

**ESTIMACIÓN NO PARAMÉTRICA DE DENSIDAD DE
PROBABILIDAD PARA EL ESTUDIO DE DESCARGAS
ATMOSFÉRICAS**

TESIS

Que para obtener el grado de
MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA

presenta

Luis Cuauhtemoc García Puente

Dr. Félix Calderón Solorio

Director de Tesis

Universidad Michoacana de San Nicolás de Hidalgo

Agosto 2011

A mi familia, porque siempre han confiado en mi. A mi esposa y mi hija por darme fuerza para seguir adelante. A mis profesores y a mis grandes amigos por su apoyo incondicional

Lista de Publicaciones

“Estimación de la Función de Densidad de Probabilidad para el estudio de Descargas Atmosféricas”

Autores Calderón Solorio Félix, García Puente Luis Cuauhtemoc, Avalos Gonzales Jose Alberto.

Publicado en XII Reunión de Otoño de Potencia, Electrónica y Computación ROPEC 2010 Internacional

“A Hybrid Parellel Processing Methodology for solving Large Scale Problems using PVM and SSH”

Autores Roberto Olmos, Sergio Dominguez, Luis Garcia, Rafael Lopez, Abel Tapia, Norberto Garcia.

Publicado en 1st International Supercomputing Conference in México ISUM 2010

Resumen

En este trabajo se presenta una estimación no paramétrica de la función de densidad de probabilidad (FDP) para el estudio de Descargas Atmosféricas (DA). Las DA son detectadas con una antena Boltek LD-250, la cual captura la distancia y ángulo al que ocurre la DA. A diferencia de una representación puntual de las DA, se propone generar la imagen de la FDP con el objetivo de proporcionar a la Comisión Federal de Electricidad un análisis probabilístico más completo de las DA detectadas sobre la región geográfica que comprende la División Centro Occidente. Con esta información se pueden tomar decisiones de valoración de riesgos referentes a la instalación de nuevas subestaciones o de equipo de transmisión, tales como líneas de alto voltaje, transformadores, bancos de reactores o capacitores, etc. Con los mapas geográficos de la FDP se tendría una referencia gráfica de las zonas de mayor incidencia de DA.

Durante el desarrollo de este trabajo abordamos la problemática de graficar puntos geográficos sobre mapas en dos dimensiones, con la finalidad de tener una referencia real de la localización de las DA y generar en dichas posiciones, gráficas de la FDP. Se planteó utilizar geometría plana, descubriendo el error de asumir que esta es la forma de la Tierra y por tal motivo fue necesario investigar con más detalle acerca de la georeferenciación de puntos sobre el planeta. Se implementaron algoritmos diseñados por Theodore Vincenty en la década de los 70's para la correcta localización de las DA obteniendo así, una representación más real de nuestras DA.

Gracias a la documentación proporcionada por compañías como Google o Microsoft, pudimos entender la manera en que los mapas desarrollados por estas empresas despliegan puntos georeferenciados sobre sus imágenes, se implementaron ecuaciones bien definidas en la documentación para graficar puntos geográficos sobre imágenes referenciadas a coordenadas en píxeles. Los algoritmos desarrollados se implementaron en JAVA utilizando mapas estáticos obtenidos mediante consultas HTTP, además se implementó el uso de mapas dinámicos mediante una aplicación web la cual puede ser accesada a través de internet. Esta aplicación nos da la posibilidad de realizar las funciones de mayor demanda sobre mapas tales como acercamientos, desplazamiento dinámico sobre el mapa, utilización de diferentes tipos de mapas y despliegue de información referente a la geografía y división política de la región.

Contenido

Dedicatoria	III
Lista de Publicaciones	V
Resumen	VII
Abstract	IX
Contenido	XI
Lista de Figuras	XIII
Lista de Tablas	XV
Lista de Algoritmos	XVII
Lista de Símbolos	XIX
1. Introducción	1
1.1. Planteamiento del problema	1
1.2. Antecedentes	3
1.3. Descripción de capítulos	6
2. Detección de descargas atmosféricas y respaldo de la información	9
2.1. Conclusiones	14
3. Uso de mapas estáticos para la graficación de los resultados	15
3.1. Despliegue de descargas atmosféricas en mapas estáticos	19
3.2. Conclusiones	24
4. Despliegue de funciones de densidad de probabilidad	25
4.1. Funciones de densidad de probabilidad	25
4.2. Ventanas de Parzen	26
4.3. Imagen de la FDP	29
4.4. Conclusiones	35
5. Corrección de los métodos de graficación aplicados	37
5.1. Errores de precisión	37
5.2. Cálculo de geodésicas	39
5.3. Georeferencia de eventos atmosféricos	40
5.4. Mapas de Google	42
5.5. Corrección de errores de graficación de DA y FDP	44
5.6. Conclusiones	46

6. Pruebas y resultados	49
6.1. Conclusiones	50
7. Conclusiones	53
7.1. Conclusiones generales	53
7.2. Trabajos futuros	54
A. Algoritmos de cálculo geodésico	55
A.1. Fórmula Directa de Vincenty	58
A.2. Fórmula Inversa de Vincenty	59
B. APACHE TOMCAT sobre WINDOWS	63
C. Diseño de Aplicación WEB para desplegar mapas	67
C.1. Java Server Pages (JSP)	68
C.2. Desarrollo de aplicación WEB utilizando NETBEANS	70
C.3. Empaquetado de la aplicación WEB	78
Referencias	81

Lista de Figuras

1.1. Antena, guarda y tarjeta de adquisición de datos BOLTEK LD-250	3
1.2. Ubicación de la antena y eje de referencia	4
1.3. Mapa de descargas atmosféricas	4
2.1. Criterio utilizado para el agrupamiento de DA.	11
2.2. Definición de la base de datos	11
3.1. Tipos de Mapas utilizados en el proyecto	17
3.2. Imagen resultante de una consulta HTTP	18
3.3. Descargas Atmosféricas detectadas entre el 01/01/2010 y 01/03/2010	22
3.4. Datos Puntuales y Estimación de la Función de Densidad de Probabilidad .	23
4.1. Histograma resultante del ejemplo numérico propuesto.	28
4.2. Implementación de ventanas de Parzen cuadradas con $h = 1$	28
4.3. Implementación de ventanas de Parzen Gaussianas con $h = 1$	29
4.4. Cálculo de la distribución de probabilidad utilizando ventanas de Parzen de tamaño 10	32
4.5. Comportamiento de ventanas cuadradas de Parzen cuando se varía su tamaño	33
4.6. Comportamiento de ventanas gaussianas de Parzen cuando se varía su tamaño	34
4.7. Tiempo de cómputo utilizando ventanas cuadradas y Gaussianas	34
5.1. Escalas gráficas en dos diferentes puntos del planeta	38
5.2. Error de graficación utilizando Geometría plana	39
5.3. Distancia Geodésica s entre dos puntos geográficos	41
5.4. Resultados del algoritmo de graficación de imágenes georeferenciadas a diferentes niveles de acercamiento.	47
6.1. Comparativa entre la gráfica puntual y la estimación de la FDP de las DA detectadas entre el 01/01/2010 y el 01/03/2010.	51
6.2. DA detectadas entre las fechas 01/05/2011 y 11/05/2011	52
A.1. Elipsoide y sus semiejes mayor (a) y menor (b)	56
A.2. Latitud Geocéntrica Ω	56

A.3. Latitud	Geodésica
.....	57
A.4. Latitud Reducida ψ	58
B.1. Ejemplo de declaración de variable.	64
B.2. Página de Bienvenida de APACHE-TOMCAT	65
C.1. Proyecto de aplicación WEB con JSP	69
C.2. Ejemplo proporcionado por Google para colocar una imagen fija sobre los mapas	70
C.3. Paquetes de descarga para NETBEANS	71
C.4. Selección de tipo de proyecto para NETBEANS	71
C.5. Selección del nombre de la aplicación WEB con NETBEANS	72
C.6. Selección del servidor de la aplicación WEB con NETBEANS	73
C.7. Página principal del proyecto SISTORM	74
C.8. Creación del servlet nombrado “ejecuta.java”	75
C.9. Aplicación desarrollada para mostrar la FDP sobre los mapas de Google ..	77

Lista de Tablas

4.1. Tiempos de Ejecución de las ventanas de Parzen	35
---	----

Lista de Algoritmos

1.	Despliegue de DA's sobre una Imagen	22
2.	Graficación de FDP sobre una imagen	30
3.	Conversión de coordenadas de imagen a geográficas	43
4.	Graficación de DA's aplicando algoritmos de Vincenty	44
5.	Graficación de FDP sobre una imagen aplicando algoritmos de Vincenty	45
6.	Fórmula Directa de Vincenty	60
7.	Fórmula Inversa de Vincenty	61

Lista de Símbolos

Z nivel de acercamiento.

I arreglo bidimensional de tamaño $nr \times nc$ donde se almacena la información de la imagen.

nr número de renglones del arreglo I .

nc número de columnas del arreglo I .

$u = [x, y]$ coordenadas planas en Km.

$\hat{u} = [\hat{x}, \hat{y}]$ coordenadas de imagen en pixeles.

r distancia en Km entre la antena y la descarga atmosférica detectada.

$\Delta x, \Delta y$ desplazamiento en Km entre la antena y un punto geográfico.

θ ángulo formado entre el Norte geográfico y la descarga atmosférica detectada medido en el sentido de las manecillas del reloj.

R región acotada del plano dentro de la cual se estima una función de densidad de probabilidad continua.

N número total de muestras que conforman el espacio muestral.

k número de muestras que se encuentran dentro de la región R .

$p(u)$ función de densidad de probabilidad continua.

D número de dimensiones del espacio muestral.

ψ latitud reducida.

ω latitud geocéntrica.

φ latitud geodésica

λ longitud geodésica

a, b son los semiejes mayor y menor de la elipsoide que representa a la Tierra.

$f = (a - b)/a$ achatamiento de la elipsoide.

s distancia geodésica en las mismas unidades que se definan a y b .

α_1, α_2 azimut geodésicos (ángulos inicial y final).

Capítulo 1

Introducción

1.1. Planteamiento del problema

El Sistema Eléctrico Nacional (SEN) es un circuito que cambia con el paso del tiempo y es importante llevar una estadística del comportamiento del mismo. Su estructura es claramente susceptible a las inclemencias del tiempo, en nuestro caso particular de interés, a las DA. En el Centro Nacional de Control de Energía (CENACE), perteneciente a la CFE, se da seguimiento detallado a las indisponibilidades de los equipos instalados debido a las inclemencias del tiempo por ejemplo. Es común tener reportes de indisponibilidad atribuidos a descargas atmosféricas y tormentas eléctricas, por tal motivo es de gran interés para la CFE el tener estudios probabilísticos referentes al comportamiento de los eventos atmosféricos que puedan afectar el funcionamiento del SEN. Un interés a largo plazo por parte de la CFE es el tener pronósticos en tiempo real del comportamiento de los eventos atmosféricos para visualizar sobre mapas que desplieguen la localización geográfica de sus instalaciones y de esta manera poder ver claramente la trayectoria de los siniestros, tales como ciclones o tormentas eléctricas. Tal es el interés de la CFE sobre estos estudios, que ha planeado la instalación de más antenas para la detección de DA de tal forma que la información recabada sea lo más exacta posible triangulando entre las mediciones de diferentes antenas.

Este trabajo propone realizar un estudio estadístico del comportamiento de estos fenómenos

atmosféricos, los cuales son de gran interés para la Comisión Federal de Electricidad División Centro Occidente (CFE-DCO); las DA tienen un gran impacto sobre las fallas detectadas en el SEN. Este trabajo aporta información de manera gráfica referente a la distribución espacial y temporal de las DA en el área geográfica correspondiente a la CFE-DCO, con el objetivo de que se puedan tomar decisiones tácticas en tiempo

Este trabajo se desarrolló en conjunto con el Laboratorio de Máquinas Eléctricas de la División de Estudios Posgrado de la Facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo (DEP-FIE) y la CFE-DCO. Para el desarrollo de este trabajo se cuenta con dos antenas de la compañía Astrogenic Systems modelo BOLTEK LD-250 [BOL10]. Estas antenas fueron adquiridas por CFE-DCO y están localizadas en el Laboratorio de Máquinas Eléctricas de la DEP-FIE. Estas antenas funcionan con una computadora personal con sistema operativo Windows XP y un software llamado NEXTORM, versión 1.20.0.2049 (proporcionado por el fabricante de la antena). La antena junto con una tarjeta de adquisición de datos y el software detectan las DA en un radio de influencia de aproximadamente 480 Km (fuera de este radio podría detectarse ruido como DA's), lo cual nos asegura cubrir la región de interés. La información detectada por la antena es procesada por NEXTORM y almacenada en archivos de texto, por lo cual fue necesario el desarrollo de software de visualización ad hoc de las necesidades de la CFE-DCO. Con la información recabada con NEXTORM de la localización de las DA se aplica el método de ventanas de Parzen [Ross96] para hacer una estimación de la FDP de dichas DA sobre la sección centro-occidente que abarca los estados de Jalisco, Michoacán y Colima. En las Figuras 1.1 a) y 1.1 b) se muestra una de las antenas instaladas en la DEP-FIE, así como la guarda para instalarla a la intemperie y en la Figura 1.1 c) la tarjeta de adquisición de datos de la misma.

