterios de Aceptación
  - Consultar la información entre una fecha de Inicio y una fecha Final.
  - Se deberá mostrar la información de las variables climáticas en el periodo seleccionado.
  - Opción para exportar la información en formato JSON o XML.



## Servicios REST

La ventaja al usar servicios de tipo REST, es la sencillez al cambiar el contenido que se expone sin tener que cambiar o generar un protocolo de comunicación ya que toma como base HTTP .

Simplemente es necesario con un cliente, desde un navegador web hasta clientes dedicados a servicios REST, para poder acceder a los recursos expuestos. Todas las API's rest siguen una convención de verbos tomados de la base de HTTP (GET, POST, PUT, DELETE).

En comparación con servicios de tipo SOAP, no se requiere una alta atomicidad y ni transacciones, una de las razones por la que los servicios WS suelen ser usados.

Finalmente, los servicios de tipo REST pueden tener respuestas en diversos formatos, principalmente JSON y XML, esto brinda al desarrollador o a la persona que consula como tratar la respuesta por bibliotecas de terceros o nativas.

El soporte para formatos JSON es nativo en los navegadores web, otra de las razones por las cuales éstos servicios han ido ganando mercado ya que el desarrollo de aplicaciones Web ha aumentado de forma drástica en los últimos años.  
[?]

## Glosario

- **JSON:** Javascript Object Notation, Es un formato ligero de intercambio de datos.
- **ITRF:** Acrónimo de International Terrestrial Reference System (Marcos de Referencia Terrestre Internacional).
- **NAD27:** Acrónimo de North American Datum of 1927. Marco de referencia terrestre usado por el INEGI hasta el año 1998.
- **GIS:** Sistema de Información Geográfica.
- **API:** Application Programming Interface, conjunto de rutinas, protocolos o herramientas para construcción de software.
- **REST:** Representational State Transfer, Arquitectura de software para sistemas web basada en un patrón “Convention over configuration” con operaciones que funciona usando el protocolo HTTP como base.
- **SMN:** Servicio Meteorológico Nacional.
- **CONAGUA:** Comisión Nacional del Agua.
- **INEGI:** Instituto Nacional de Estadística y Geografía.
- **Políglota:** En área informática, que tiene soporte o comprende varios lenguajes de programación.
- **StakeHolder:** Personas u organizaciones que participan o tienen un interés en un proceso específico o definido dentro de un empresa. Se convierten en fuente clave de información para desarrollar un sistema o mejorar algún proceso.
- **SaaS:** Acrónimo de Software as a Service. Es un modelo de distribución de software donde el soporte lógico y los datos que maneja se alojan en servidores de una compañía de tecnologías de información y comunicación (TIC).

# Bibliografía

- [1] Mapa Digital de México, INEGI 2014, Available at: <http://gaia.inegi.org.mx/mdm6/>
- [2] Modelo de datos CIM, 2013, Available at: <https://earthsystemcog.org/projects/es-doc-models/cim>
- [3] GIS México - Sistemas de Información Geográfica, Available at: <http://www.sigmexico.com.mx/>
- [4] GEOJson Format Specification, 2008, Available at: <http://geojson.org/geojson-spec.html>
- [5] GIS (Geographic information system), 2010, Available at: <http://education.nationalgeographic.com/education/encyclopedia/geographic-information-system-gis>
- [6] INEGI, Acerca del INEGI, N/A, Available at: <http://www.inegi.org.mx/inegi/acercade/default.aspx>
- [7] Web Scrapping Introduction, Geoff Boeing, Berkeley University, 2014, Available At: [http://dlab.berkeley.edu/sites/default/files/training\\_materials/web-scrapping-talk.pdf](http://dlab.berkeley.edu/sites/default/files/training_materials/web-scrapping-talk.pdf)
- [8] Convention over configuration, Microsoft Developers Network, 2012, Available At: <https://msdn.microsoft.com/en-us/magazine/dd419655.aspx>.
- [9] INSPIRE Project, Aboute INSPIRE, Available at: <http://inspire.ec.europa.eu/>
- [10] NoSQL Explained, MongoDB Courseware, Available at: <https://www.mongodb.com/nosql-explained>
- [11] Seven databases in seven weeks: A guide to Modern Databases and NoSQL movement, Eric Redmond, Pragmatic Bookself, 2013.
- [12] MongoDB Case Study: Foursquare, Mongodb Courseware, Available at: <http://www.mongodb.com/post/15400944604/mongodb-case-study-foursquare>
- [13] Weather Underground, About the Data, Available at: <http://www.wunderground.com/about/data.asp>
- [14] Forecast.io, Data Sources, Available at: <http://forecast.io/raw/>
- [15] Servicio Meteorológico Nacional, Conceptos, Available at: [http://smn.cna.gob.mx/index.php?option=com\\_content&view=article&id=23&Itemid=120](http://smn.cna.gob.mx/index.php?option=com_content&view=article&id=23&Itemid=120)

- [16] The JSON Data Interchange Format, ECMA International, 2013, Available at: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
- [17] Base de datos Estadísticas BADESNIARN, Semarnat, Available at: <http://www.semarnat.gob.mx/temas/estadisticas-ambientales/badesniar?De=SNIARN>
- [18] Geographic System Information, Geographic Information Centre (Canada), Available at: <http://gic.geog.mcgill.ca/gis-resources/>
- [19] Geoplatform Overview, Geoplatform.gov, 2015, Available at: <https://www.geoplatform.gov/overview-page>
- [20] Common Core Metadata Schema, OpenData.gov, 2013, Available at: <https://project-open-data.cio.gov/schema/>
- [21] Open Government Initiative, The White House, 2009, Available at: <https://www.whitehouse.gov/open>
- [22] Agile Methodology, 2015, Available at: <http://agilemethodology.org/>
- [23] The Scrum Reference Card, Michael James, 2015, Available at: <http://scrumreferencecard.com/reference-card-de-scrum/>
- [24] Agile Methodologies for Software Development, Mike McLaughlin, Available at <http://www.versionone.com/agile-101/agile-development-methodologies-scrum-kanban-lean-xp/>
- [25] Guidelines for Test-Driven Development, Jeffrey Palermo, 2006, Available at [https://msdn.microsoft.com/en-us/library/aa730844\(v=vs.80\).aspx](https://msdn.microsoft.com/en-us/library/aa730844(v=vs.80).aspx)
- [26] Refactoring, Martin Fowler, Available at <http://refactoring.com/>
- [27] Continuous Integration, Martin Fowler, Available at <http://www.thoughtworks.com/continuous-integration>
- [28] Refactoring Improving the design of existing code, Martin Fowler, Pragmatic Bookshelf, 1999
- [29] Prototyping Model, Margaret Rouse, N/A, Available at <http://searchcio.techtarget.com/definition/Prototyping-Model>
- [30] Algorithms Make Better Predictions — Except When They Don't, Harvard Business Review, Thomas C.Redman, 2014, Available at <https://hbr.org/2014/09/algorithms-make-better-predictions-except-when-they-dont>
- [31] REST vs SOAP The difference between Soap and REST, Hacking Management, 2010, Available at: <http://spf13.com/post/soap-vs-rest>