La antena tiene una orientación hacia el Norte geográfico terrestre, esta orientación la especifica el fabricante para tener una correcta referencia de la localización de los eventos ya que la manera en que la antena define la posición de una DA es mediante la distancia que existe entre la antena y la DA así como el ángulo que se forma entre el Norte Geográfico y la DA medido en el sentido de las manecillas del reloj, como se muestra en la Figura 1.2

El fabricante, en el área de descargas de la página web [BOL10] nos proporciona el

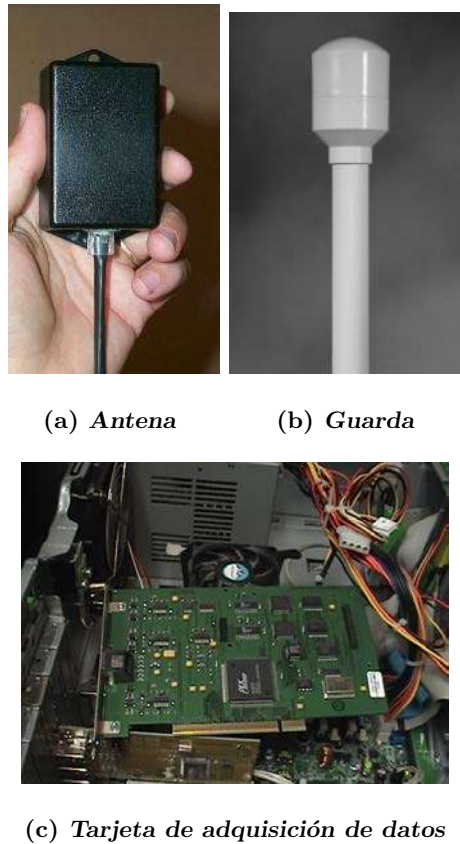


Figura 1.1: Antena, guarda y tarjeta de adquisición de datos BOLTEK LD-250

manual de operación del hardware y software que hemos descrito. El fabricante aclara que aunque el máximo rango de visión de las DA's es de 1200 km (radiales), si la distancia a la que fue detectada la DA es mayor de 500 km la veracidad de la información irá decreciendo conforme la distancia aumente y DA's de baja intensidad podrían no ser detectadas.

1.2. Antecedentes

Antes de proponer este trabajo, en el laboratorio de máquinas eléctricas se desarrolló una aplicación para respaldar la información contenida en los archivos de texto generados por el SOFTWARE de la antena, en una base de datos MySQL. Además se diseñó una página WEB para visualizar esta información, graficando de manera puntual



Figura 1.2: Ubicación de la antena y eje de referencia

la localización aproximada de los eventos que fueron detectados, sobre una fotografía del planeta a diferentes niveles de acercamiento. En la Figura 1.3 se muestra una simulación entre dos fechas, tal y como se tiene implementado en la página web donde se tiene alojada la aplicación [SIS10].



Figura 1.3: Mapa de descargas atmosféricas

En esta página WEB se tiene la posibilidad de realizar las consultas a los eventos registrados, recibiendo como parámetros una fecha de inicio del análisis y una fecha final,

además del nivel de acercamiento que se quiera realizar sobre la posición de la antena, los resultados obtenidos son graficados mediante puntos coloreados sobre las fotografías del territorio geográfico cercano a la antena. El usuario tiene la posibilidad de realizar un acercamiento a la imagen en diferentes niveles, pero con la limitante de que dicho acercamiento se realiza sobre el mismo punto de referencia central, específicamente sobre la antena instalada en la UMSNH. Esta implementación de acercamiento limita al usuario a sólo visualizar el área cercana a la antena y conforme aumenta la resolución, el área geográfica de visualización disminuye debido a que no se tiene la posibilidad de desplazarse libremente sobre el mapa.

Se asume que las escalas que manejan los mapas utilizados conservan una relación constante entre el nivel de acercamiento y la distancia en km representada en el mapa, además se hace uso de geometría plana para determinar las componentes en x y y de los vectores de posición, lo cual a niveles de alta definición o nivel de piso, es válido, pero a niveles de acercamiento bajos donde podemos visualizar países o estados completos no se cumple, esto debido a la irregularidad de la forma del planeta.

En un principio, para la elaboración de esta tesis, hicimos las mismas suposiciones, pero durante el desarrollo del trabajo mostramos la magnitud del error que puede acarrear el uso de estas hipótesis de geometría plana.

1.3. Descripción de capítulos

En el primer capítulo se da detalle de lo que se tenía desarrollado antes de comenzar con el proyecto de maestría para la realización de este trabajo. Se da una descripción de la antena con que se cuenta para la detección de DA y cómo se obtiene la información referente a dichos eventos atmosféricos. Se plantea la problemática a resolver y los requisitos del proyecto.

En el capítulo 2 se da una descripción de cómo se detectan y almacenan las DA censadas por la antena y una descripción de la información de interés para el desarrollo de este trabajo.

En el capítulo 3 explicamos por qué el uso de mapas estáticos para la graficación de las DA y desarrollamos el primer algoritmo para la graficación de DA. Se presentan las primeras hipótesis para el cálculo de la posición de los puntos sobre los mapas, así como algunas aplicaciones del algoritmo presentado.

En el capítulo 4 explicamos el método de ventanas de Parzen para la estimación de la FPD de las DA. Desarrollamos el algoritmo para graficar nuestra estimación de la FPD sobre los mapas estáticos y se muestran algunas pruebas al algoritmo desarrollado en este capítulo.

Las hipótesis hechas en el capítulo 3 para la graficación de DA no son adecuadas para la graficación exacta de las DA ni la FDP. En el capítulo 5 se plantea el uso de los algoritmos presentados en el Anexo A, desarrollados en los 70's por Theodore Vincenty en su trabajo titulado "Direct and Inverse Solutions of Geodesics on the Ellipsoid with application on nested Equations", para el cálculo de distancias geodésicas el cual implementamos para mejorar los algoritmos presentados en los capítulos 3 y 4. Además se implementan fórmulas para la localización de puntos geográficos sobre los mapas, definidas en la documentación proporcionada por los desarrolladores de los servicios de mapas. Estas correcciones nos per-

miten generar imágenes georeferenciadas de los resultados obtenidos.

En el capítulo 6 se muestran las pruebas realizadas a la aplicación desarrollada en su versión final y por último en el capítulo 7 tenemos las conclusiones obtenidas gracias al desarrollo de este trabajo, así como algunos trabajos propuestos para dar continuidad a esta investigación.

Capítulo 2

Detección de descargas atmosféricas y respaldo de la información

Este trabajo parte de la suposición de que la información que se tiene respalda de las DA detectadas es correcta. Es necesario pensar como trabajo posterior la validación de dicha información y para esto es necesario poder comparar estos resultados con mas de un punto de referencia, esto significa triangular las DA detectadas con las detectadas por otras antenas instaladas en diferentes puntos del país.

La CFE-DCO realizó la entrega e instalación de una de las antenas mencionadas en el Capítulo 1. Esta antena se encuentra físicamente en las intalaciones de la UMSNH, específicamente en las coordenadas latitud $\varphi = 19.689545$ y longitud $\lambda = -101.206491$ en el sistema WGS84 [Loweth97]. Dicha antena tiene la capacidad de sensor eventos atmosféricos, realizando una clasificación en base al origen de las descargas y su polarización, además de almacenar la fecha y localización de los eventos detectados mediante coordenadas polares. La antena, al detectar una DA de determinada intensidad, almacena la información que define a dicho evento. Este proceso se registra mediante una tarjeta de adquisición de datos instalada en nuestra PC; este equipo (antena y tarjeta) viene acompañado con el programa NEXTORM para configurar los ajustes de la antena y respaldar la información, así co-

mo para la visualización de los datos. La información referente a las DA se encuentra en archivos de texto plano en la ubicación que nosotros definamos en la configuración del programa NEXTORM.

El personal del laboratorio de Máquinas Eléctricas del Posgrado, el cual es el encargado de la operación y mantenimiento de la antena instalada, tomó la decisión de implementar una base de datos para respaldar la información de las DA que se tenían en un principio contenidas en archivos de texto plano.

Esta base de datos se creó con la característica de almacenar la información referente a la situación espacial de las descargas sensadas. Para esto se almacenó la distancia en Kilómetros entre la antena y la DA detectada, además del ángulo que se forma entre el Norte geográfico y la DA, como se mostró en la Figura 1.2. La probabilidad de que una DA se de sobre un mismo punto es muy pequeña y sería algo extraño pensar que la antena registre más de una DA sobre un mismo punto geográfico. NEXTORM agrupa las DA y las clasifica con un mismo identificador de tormenta cuando se dan dentro una región especificada en las configuraciones de NEXTORM, dicha región se define mediante el ángulo en grados que la circunferencia abarca, y a qué distancia de la antena aún se considera dentro de este rango. En la Figura 2.1 se muestra gráficamente cuál es el criterio para agrupar las DA utilizado por el programa NEXTORM definiendo la distancia d y el ángulo $\Delta\theta$ dentro de los cuales se considerará un mismo identificador para todas las DA que se detecten dentro de este rango. Por ejemplo, todos los puntos dentro de la figura mostrada, serán clasificados mediante un mismo identificador de tormenta, y para el programa, todos los puntos los tomará como uno solo.

La base de datos que se tiene implementada para el proyecto base es muy sencilla, de tal modo que solo cuenta con dos tablas principales donde se almacena la información referente a los eventos detectados por la antena.

En la figura 2.2 se observa los campos que contiene cada una de las tablas de la base de datos SISTORM utilizada para el desarrollo de este trabajo de tesis. Las dos tablas se encuentran relacionadas entre si mediante los campos tormentas.IdReporte y reportes.IdReportes. A continuación se detalla el contenido de estas dos tablas.

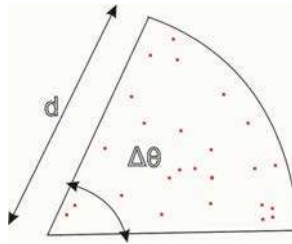


Figura 2.1: Criterio utilizado para el agrupamiento de DA.

tormentas			reportes	
<u>IdTormenta</u>	integer unsigned	<pk>	<u>IdReportes</u>	integer unsigned <pk>
IdReporte	integer unsigned		Fecha	date
IdentTormenta	varchar(50)		Hora	varchar(50)
HoraDetectada	varchar(50)		TotalTormentas	integer unsigned
Angulo	float unsigned		IdEstacion	integer unsigned
Distancia	float unsigned			
UltimaActividad	varchar(50)			
ClaseIntensidad	varchar(50)			
EstadoIntensidad	varchar(50)			
MinimoDescargas	varchar(50)			
MaximoDescargas	varchar(50)			
TotalDescargas	varchar(255)			
DescargasGC	varchar(255)			
DescargasIC	varchar(255)			
GCPos	varchar(255)			
GCNeg	varchar(255)			
ICPos	varchar(255)			
ICNeg	varchar(255)			

Figura 2.2: Definición de la base de datos

Las columnas de la tabla reportes estan definidas de la siguiente manera:

1. **IdReporte.**- es el identificador asignado por el sistema para el evento detectado. Pueden detectarse varios eventos en la misma fecha pero el Identificador es único para cada tormenta.
2. **Fecha.**- Fecha en la cual se ha detectado la tormenta con formato de AAAA-MM-DD.
3. **Hora.**- Hora en la cual se ha detectado la tormenta con formato de hh:mm:ss p.m./a.m.

4. **TotalDescargas** .- Número Total de Descargas Atmosféricas contabilizadas para cada Tormenta.
5. **IdEstacion**.- El proyecto está planeado para tener mas de una estación de monitoreo, pero a la fecha solo se tiene una estación implementada, por tal motivo este campo siempre está en 1.

La tabla reportes está ligada a la segunda tabla ya que con el campo "IdReportes" se almacena información detallada del evento en la tabla "tormentas". La información que se almacena en la tabla "tormentas" es la siguiente:

1. **IdTormenta**.- este identificador es secuencial y se generará para cada evento almacenado en esta tabla.
2. **IdReporte**.- con este identificador se ligan los eventos de ambas tablas.
3. **IdentTormenta**.- este identificador es generado por el software NEXTORM para la identificación de la tormenta detectada, este consiste en una letra seguido de un guion y uno o mas digitos. Este identificador tiene que ver con el momento en que fue detectada la tormenta.
4. **HoraDetectada**.- Es la hora en que fue detectada la tormenta por primera vez.
5. **Angulo**.- La forma en que el sistema posiciona geograficamente la tormenta detectada es por medio de coordenadas polares con respecto al Norte Geografico siendo este nuestra referencia de cero grados. Este campo de la base de datos nos da el angulo en grados al cual fue detectada la tormenta.
6. **Distancia**.- como complemento del campo anterior tenemos la distancia en kilometros a la cual fue detectada la tormenta. Asi, con el ángulo y la distancia podemos localizar geograficamente donde se situó el evento.
7. **UltimaActividad**.- hora en la cual se tubo la última actividad detectada de la tormenta en cuestión.

8. **ClaseIntensidad.**- Clasificación de la intensidad de la tormenta detectada. Esta clasificación puede tomar valores de:

- Weak
- Moderate
- Strong
- Severe

9. **EstadoIntensidad.**- Razón de descargas por minuto detectada en los últimos minutos. Puede tomar valores de:

- Undetermined
- No change
- Increasing
- Decreasing

Hay más información que detecta y clasifica NEXTORM, pero para la realización de este trabajo, solo fue necesario trabajar con los datos comentados en este capítulo. Sin embargo, la base de datos almacena toda la información referente a las DA detectadas por la antena, por ejemplo la intensidad de las DA, la polaridad de la descarga, si fueron entre nubes o de nube a tierra, entre otras características. El diseño e implementación de esta base de datos en MySQL fueron realizados por el personal del Laboratorio de Máquinas Eléctricas del Posgrado. La ventaja de haber implementado dicha base de datos es que se tiene una administración de la información recabada por la antena con programa de distribución libre, del cual se tiene una amplia documentación en la red y las bibliotecas de la Universidad, además de que existe una gran cantidad de información referente al manejo de consultas a bases de datos MySQL utilizando el lenguaje de programación JAVA, y esto facilitó considerablemente la interacción entre nuestro proyecto de tesis y la información que necesitábamos.

2.1. Conclusiones

Terminando este capítulo tenemos almacenado el conjunto de datos necesarios para aplicar las técnicas probabilísticas propuestas para este trabajo de tesis. A partir de aquí se tiene la información almacenada en una base de datos con las características necesarias para el desarrollo de nuestras aplicaciones, con la gran ventaja de la facilidad para interactuar con el lenguaje de programación JAVA que escogimos para desarrollar nuestro trabajo, y la facilidad de poder realizar consultas a la base de datos a través de aplicaciones WEB. En pocas palabras, podemos acceder a la información de las DA´s utilizando JAVA, y desplegar los resultados obtenidos mediante una página WEB. El siguiente paso es definir una metodología para procesar la información que tenemos de las DA para poder visualizarlas sobre los mapas que hemos decidido implementar para este trabajo (Google Maps), lo cual se explicará en el siguiente capítulo.

Capítulo 3

Uso de mapas estáticos para la graficación de los resultados

A partir del trabajo realizado previamente en el Capítulo 2, tenemos la información referente a las DA detectadas, esto es (r, θ) como podemos observarlo en la Figura 1.2, además del número N de incidencias que se dieron sobre ese mismo punto. Lo siguiente era definir la manera de desplegar la información; para esto necesitamos un mapa con la suficiente definición para ubicar las posiciones geográficas obtenidas, de tal manera que pudieramos asociar nuestras DA con la posición real de las instalaciones de la CFE-DCO, esto implica tener un mapa con división política con referencias preferentemente recientes. Decidimos usar los mapas proporcionados por Google, ya que consideramos que los mapas que proporciona cumplen con las necesidades del proyecto [Miller08].

Los mapas implementados por Google utilizan una representación cartográfica llamada Mercator [W. Schofield07], la cual pretende representar la superficie esférica terrestre sobre una superficie cilíndrica, tangente al ecuador, que al desplegarse, genera un mapa terrestre plano. Esta representación nos muestra una buena aproximación en su zona central, pero las zonas superior e inferior correspondientes a Norte y Sur presentan grandes deformaciones debido a la forma esférica del planeta y es por esto que no se representa el planeta en su totalidad, alcanzando una latitud máxima de ± 85.05 grados aproximadamente.

Los mapas estáticos son imágenes o fotografías del planeta Tierra que pueden utilizarse

para presentar información geográfica por ejemplo en una página WEB. Es muy importante señalar que aunque los mapas que se implementaron en este trabajo fueron los proporcionados por la compañía Google, solo se encontró documentación detallada sobre los mapas proporcionados por la empresa MICROSOFT [MUL10] ya que actualmente presta servicios tanto de mapas estáticos como dinámicos, muy parecidos a los servicios de Google Maps. Desde un principio se optó por utilizar los mapas estáticos de Google, ya que con este método podemos obtener imágenes estáticas del área geográfica de interés mediante una consulta HTTP. Para obtener la porción de mapa que deseamos utilizar, la consulta HTTP tiene la siguiente estructura:

```
http://maps.google.com/staticmap?center="+latitud+", "+longitud+"&zoom="+Z+"&size="+nr+"x"+nc+"&motype="+tipo
```

donde *latitud* y *longitud* son las coordenadas geográficas, *Z* es el nivel de acercamiento, *nr* y *nc* definen las dimensiones rectangulares de la imagen del mapa en pixeles y por último se especifica el tipo de mapa. Existen diferentes tipos de mapas (como se muestra en la Figura 3.1) a los que se pueden tener acceso y entre ellos tenemos los mapas de carretera, los satelitales y los híbridos. En el tipo de mapa de la Figura 3.1a) se muestra información referente a la división política de la región, así como de las diferentes vialidades y nombres de lugares principales; en la Figura 3.1b) se muestra una visualización tipo aérea del territorio geográfico de interés; la Figura 3.1c) muestra el tipo híbrido, el cual es una mezcla de los dos últimos tipos de mapa.

Nuestro software se desarrolló en lenguaje Java, por lo cual, en un principio se utilizó la librería JAI (Java Advanced Imaging) para el manejo de imágenes [JAI10]. Durante el desarrollo de una primera aplicación de escritorio, dicha librería fue innecesaria en la versión final del SOFTWARE, gracias a la implementación de los mapas dinámicos implementados en el Capítulo 6. Esta librería consta de un método para leer una imagen desde una dirección URL donde se encuentra almacenada la imagen:

```
PlanarImage imagen = JAI.create("url",url);
```

En la instrucción pasada, la variable URL será una cadena de caracteres con la dirección URL donde se aloje la imagen; en nuestro caso particular, tomará el valor de

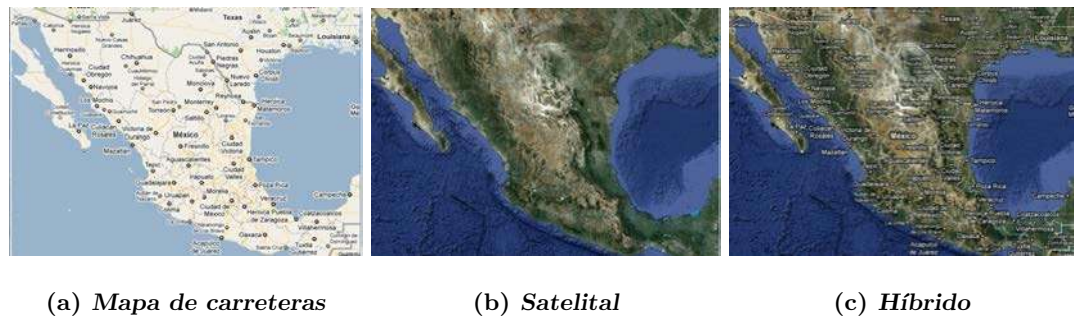


Figura 3.1: Tipos de Mapas utilizados en el proyecto

nuestra petición HTTP. En nuestro caso particular creamos un objeto **PlanarImage**, el cual está definido en la librería JAI. La ventaja de este tipo de objeto es que podemos usarlo como un buffer de imagen, esto para poder manipularla pixel a pixel. El método *create* que crea nuestra imagen recibe como parámetro, la fuente de la imagen que deseamos utilizar, en nuestro caso, la fuente de nuestras imágenes será mediante las consultas HTTP, entonces el primer parámetro será el **URL** entre comillas. El segundo parámetro es el objeto URL que debemos declarar previamente de la siguiente manera:

```
URL url = new URL("http://maps.google.com/staticmap...");
```

donde el constructor del objeto de tipo URL recibe como parámetro la ubicación URL donde se encuentra alojada la imagen con la que deseamos trabajar, escrita entre comillas.

Con las acciones descritas anteriormente ya tenemos nuestro mapa almacenado en un objeto definido en la librería JAI, con el cual podemos manipular nuestra imagen pixel a pixel, o utilizarlo como recurso para la creación de nuestra aplicación. En nuestro caso modificaremos la imagen en los pixeles necesarios para representar los resultados que obtengamos de nuestras implementaciones, por ejemplo, para graficar las DA. En la Figura 3.2 se muestra un mapa generado mediante una consulta HTTP:

```
http://maps.google.com/maps/api/staticmap?center=19.689516,-101.206564&zoom=7&size=640x640&motype=roadmap&size=640*640
```

Con el centro situado en las coordenadas donde se localiza nuestra antena, de tamaño 640×640 pixeles, y tipo de mapa de carreteras.



Figura 3.2: Imagen resultante de una consulta HTTP

Para poder manipular esta imagen contenida en un objeto de tipo **PlanarImage** lo que necesitamos hacer es convertirlo a un objeto de tipo **BufferedImage** de la siguiente manera:

```
BufferedImage imagenBuffer = imagen.getAsBufferedImage();
```

donde **imagen** es el objeto de tipo **PlanarImage** que creamos anteriormente y que contiene, en nuestro caso particular, la imagen del mapa que queremos manipular. De esta forma, si queremos cambiar el color de algún pixel en específico, lo podemos hacer de la siguiente manera:

```
imagenBuffer.setRGB(i, j, rgb);
```

donde este método recibe como parámetros, el entero i que se refiere al pixel en el eje de las abscisas y el entero j el eje de las ordenadas, por último el valor entero del color que queremos que tome la coordenada de imagen en la posición $\langle i, j \rangle$.

La imagen obtenida por la consulta HTTP se representará como un arreglo bidimensional I de tamaño $nr \times nc$ y definiremos $I[\hat{y}, \hat{x}]$ como la información del color (pixel),

correspondiente a las coordenadas \hat{x} y \hat{y} de la imagen $\forall \hat{y} \in [1, \dots, nr], \forall \hat{x} \in [1, \dots, nc]$.

Dado que tenemos coordenadas geográficas (φ, λ) , coordenadas de imagen (\hat{y}, \hat{x}) y coordenadas en Kilómetros (y, x) , es necesario tener constantes para hacer la equivalencia entre ellas. Las constantes para hacer la equivalencia entre latitud y longitud en Kilómetros son $110.57Km/^\circ$ y $112.09Km/^\circ$ [FIS10], estos datos son aproximaciones que son de utilidad para puntos geográficos situados cerca de la línea del Ecuador, pero en realidad no se mantienen constantes y varían en función de la distancia que se aleja del Ecuador. Estas aproximaciones las utilizamos para calcular los desplazamientos entre nuestra antena y un punto con coordenadas longitud y latitud dada por medio de las Ecuaciones 3.1 y 3.2.

$$\Delta x = 110.57km(longitud - 19.689545^\circ) \quad (3.1)$$

$$\Delta y = 112.09km(latitud + 101.206491^\circ) \quad (3.2)$$

donde Δx y Δy representan la distancia horizontal y la vertical en km medidas a partir de la antena. Este cálculo es importante si consideramos que el centro de nuestro mapa podría situarse en una coordenada diferente a la localización geográfica de la antena. Esto nos puede servir para implementar el desplazamiento a distintas zonas geográficas del país ya que con Δx y Δy tenemos la distancia entre nuestra antena y un punto geográfico de nuestro interés. Con esta información podemos calcular las distancias entre el nuevo centro del mapa y las DA detectadas por la antena, como se mostrará en la siguiente sección.

3.1. Despliegue de descargas atmosféricas en mapas estáticos

Para su representación gráfica mediante mapas planos, empresas como Google y Microsoft proponen lo siguiente. La fotografía del planeta Tierra se distribulle sobre una malla o matriz bidimensiona, la cual cambia de dimensiones con respecto al nivel de acercamiento o detalle que queramos visualizar. Por ejemplo, para un nivel de acercamiento bajo o de poco detalle ($Z = 1$) ocuparemos una fotografía mas pequeña de la Tierra que si queremos una fotografía del planeta a un alto nivel de acercamiento ($Z = 19$) con mucho

detalle. En la documentación de los mapas proporcionados por Microsoft ??, se define la ecuación 3.3 para definir las dimensiones de la matriz o malla que contendrá la información de la fotografía del planeta a un determinado nivel de acercamiento Z .

$$nr = nc = 256 * 2^Z \quad (3.3)$$

De tal forma que nuestra matriz tendrá dimensiones de $nr * nc$.

En la documentación de la empresa Microsoft, se plantea una aproximación para calcular la escala de sus mapas en función del nivel de acercamiento Z y la posición del área geográfica que queremos visualizar. Esta ecuación trata de considerar el hecho de que se tiene una representación de Mercator [W. Schofield07] por lo cual tenemos una variación de la escala ($Km/pixel$) de los mapas con respecto a la componente latitud del área geográfica que deseamos visualizar. Lo anterior significa que no tendremos la misma escala si queremos visualizar un mapa en la región del Ecuador que la escala que tendremos de un mapa de la región de Alaska. La manera de definir las escalas utilizadas en cada área geográfica se define en función del nivel de acercamiento Z y la latitud de la coordenada central del área de estudio. La Resolución Terrestre (RT), es la distancia en Kilómetros representada por un pixel de la imagen [BIN10]. La RT en Km/pixel puede ser calculada con 3.4:

$$RT(latitud, Z) = \text{Cos}\left(\frac{latitud * \pi}{180}\right) * \frac{2\pi * 6378.137}{256 * 2^Z} (Km/pixel) \quad (3.4)$$

donde la latitud corresponderá a la ubicación de la antena y Z es el nivel de acercamiento, 6378.137 es el radio aproximado de la Tierra, por tal motivo, $2\pi * 6378.137$ es el perímetro de la circunferencia que se propone para representar el meridiano mayor de la Tierra. Entonces, la ecuación 3.4 nos calcula cuantos Kilometros representará cada pixel de la imagen de tamaño $nr * nc$ calculado con la ecuación 3.3 en función del Coseno de la componente latitud de nuestra área geográfica de interes, esto significa que conforme nos acerquemos a los polos la escala irá disminuyendo por efecto de la representación de Mercator [W. Schofield07].

En nuestro caso particular con la antena localizada en la latitud 19.689565, RT será considerado solamente función de Z , y podemos reemplazar por 3.5.

$$RT(Z) = \frac{156.54022}{2Z} (Km/pixel) \quad (3.5)$$

Nuestro primer prototipo desarrollado toma la información r, θ (como se mostró en la Figura 1.2) de cada DA y la localización de la antena en coordenadas geográficas de acuerdo al WGS84 [Loweth97]. Las DA son almacenadas como un conjunto de N puntos $u_i = [r_i, \theta_i] \equiv [x_i, y_i], \forall i \in [1, 2, \dots, N]$, por lo cual es necesario hacer la conversión de polar a rectangular utilizando la Ecuación 3.6 para las N descargas sensadas:

$$u_i = [x_i, y_i]^T = [r_i \sin(\theta_i), r_i \cos(\theta_i)]^T \quad (3.6)$$

Para poder graficar cada uno de los puntos o descargas u_i , es necesario calcular su equivalencia \hat{u}_i (con $\hat{u}_i \in \mathbb{Z}^2$) en coordenadas de la imagen en función del nivel de acercamiento Z y la coordenada geográfica del centro de dicha imagen. Este punto geográfico (centro de la imagen) estará representado en el centro de la misma con coordenadas de imagen $\hat{x}_i = \frac{nc}{2}$ y $\hat{y}_i = \frac{nr}{2}$. Para esto, dividimos nuestra coordenada u_i , referidas a la latitud y longitud, entre el valor $RT(Z)$. Las nuevas coordenadas \hat{u}_i de imagen las calculamos con las ecuaciones 3.7 y 3.8.

$$\hat{x}_i = \left\lfloor \frac{nc}{2} + \frac{x_i - \Delta x}{RT(Z)} \right\rfloor \quad (3.7)$$

$$\hat{y}_i = \left\lfloor \frac{nr}{2} - \frac{y_i - \Delta y}{RT(Z)} \right\rfloor \quad (3.8)$$

donde la función $\lfloor x \rfloor$ calcula el valor entero de x , ya que estas son coordenadas de imagen en pixeles y es necesario redondear a números enteros.

El siguiente algoritmo muestra el procedimiento completo para graficar cada una de las DA sobre una imagen I .

Por ejemplo, en la Figura 3.3 graficamos de manera puntual las DA detectadas entre el 1 de Enero del 2010 y el 1 de Marzo del mismo año, podemos observar que la imagen resultante no nos da más información que la localización geográfica de los puntos y es imposible definir

DESPLIEGA($I, Z, n_r, n_c, longitud, latitud, u$)

- 1 Dadas longitud y latitud calcular Δx y Δy utilizando 3.1 y 3.2
- 2 Calcular $RT(Z)$ utilizando 3.5
- 3 **para** $i \leftarrow 1$ **hasta** N
- 4 Convertir u_i a coordenadas rectangulares con 3.6
- 5 Calcular las coordenadas de imagen \hat{u}_i con 3.7 y 3.8
- 6 Marcar el pixel de la imagen haciendo $I[\hat{y}_i, \hat{x}_i] = 0$, Si $0 \leq \hat{x}_i < n_c$ y $0 \leq \hat{y}_i < n_r$
- 7 **regresar** imagen marcada I

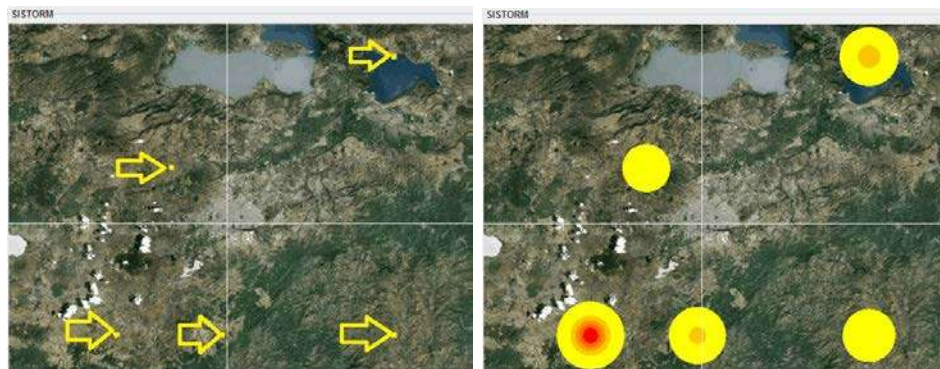
Algoritmo 1: Despliegue de DA's sobre una Imagen

cual es el área de mayor incidencia de DA. Es por esta razón que nosotros proponemos crear imágenes georeferenciadas con las cuales podamos responder a la siguiente pregunta ¿Cuál es el área de mayor incidencia de DA?



Figura 3.3: Descargas Atmosféricas detectadas entre el 01/01/2010 y 01/03/2010

Para mostrar y justificar la estimación de la función de densidad de probabilidad se realizó un experimento ilustrativo. Generamos un número N de DA al azar, con la particularidad de que la distribución de los puntos se concentraba sobre 5 regiones geográficas elegidas al azar, pero dicha distribución no es uniforme. Por ejemplo, de 1000 DA generadas aleatoriamente podrían distribuirse de la siguiente manera: 400, 300, 200, 50 y 50 DA respectivamente. Si sólo graficáramos la localización geográfica de las DA, podríamos llegar a un caso extremo como el mostrado en la Figura 3.4 a) donde las DA se concentran en 5 puntos bien específicos del mapa señalados con flechas. De esta manera no podríamos distinguir en qué lugar se dió un mayor número de descargas. En la Figura 3.4 b) mostramos la estimación de la FDP, en la cual se colorea de rojo la zona de mayor probabilidad de incidencia de DA y en color amarillo la menor probabilidad de incidencia. Note que en la



(a) Imagen de los puntos de DA

(b) Estimación de la FDP

Figura 3.4: Datos Puntuales y Estimación de la Función de Densidad de Probabilidad

Figura 3.4 a) solamente se hace la representación puntual de las DA lo cual no clarifica en qué lugar de ésta hay mayor incidencia de DA. En la Figura 3.4 b) se presenta la estimación de la FDP, de las DA mostradas en la Figura 3.4 a). A diferencia de la Figura 3.4 a), en la Figura 3.4 b) podemos ver la influencia y la probabilidad de las DA. En este trabajo proponemos que en lugar de hacer una representación puntual de los datos, sea remplazada por la estimación de la FDP, que nos permita dar una imagen más clara de cómo se distribuyen la descargas en un mapa. Esta estimación de la FDP tendrá el propósito de dar

información adicional a la posición de las DA. En el siguiente capítulo se dan los detalles de cómo estimar la FDP utilizando ventanas de Parzen.

3.2. Conclusiones

Finalizado este Capítulo podemos imprimir las DA que ha sentido la antena instalada. Para esto hicimos uso de los mapas proporcionados por Google, y definimos un algoritmo para graficar los puntos sobre dichos mapas. Es claro que la representación gráfica de las DA sobre el mapa no es suficiente información para evaluar el comportamiento de los eventos atmosféricos en el país. Para esto proponemos hacer la estimación de la FDP y colorear de diferentes tonalidades las áreas de mayor probabilidad de descarga, esto lo trataremos en el Capítulo 4.

Capítulo 4

Despliegue de funciones de densidad de probabilidad

4.1. Funciones de densidad de probabilidad

Hasta el Capítulo 3 sólo se han generado imágenes puntuales de las DA detectadas por la antena. La información que obtenemos de dichas imágenes no es suficiente para valorar la incidencia de las DA sobre el territorio geográfico de interés. Hasta ahora sólo tenemos la ubicación de los puntos. El siguiente paso que proponemos es mejorar estos resultados presentando la estimación de la FDP sobre los mapas estáticos que hemos utilizado hasta ahora. Con esta nueva implementación podremos visualizar fácilmente el lugar donde se dio la mayor incidencia de DA.

La definición matemática de una Función de Densidad de Probabilidad continua $p(u)$ [Ross96] satisface las siguientes propiedades:

1. La probabilidad $P(u \in R)$ de que u esté dentro de una región R se define como:

$$P(u \in R) = \int_R p(u) du \quad (4.1)$$

2. La probabilidad de que u esté en cualquier región siempre es positiva o cero.

3. La integral de la función de probabilidad en todo el universo es:

$$\int_U p(u) du = 1 \quad (4.2)$$

4.2. Ventanas de Parzen

Dado un conjunto de N muestras $\{u_1, u_2, \dots, u_N\}$ con $u_i \in \mathbb{R}^D$ donde D es el número de dimensiones de nuestro espacio muestral, podemos estimar la FDP. Esta FDP será útil para calcular la probabilidad $p(v)$ de que una nueva muestra v esté en una región R .

Las ideas básicas detrás de muchos de los métodos de estimación de una FDP son muy simples. La idea básica se basa en calcular la probabilidad de que un punto caiga dentro de una región R , para lo cual debemos hacer la estimación de 4.1. Para estimar 4.1 asumimos que la región R es tan pequeña que $p(u)$ no varía mucho y puede ser considerada constante dentro de ésta. Con esta consideración podemos partir de 4.1, calcular 4.3.

$$P(u \in R) = \int_R p(u) du \approx p(u) \int_R du \approx p(u)V \quad (4.3)$$

donde V es el volumen de nuestra región R .

Por otro lado vamos a suponer que las N muestras son independientes y que k es el número de muestras que caen dentro de la región R , por lo tanto la probabilidad de que una muestra esté dentro de una región, estará dada por:

$$P(u \in R) = k/N \quad (4.4)$$

Las ecuaciones 4.3 y 4.4 nos llevan a la siguiente estimación para $p(u)$

$$p(u) = \frac{k/N}{V} \quad (4.5)$$

El método de ventanas de Parzen, para la estimación de la FDP se basa en considerar que R es un hipercubo centrado en la posición u_i . Consideremos que h es la longitud de las aristas del hipercubo, entonces en dos dimensiones R es un cuadrado con volumen $V = h^2$ y en tres dimensiones R es un cubo con volumen $V = h^3$.

Definimos ϕ como la función de ventana, la cual nos permite determinar si una muestra v se encuentra dentro de un hipercubo centrado en u_i y de lado h . Esta función está definida como:

$$\phi\left(\frac{u_i - v}{h}\right) = \begin{cases} 1 & \text{si } \frac{|u_i - v|}{h} \leq 0.5, \forall i = 1, 2, \dots, N \\ 0 & \text{en cualquier otro caso} \end{cases} \quad (4.6)$$

Podemos calcular el total de muestras k , dentro de la región R , utilizando la suma de N funciones de ventana cuadradas, centradas en u_i y dadas por 4.6; así k se puede calcular utilizando 4.7 y la FDP utilizando ventanas cuadradas, estará dada por la ecuación 4.8.

$$k = \sum_{i=1}^N \phi\left(\frac{u_i - v}{h}\right) \quad (4.7)$$

$$p(v) = \frac{1}{Nh^D} \sum_{i=1}^N \phi\left(\frac{u_i - v}{h}\right) \quad (4.8)$$

donde D es el número de variables aleatorias dentro de nuestro espacio muestral.

Mostraremos gráficamente la implementación de estas definiciones mediante un ejemplo numérico con el siguiente conjunto de datos:

$$x = [5.1 \ 5.2 \ 4.9 \ 5.3 \ 4.8 \ 8.9 \ 9.0 \ 9.1 \ 9.2 \ 8.2]$$

En la Figura 4.1 podemos observar el histograma correspondiente al ejemplo numérico que planteamos. Recordemos que un histograma es la representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados.

Si implementamos el método de ventanas de Parzen aplicando ventanas vuadradas, podemos observar una gráfica idéntica al histograma de la Figura 4.1. En la Figura 4.2 se muestra la gráfica de barras correspondiente a la implementación de ventanas de Parzen vuadradas de tamaño $h = 1$.

Podemos generalizar la idea para que permita utilizar otras funciones ϕ de ventana, dando lugar a otros métodos para la estimación de la FDP utilizando ventanas de Parzen. Por ejemplo si utilizamos una función Gaussiana de ventana, la formulación de ventanas de

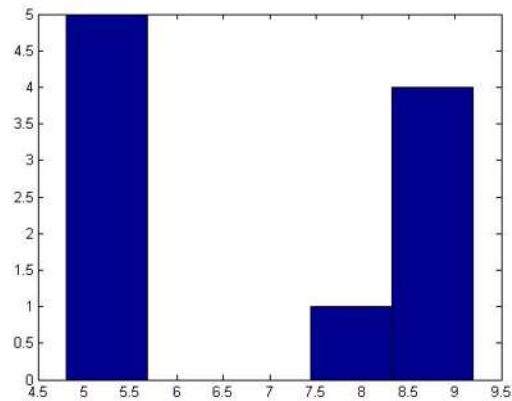


Figura 4.1: Histograma resultante del ejemplo numérico propuesto.

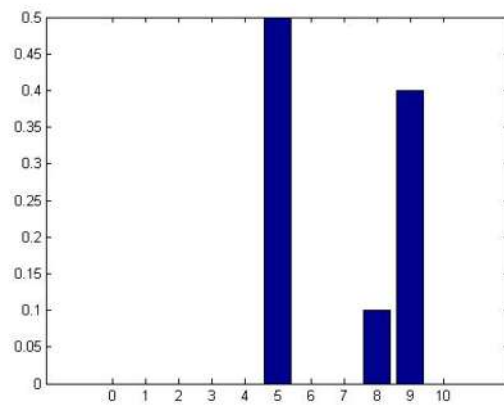


Figura 4.2: Implementación de ventanas de Parzen cuadradas con $h = 1$.

Parzen estará dada por:

$$p_g(v) = \frac{1}{Nh(2\pi)^{D/2}} \sum_{i=1}^N \exp\left(-\frac{(u_i - v)^2}{2h^2}\right) \quad (4.9)$$

Si implementamos las ventanas Gaussianas al método de Parzen, obtendremos una estimación de la FDP mas suavizada con respecto a las ventanas cuadradas. En la Figura 4.3 tenemos la gráfica de barras para la implementación de ventanas Gaussianas de tamaño

$h = 1$ para el mismo ejemplo numérico de las Figuras 4.1 y 4.2. En esta figura podemos observar una gráfica que tiene valores intermedios entre las dos modas, lo cual es una función de densidad de probabilidad más suavizada que la presentada en el histograma.

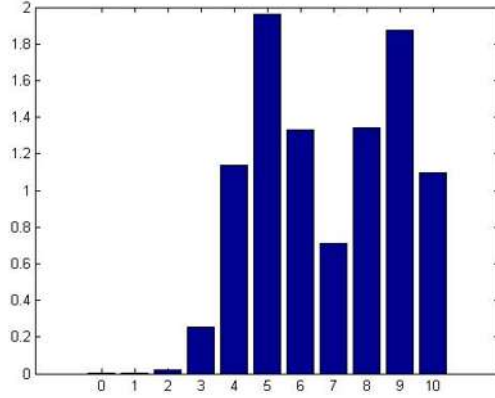


Figura 4.3: Implementación de ventanas de Parzen Gaussianas con $h = 1$

Con las expresiones para el cálculo de la FDP utilizando ventanas cuadradas y ventanas Gaussianas, podemos hacer la implementación sobre una imagen. En la siguiente sub-sección se muestran los detalles y el algoritmo para ello.

4.3. Imagen de la FDP

Dado un conjunto de descargas $\{u_1, u_2, \dots, u_N\}$ es necesario hacer la representación de la FDP sobre una imagen. Para ello necesitamos conocer las coordenadas geográficas $\langle \varphi, \lambda \rangle$ y nivel de acercamiento Z para poder hacer el equivalente en coordenadas de la imagen $\hat{u}_i = [\hat{x}_i, \hat{y}_i]^T$ a coordenadas planas $u_i = [x_i, y_i]^T$.

A partir de las ecuaciones 3.7 y 3.8, si despejamos x_i y y_i calculamos el equivalente de las coordenadas de la imagen utilizando 4.10 y 4.11.

$$x_i = \Delta x + \left(\hat{x}_i - \frac{nc}{2}\right) * RT(Z) \quad (4.10)$$

$$y_i = \Delta y - \left(\hat{y}_i - \frac{nr}{2}\right) * RT(Z) \quad (4.11)$$

Con los valores calculados utilizando las ecuaciones 4.10 y 4.11 podemos estimar para todos los puntos de la imagen un vector $v = [x, y]^T$ y hacer la evaluación de la FDP por medio de la ecuación 4.8 o 4.9, en el caso de ventanas cuadradas o ventanas Gaussianas, respectivamente.

Con propósitos de visualización los valores de la imagen de la FDP, son escalados a un rango entre 0 y 255, todo esto da lugar al siguiente algoritmo.

GRAFICAFDP($I, Z, nr, nc, \lambda, \varphi, v$)

- 1 Dado longitud y latitud calcular Δx y Δy utilizando 3.1 y 3.2
- 2 Calcular $RT(Z)$ utilizando 3.5
- 3 **para** $\hat{y} \leftarrow 0$ **hasta** nr
- 4 **para** $\hat{x} \leftarrow 0$ **hasta** nc
- 5 Calcular $v = [x, y]^T$ con 4.10 y 4.11
- 6 Calcular $p(v)$ utilizando 4.8 o 4.9
- 7 Hacer $I[\hat{y}, \hat{x}] = p(v)$
- 8 Hacer max igual al Máximo de I
- 9 Hacer min igual al Mínimo de I
- 10 **para** $\hat{y} \leftarrow 1$ **hasta** nr
- 11 **para** $\hat{x} \leftarrow 1$ **hasta** nc
- 12 Hacer $I[\hat{y}, \hat{x}] = 255 * (I[\hat{y}, \hat{x}] - min) / (max - min)$
- 13 **regresar** imagen marcada I

Algoritmo 2: Graficación de FDP sobre una imagen

Si tenemos la imagen del mapa que queremos utilizar (obtenida mediante la peti-

ción HTTP descrita en el Capítulo 3 de este trabajo), y sabemos a que coordenadas $\langle \varphi, \lambda \rangle$ corresponde cada pixel de nuestro mapa, con el algoritmo anteriormente propuesto podemos crear un mapa de la FDP correspondiente a las DA que fueron detectadas por la antena. Para crear la imagen de la FDP utilizamos el algoritmo para graficar la FDP utilizando como base el mapa estático de la región que vamos a estudiar y modificando sólo los pixeles donde la FDP sea mayor a cierta tolerancia que debemos especificar, esto es para que nuestro mapa no sea reemplazado completamente y sólo se cambien los pixeles donde la FDP sea más representativa.

Se tomaron algunos datos reales almacenados de DA. El conjunto de DA presentado es de tamaño $N = 1226$ y los mapas fueron obtenidos con un nivel de acercamiento $Z = 7$. La implementación de este algoritmo se presenta, utilizando ventanas cuadradas en la Figura 4.4 a) y gaussianas en la Figura 4.4 b) respectivamente.

En la Figura 4.4 podemos observar las curvas de nivel de la FDP para los eventos detectados para el mes de Septiembre del 2008 utilizando ventanas de Parzen de tamaño 10. Note como las ventanas Gaussianas nos dan una distribución que luce mas suave. Para generar estas imágenes se implementó un detector de bordes sobre la imagen generada de la FDP ya que de lo contrario, al graficar nuestra estimación sobre nuestro mapa, ocultaríamos información importante sobre la localización de nuestro mapa.

La antena solo censa el punto donde ocurre la DA adoleciendo de información referente al área de influencia de las DA. Por esta razón, el tamaño de ventana h tendrá que ser estimado por el usuario y servirá para marcar la influencia de la DA. Con el propósito de dar una idea clara de cómo influye esta variable se realizó un experimento en el cual se calculó la FDP utilizando cuatro valores diferentes para h , estos valores fueron 20, 30, 40 y 50 utilizando ventanas de Parzen cuadradas. El área de influencia de la DA será proporcional al tamaño de la ventana, dando lugar a que cada dato tenga mayor influencia sobre su alrededor y si h es lo suficientemente grande, los datos puntuales pueden verse como una bola. En las Figura 4.5 y Figura 4.6 se muestra la FDP para el mismo conjunto de la Figura 4.4 con diferentes tamaños de ventana, note como a medida que el tamaño de la ventana aumenta tenemos un área de influencia mayor.

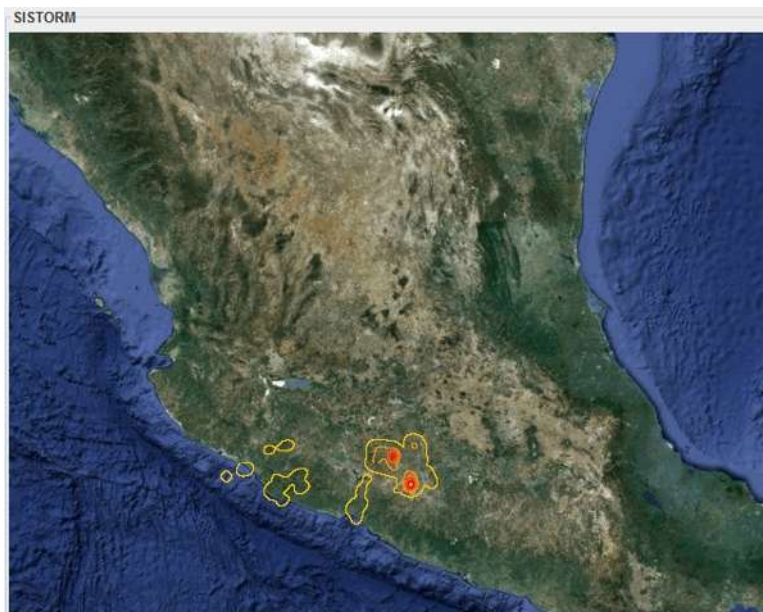
(a) *ventanas cuadradas*(b) *ventanas Gaussianas*

Figura 4.4: Cálculo de la distribución de probabilidad utilizando ventanas de Parzen de tamaño 10

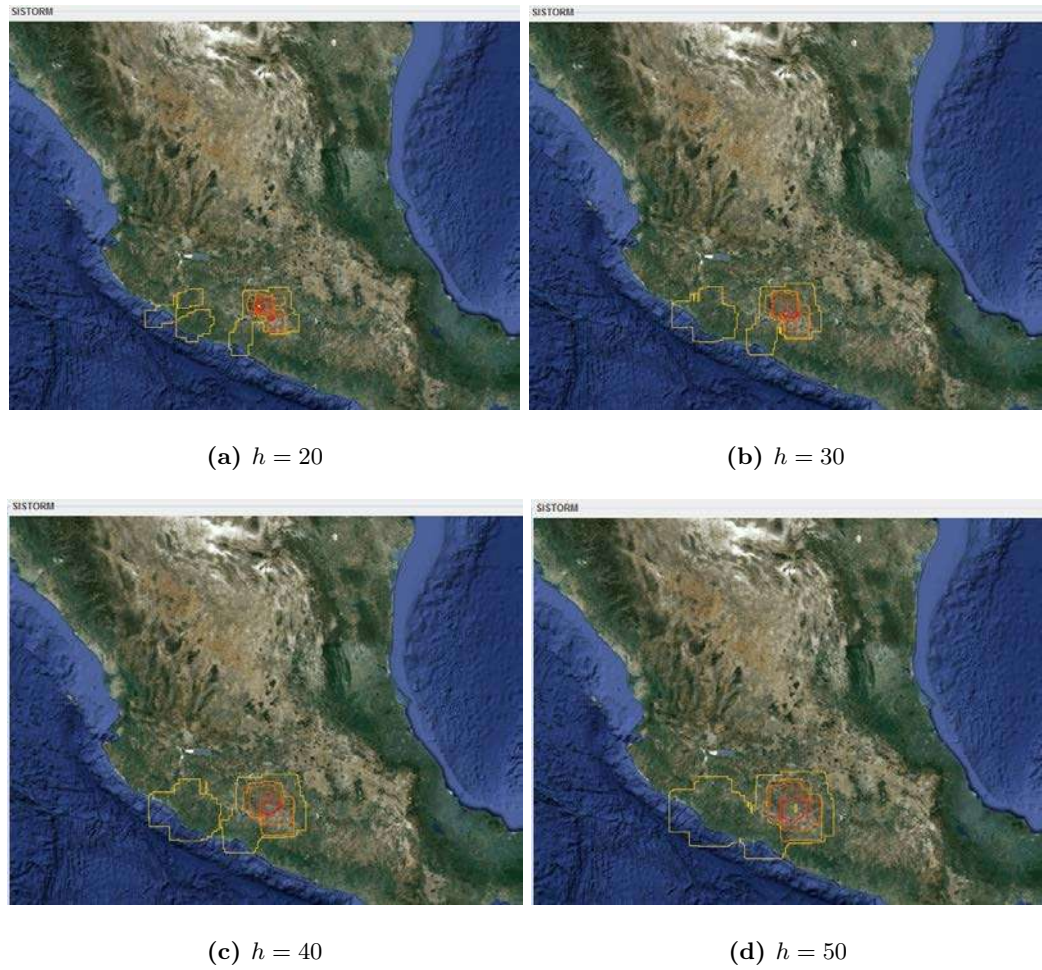


Figura 4.5: Comportamiento de ventanas cuadradas de Parzen cuando se varía su tamaño

El tiempo de cómputo para el cálculo varía de manera diferente cuando se utilizan ventanas de Parzen Gaussianas y ventanas cuadradas, esto es debido al cálculo de la exponencial de la ecuación 4.9. En la Tabla 4.1 se muestra el tiempo de ejecución en milisegundos, obtenido cuando se varía el número de datos y en la Figura 4.7 la gráfica correspondiente. Note en ambas como el tiempo utilizando ventanas Gaussianas es mucho mayor.

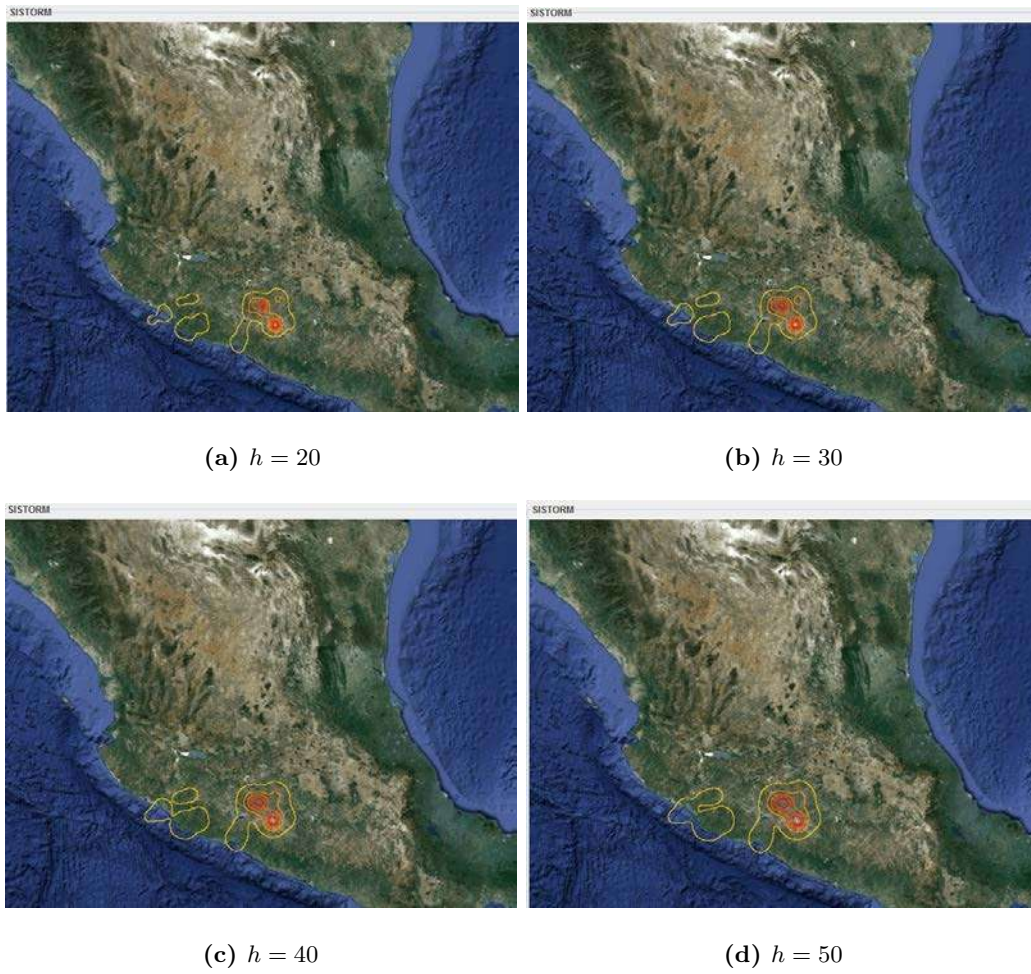


Figura 4.6: Comportamiento de ventanas gaussianas de Parzen cuando se varía su tamaño

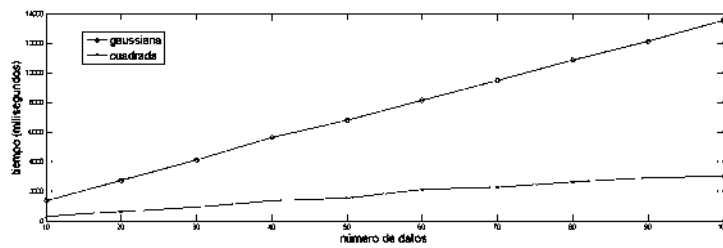


Figura 4.7: Tiempo de cómputo utilizando ventanas cuadradas y Gaussianas

Tabla 4.1: Tiempos de Ejecución de las ventanas de Parzen

Número de datos	Tiempo(ms) cuadradas	Tiempo (ms) Gaussianas
10	312	1373
20	624	2746
30	920	4102
40	1357	5631
50	1525	6801
60	2122	8127
70	2277	9501
80	2636	10857
90	2917	12121
100	3023	13557

4.4. Conclusiones

Con el algoritmo propuesto podemos generar una imagen con la estimación de la FDP, y si tomamos como fondo de esta imagen los mapas estáticos que hemos estado manejando, podemos crear mapas de las zonas geográficas de nuestro interés que nos den información gráfica del comportamiento de las DA. Hasta esta etapa del trabajo de tesis se habían tomado ciertas suposiciones referentes al cálculo de las distancias geográficas y su relación con las distancias equivalentes en los mapas usados, se asumió que podrían utilizarse constantes numéricas para calcular la equivalencia entre pixeles del mapa y kilómetros de la Tierra, en el siguiente capítulo se explica el error generado por dichas suposiciones y la forma en que fue corregido.

Capítulo 5

Corrección de los métodos de graficación aplicados

5.1. Errores de precisión

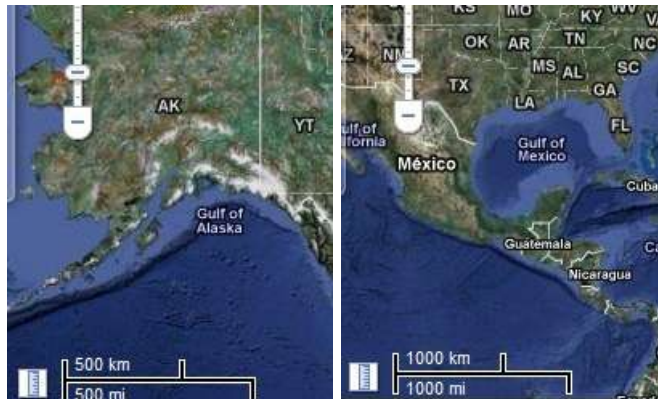
En la primera parte de este trabajo se utilizaron algunas aproximaciones para graficar la información referente a los eventos sensados por la antena sobre los mapas estáticos. La primer aproximación fue pensar que las coordenadas latitud y longitud conservaban una relación constante conforme aumentaban o disminuían.

1 grado de longitud = $112.09Km$

1 grado de latitud = $110.57Km$

Estas medidas son muy cercanas a la realidad en la línea del ecuador, pero conforme nos alejamos de esta línea imaginaria dicha relación no se cumple. En la Figura 5.1 a) en la parte inferior izquierda podemos observar la escala que corresponde a una región cerca de Alaska en el Norte del Planeta, mientras que en la Figura 5.1 b) en la parte inferior izquierda tenemos la escala correspondiente a una región cerca de Centro América al mismo nivel de acercamiento, observaremos que son totalmente distintas debido al tipo de proyección utilizado y a la curvatura del Planeta.

Investigando un poco más, en la documentación de los mapas estáticos de Microsoft [BIN10] se propone la Ecuación 3.4 (previamente descrita en el Capítulo 3 de este



(a) *Escala cerca de Alaska* (b) *Escala cerca de Centro América*

Figura 5.1: Escalas gráficas en dos diferentes puntos del planeta

trabajo) para calcular la resolución terrestre de sus mapas en función de la componente longitud de la coordenada terrestre del punto central del mapa y el nivel de acercamiento. Para mostrar gráficamente el error generado al utilizar escalas constantes presentamos los siguientes ejemplos. En la figura 5.2 tenemos simulados una serie de puntos equidistantes y graficados con puntos sobre el mapa. Los puntos señalados con marcadores tipo globo en las Figuras 5.2 a) y 5.2 b) son implementados mediante las herramientas de Google Maps. Esta herramienta nos permite situar marcadores sobre sus mapas definiendo las coordenadas geográficas donde queremos graficar nuestros puntos. Los puntos en color azul que se muestran debajo de la posición de los marcadores de globo, son calculados utilizando geometría plana con las relaciones constantes que comentamos al principio de este capítulo y que fueron implementados en los primeros cuatro capítulos de este trabajo. Estas posiciones se calcularon implementando los algoritmos que se presentarán en la siguiente sección de este capítulo y que hacen uso de geometría elíptica para su desarrollo.

Podemos observar que a un nivel de acercamiento alto ($Z = 10$ como en la Figura 5.2 a) el error acumulado es imperceptible a la vista, pero conforme el nivel de acercamiento disminuye ($Z = 2$ Figura 5.2 b) es fácil observar el error que origina utilizar geometría plana utilizando las hipótesis y constantes antes mencionadas, este error es debido a que no consideramos

que la distancia entre dos puntos geográficos no es una recta sino un arco llamado geodésica.



Figura 5.2: Error de graficación utilizando Geometría plana

La solución a estos problemas de graficación radica en el cálculo de geodésicas y uso de geometría elíptica para el estudio de los arcos que se originan entre dos puntos sobre la superficie terrestre, en la siguiente sección se definirán los conceptos básicos para el Cálculo de Geodésicas.

5.2. Cálculo de geodésicas

Al arco más corto entre dos puntos sobre una esfera se le llama geodésica y tal distancia se representa por la letra s . En nuestro caso de estudio tenemos un conjunto de puntos situados sobre el globo terraqueo, por lo cual es necesario seguir una metodología para el cálculo de las distancias entre dichos puntos. En la Figura 5.3, ilustramos la distancia geodésica s entre dos puntos p_1 y p_2 situados en coordenadas $\langle \varphi_1, \lambda_1 \rangle$ y $\langle \varphi_2, \lambda_2 \rangle$ sobre una elipse de revolución. Si p_1 y p_2 fueran dos puntos sobre un plano, podríamos desplazarnos de p_1 a p_2 con un vector de magnitud $p_1 - p_2$ en dirección a p_2 , y para regresar

de p_2 a p_1 simplemente cambiaría la dirección en 180° . Ahora, si nuestros puntos p_1 y p_2 están situados sobre una elipse de revolución, el desplazamiento de p_1 a p_2 y viceversa se complica para distancias largas, porque si la distancia entre p_1 y p_2 es muy pequeña (menos de la diezmilésima parte del diámetro mayor de la) podemos utilizar geometría plana para trazar la ruta entre estos dos puntos y los resultados obtenidos serían de muy alta precisión, pero si la distancia entre p_1 y p_2 es muy grande, regresar de p_1 a p_2 con un ángulo de 180° ya no es posible.

Para ir de p_1 a p_2 sobre una elipsoide de revolución necesitamos recorrer una distancia s saliendo de p_1 con un ángulo α_1 y para regresar de p_2 a p_1 , es necesario recorrer una distancia s saliendo de p_2 con un ángulo $\alpha_2 \neq \alpha_1 - 180^\circ$, como se puede observar en la Figura 5.3.

Como recordaremos, contamos con las coordenadas geográficas del lugar donde se encuentra instalada la antena y tenemos la distancia y el ángulo que se forma entre la antena y la DA, el problema se centra en calcular las coordenadas geográficas $\langle \varphi, \lambda \rangle$ que corresponde a cada DA, esto es solucionado mediante la implementación de los algoritmos formulados por Theodore Vincenty en la década de los 70's [Vincenty75], el pseudocódigo de estos algoritmos se presentan en el Anexo A de este trabajo.

5.3. Georeferencia de eventos atmosféricos

Teniendo nuestros eventos atmosféricos en coordenadas polares $\langle r, \theta \rangle$ podemos calcular los puntos geográficos correspondientes a cada evento aplicando la fórmula directa de Vincenty implementada en el Anexo A de este trabajo, pasando como argumentos la localización geográfica de nuestra antena como primer punto p_1 en coordenadas $\langle \varphi_1, \lambda_1 \rangle$, y la información de la base de datos referente a la posición en coordenada polar $\langle r, \theta \rangle$ de cada DA, y como resultado obtendríamos las coordenadas $\langle \varphi_2, \lambda_2 \rangle$ correspondientes a cada una de las DA.

Al tener todos nuestros eventos georeferenciados podemos graficarlos de manera puntual y precisa mediante las herramientas que ofrece Google Maps con marcadores en Mapas

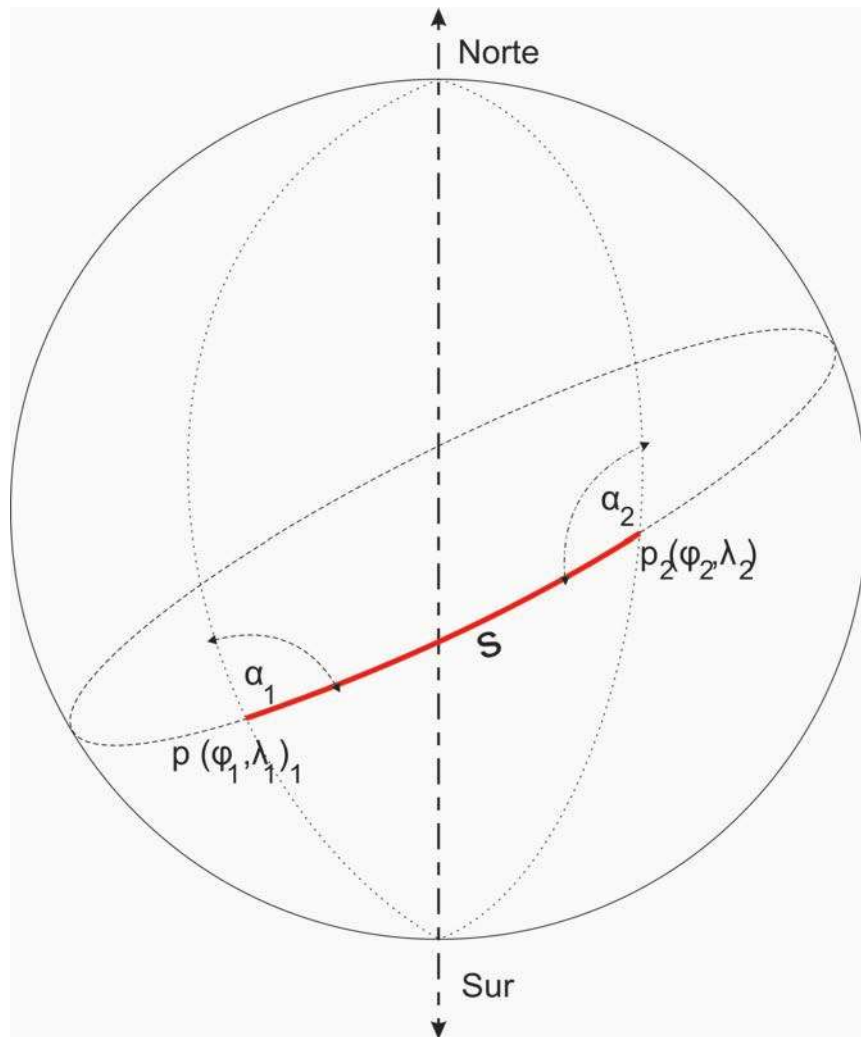


Figura 5.3: Distancia Geodésica s entre dos puntos geográficos

Estáticos. De esta forma podríamos tener una representación puntual más exacta que la que se proponía al principio de este trabajo.

Google permite al usuario generar marcadores georeferenciados sobre sus mapas tomando como argumento las coordenadas geográficas $\langle \varphi, \lambda \rangle$ donde se colocará dicho marcador, esta opción incluye a los mapas estáticos de Google agregando las siguientes instrucciones a nuestra petición HTTP para cada punto que queramos marcar en nuestro mapa.

`&markers=latitud,longitud`

En donde sustituiremos *latitud, longitud* con las coordenadas geográficas de nuestros eventos a graficar. De esta forma podemos generar un mapa con marcadores georeferenciados que señalen donde se detectaron las DA de manera puntual. Pero el objetivo de este proyecto no es tener una representación puntual de las DA debemos calcular una representación gráfica de la FDP de las DA y graficarla sobre los mapas que hemos estado utilizando a lo largo de este trabajo. Para esto, es necesario realizar las modificaciones necesarias para obtener la posición exacta en el mapa de nuestra imagen de la FDP.

Hasta este momento, con los algoritmos desarrollados por Theodore Vincenty e implementados en esta sección, podemos calcular las coordenadas geográficas $\langle \varphi, \lambda \rangle$ correspondientes a cada DA. Mediante la fórmula Directa de Vincenty, proporcionando como argumentos para nuestro algoritmo los valores de r, θ correspondientes a cada DA y las coordenadas geográficas $\langle \varphi_1, \lambda_1 \rangle$ correspondientes a la localización de nuestro centro de referencia (en el caso de no tener desplazamiento, sería la localización de la antena), obtendremos como resultado para cada DA las coordenadas $\langle \varphi_2, \lambda_2 \rangle$ para cada una de las DA.

Si ya tenemos nuestras DA georeferenciadas, el siguiente paso es calcular a qué pixel de la imagen mapea cada una de las coordenadas geográficas de nuestras DA, esto se realiza aplicando las fórmulas definidas en la documentación de los mapas y presentadas en la siguiente sección.

5.4. Mapas de Google

Los mapas implementados por Google y Microsoft representan la superficie terrestre mediante imágenes cuyo tamaño se define en función del nivel de acercamiento Z mediante la Ecuación 5.1.

$$size(Z) = (256 * 2^Z) - 1 \quad (5.1)$$

Por ejemplo, al nivel 3 de acercamiento el rango de coordenadas en pixeles va de (0,0) a (2047,2047). En la documentación proporcionada por Microsoft referente a sus mapas estáticos, se nos proporcionan algunas ecuaciones para calcular las coordenadas de imagen $\langle \hat{i}, \hat{j} \rangle$ dadas las coordenadas geográficas $\langle \varphi, \lambda \rangle$ en grados y el nivel de acercamiento

Z , mediante las ecuaciones 5.2 y 5.3 definidas en [BIN10]:

$$\hat{i}(\lambda, Z) = ((\lambda + 180)/360) * 256 * 2^Z \quad (5.2)$$

$$\hat{j}(\varphi, Z) = (0.5 - \log((1 + \text{Sen}(\frac{\varphi\pi}{180}))/ (1 - \text{Sen}(\frac{\varphi\pi}{180}))) / 4\pi) * 256 * 2^Z \quad (5.3)$$

De manera inversa, podemos obtener las coordenadas geográficas $\langle \varphi, \lambda \rangle$ de cualquier pixel en el mapa a determinado nivel de acercamiento Z mediante el siguiente algoritmo:

PIXLATLON(\hat{i}, \hat{j}, Z)

- 1 Calcular $mapsize = size(Z)$ utilizando 5.1
- 2 $x = ((\text{Min}(\text{Max}(\hat{i}, 0), mapsize - 1)/mapsize) - 0.5)$
- 3 $y = (0.5 - (\text{Min}(\text{Max}(\hat{j}, 0), mapsize - 1)/mapsize))$
- 4 $\varphi = 90 - 360(\text{Atan}(e^{-2\pi y})/\pi)$
- 5 $\lambda = 360x$
- 6 **regresar** (φ, λ)

Algoritmo 3: Conversión de coordenadas de imagen a geográficas

Con el Algoritmo 3 ya tenemos las coordenadas geográficas $\langle \varphi, \lambda \rangle$ correspondientes a cada coordenada de imagen $\langle \hat{i}, \hat{j} \rangle$, con las ecuaciones 5.2 y 5.3 podríamos calcular a qué pixel mapea cada una de las DA que calculamos previamente con el algoritmo directo de Vincenty en coordenadas geográficas. El siguiente paso es aplicar este algoritmo para mejorar nuestras imágenes de FDP que propusimos en el Capítulo 4 de este trabajo, georeferenciando cada uno de los puntos de la imagen de la FDP.

5.5. Corrección de errores de graficación de DA y FDP

La mejora al proceso de graficación presentado en el Capítulo 3 de este trabajo se detalla a continuación, aplicando las ecuaciones previamente descritas y definidas en la documentación de los mapas de Microsoft [BIN10].

Recordemos que la localización geográfica de la antena es la referencia para definir las coordenadas polares $\langle r, \theta \rangle$ de cada DA, entonces definiremos $\langle \varphi_0, \lambda_0 \rangle$ como el punto inicial o centro de referencia (posición de la antena) y al conjunto $\{\langle r_1, \theta_1 \rangle, \langle r_2, \theta_2 \rangle, \dots, \langle r_N, \theta_N \rangle\}$ como el conjunto de coordenadas polares con respecto a la posición de la antena. Con estos datos podemos definir el Algoritmo 4 para graficar una DA en coordenadas $\langle r, \theta \rangle$ dados el centro de referencia $\langle \varphi_0, \lambda_0 \rangle$, un nivel de acercamiento Z y la imagen I de tamaño $nr \times nc$ donde se graficarán los puntos de la siguiente manera:

GRAFICAVINCENTY($\varphi_0, \lambda_0, r, \theta, Z, I, nr, nc$)

- 1 Calcular i_0 utilizando 5.2 con λ_0 y Z como argumentos
- 2 Calcular j_0 utilizando 5.3 con φ_0 y Z como argumentos
- 3 Calculamos $\varphi_{DA}, \lambda_{DA}$ aplicando Algoritmo 6 con $r, \theta, \varphi_0, \lambda_0$
- 4 Calcular i_{DA} utilizando 5.2 con λ_{DA} y Z como argumentos
- 5 Calcular j_{DA} utilizando 5.3 con φ_{DA} y Z como argumentos
- 6 $\hat{x}_i = (i_{DA} - i_0) + \frac{nc}{2}$
- 7 $\hat{y}_i = (j_{DA} - j_0) + \frac{nr}{2}$
- 8 Marcar el pixel de la imagen haciendo $I[\hat{y}_i, \hat{x}_i] = 0$, Si $0 \leq \hat{x}_i < nc$ y $0 \leq \hat{y}_i < nr$
- 9 **regresar** (imagen marcada I)

Algoritmo 4: Graficación de DA's aplicando algoritmos de Vincenty

Si calculamos las coordenadas en pixeles equivalentes aplicando el algoritmo an-

terior, para cada una de las DA detectadas por la antena y asociamos dicho vector a la variable u , podemos modificar el Algoritmo 2 de la sección 4.3 de este trabajo, mediante las ecuaciones descritas en el presente capítulo mediante el Algoritmo 5

```

GRAFICAFDPVINCENTY( $\varphi_0, \lambda_0, Z, nr, nc, I, u$ )
1  Calcular  $i_0$  utilizando 5.2 con  $\lambda_0$  y  $Z$  como argumentos
2  Calcular  $j_0$  utilizando 5.3 con  $\varphi_0$  y  $Z$  como argumentos
3  para  $\hat{y} \leftarrow i_0 - \frac{nr}{2}$  hasta  $i_0 + \frac{nr}{2}$ 
4    para  $\hat{x} \leftarrow j_0 - \frac{nc}{2}$  hasta  $j_0 + \frac{nc}{2}$ 
5      Calcular  $p(v)$  con  $v = [\hat{x}, \hat{y}]^T$  utilizando 4.8 o 4.9
6      Hacer  $I[\hat{y}, \hat{x}] = p(v)$ 
7      Hacer max igual al Máximo de I
8      Hacer min igual al Mínimo de I
9    para  $\hat{y} \leftarrow i_0 - \frac{nr}{2}$  hasta  $i_0 + \frac{nr}{2}$ 
10   para  $\hat{x} \leftarrow j_0 - \frac{nc}{2}$  hasta  $j_0 + \frac{nc}{2}$ 
11     Hacer  $I[\hat{y}, \hat{x}] = 255 * (I[\hat{y}, \hat{x}] - min) / (max - min)$ 
12 regresar imagen marcada I

```

Algoritmo 5: Graficación de FDP sobre una imagen aplicando algoritmos de Vincenty

Al implementar los algoritmos mostrados en este capítulo fuimos capaces de obtener la misma exactitud gráfica que las herramientas proporcionadas por la compañía Google para graficar marcadores georeferenciados. Recordemos que para graficar un marcador es necesario especificar la coordenada geográfica donde queremos graficar nuestro marcador. Para probar la exactitud de los métodos de graficación aquí comentados, realizamos los siguientes experimentos. Calculamos las coordenadas de seis puntos geográficos distanciados equitativamente sobre el planeta y los graficamos con marcadores verdes de Google

partiendo de la localización de la antena como punto inicial. Con la implementación de los algoritmos presentados en este capítulo calculamos las coordenadas de imagen correspondientes a los seis puntos geográficos, tomando como parámetros el nivel de acercamiento y las coordenadas geográficas de los puntos que calculamos; dibujamos puntos rojos bajo la posición de los marcadores de globo. Si observamos la Figura 5.2 tenemos que si la distancia entre los puntos es muy grande, los puntos que nosotros calculábamos no coincidían con los marcadores de Google. En la Figura 5.4 mostramos la corrección de dichos errores comparando los marcadores de Google en color verde contra los puntos en rojo que nosotros graficamos exactamente debajo de los marcadores.

De esta forma tenemos una representación georeferenciada de la FDP del conjunto de DA de nuestro interés graficada sobre mapas estáticos, recordando que esto es válido tanto para los mapas implementados por Google, como para los implementados por Microsoft.

5.6. Conclusiones

La aplicación de geometría plana para la localización de puntos geográficos solo es válida para distancias que no sobrepasan las decenas de kilómetros, por lo tanto, para distancias mayores es necesario aplicar geometría elíptica para el cálculo de dichas distancias. En este capítulo se corrigieron los errores de precisión acarreados por las suposiciones hechas a lo largo de los primeros cuatro capítulos. Se mostró la manera correcta de graficar una coordenada geográfica sobre un mapa georeferenciado a diferentes niveles de acercamiento. Aplicamos el algoritmo directo diseñado por Theodore Vincenty publicado en el artículo "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations", con el cual pudimos pasar de las coordenadas polares $\langle r, \theta \rangle$ de cada DA a coordenadas geográficas $\langle \varphi, \lambda \rangle$ para cada una de las DA, dando como resultado un conjunto de puntos georeferenciados de nuestras DA.

Además utilizamos la formulación definida en la documentación de los mapas BING [BIN10], lo cual nos permite calcular a qué coordenada de imagen $\langle \hat{i}, \hat{j} \rangle$ de nuestro mapa corresponde un punto con coordenadas geográficas $\langle \varphi, \lambda \rangle$ a un determinado nivel de acer-

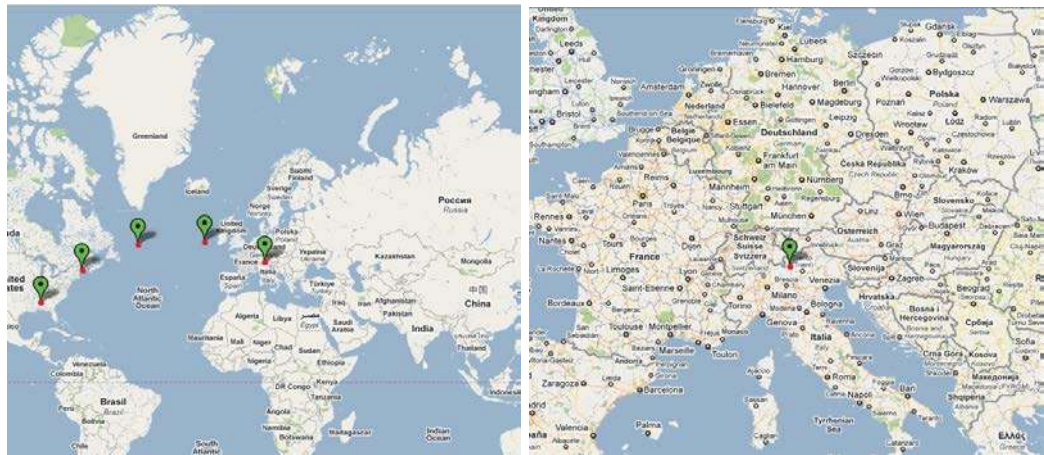
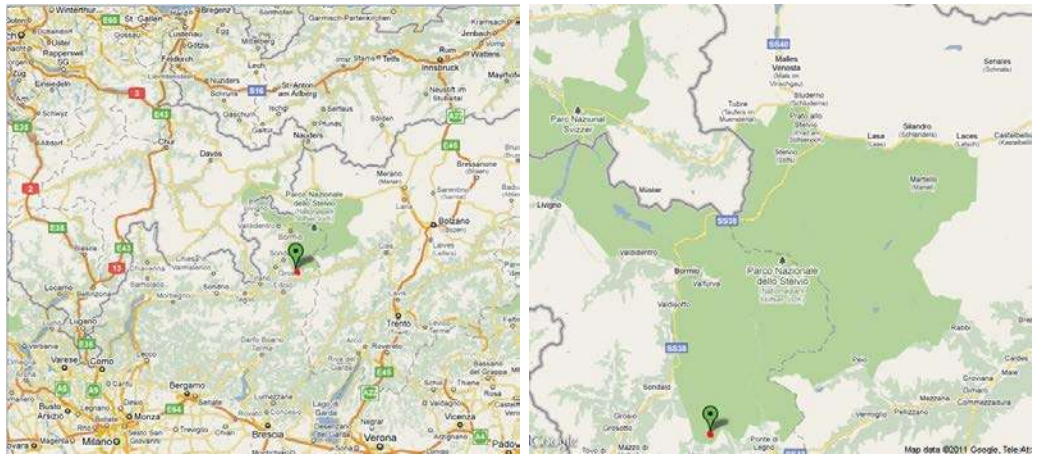
(a) $Z = 2$ (b) $Z = 5$ (c) $Z = 8$ (d) $Z = 10$

Figura 5.4: Resultados del algoritmo de graficación de imágenes georeferenciadas a diferentes niveles de acercamiento.

camiento Z . De esta forma podemos mapear cada coordenada $\langle \varphi, \lambda \rangle$ de cada una de las DA a los mapas estáticos que hemos estado utilizando a lo largo de este trabajo.

Aplicando todas estas herramientas presentadas a lo largo de este capítulo podemos crear imágenes georeferenciadas de nuestras DA y FDP.

Capítulo 6

Pruebas y resultados

Con nuestro servidor en marcha y teniendo empaquetada nuestra aplicación, se muestra la implementación tanto de una representación puntual como la estimación de la FDP de las mismas DA's en la Figura 6.1. En la Figura 6.1 a) se grafican de manera puntual las DA detectadas entre el 01/01/2010 y el 01/03/2010 y en la Figura 6.1 b) mostramos la implementación de nuestra aplicación web. De este par de figuras podemos deducir lo siguiente:

En la figura de la representación puntual de las descargas no podemos distinguir cuál es el área de mayor incidencia de DA's, tan solo nos da una idea de donde se detectaron tales eventos. La gráfica de la estimación de la FDP nos muestra que el área de mayor incidencia de DA es cerca de la ciudad de Morelia, representado por las zonas en color rojo sobre el mapa.

Una mejora importante con respecto a la implementación de mapas estáticos fue el uso de imágenes en formato PNG para agregar efectos de transparencia en las imágenes de la FDP, esto para mantener la visibilidad de la información presentada en los mapas de Google tales como nombre de ciudades, vialidades, terreno, etc. En la Figura 6.2 trabajamos con las DA detectadas entre el día 01/05/2011 y el 11/05/2011, primeramente graficando de manera puntual la localización de las DA como se muestra en la Figura 6.2 a). En la Figura 6.2 b) aplicamos ventanas de parzen gaussianas sobre los mismos eventos detectados al mismo nivel de acercamiento, notemos que a este nivel es difícil distinguir cuáles son

las zonas de mayor incidencia de DA, pero finalmente en la imagen 6.2 c) realizamos un acercamiento más a detalle mediante la aplicación WEB desarrollada y podemos distinguir claramente tres zonas de color rojo de mayor incidencia de DA, cabe mencionar que el uso de transparencias en el archivo PNG de la imagen de la FDP permitió mostrar la información geográfica del mapa lo cual hubiera requerido de la graficación de curvas de nivel de no haber aplicado transparencia en la imagen.

6.1. Conclusiones

Sin la ayuda de las herramientas que Google o Microsoft pone a nuestro alcance para la implementación de mapas georeferenciados este trabajo final necesitaría de mucho más trabajo para una visualización ágil y rápida de los resultados calculados con los algoritmos presentados en esta tesis. Quiero hacer énfasis en el hecho de que la parte principal de este trabajo no es el uso de mapas georeferenciados, pero sin ellos no hubiera sido posible dar una idea clara de la posición geográfica de los resultados calculados en este trabajo.

Existe una gran teoría acerca de la georeferenciación y el cálculo geodésico, son disciplinas muy complejas que manejan un nivel de cálculo matemático y de geometría elíptica que no se pretende abordar en este trabajo de tesis, sólo mostramos la aplicación muy particular a resolver los problemas relacionados a la georeferencia de los cálculos de la FDP realizados en este trabajo de tesis.

Al final de este trabajo logramos proponer bases y herramientas útiles para el desarrollo de trabajos gráficos georeferenciados y dimos una breve introducción a las grandes herramientas que son los mapas georeferenciados tanto de la compañía Google como Microsoft.

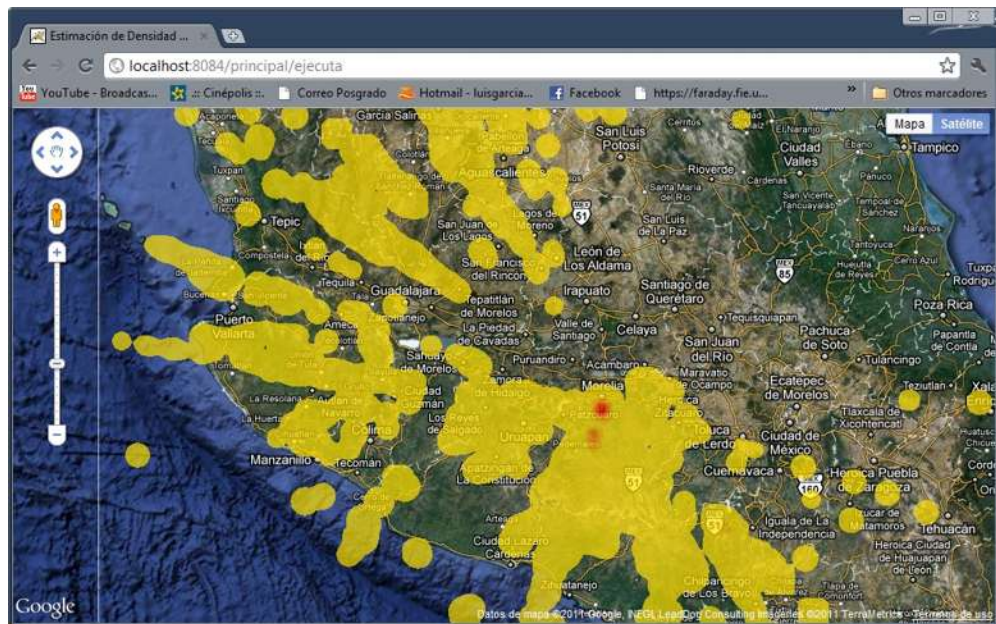
(a) *Gráfica puntual de DA*(b) *Estimación de la FDP*

Figura 6.1: Comparativa entre la gráfica puntual y la estimación de la FDP de las DA detectadas entre el 01/01/2010 y el 01/03/2010.

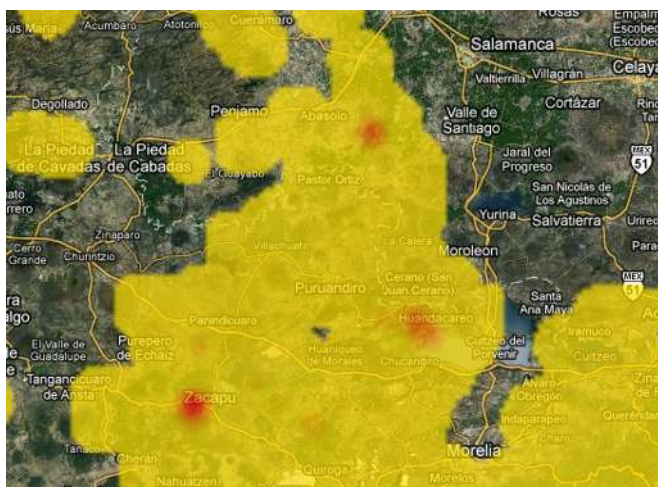
(a) *Gráfica puntual de DA*(b) *Ventanas de Parzen Gaussianas*(c) *Identificación de zonas de mayor incidencia de DA*

Figura 6.2: DA detectadas entre las fechas 01/05/2011 y 11/05/2011

Capítulo 7

Conclusiones

7.1. Conclusiones generales

Con este trabajo mostramos la gran ventaja que resulta el estimar la Función de Densidad de Probabilidad de descargas atmosféricas para evaluar gráficamente las zonas de mayor incidencia de descargas sobre una región geográfica. Es clara la ventaja de usar estas gráficas de densidad con respecto a una representación puntual de las descargas atmosféricas. Las gráficas de densidad de probabilidad aquí presentadas son georeferenciadas utilizando los algoritmos diseñados por Theodore Vincenty [Vincenty75], los cuales son ampliamente aplicados en la rama de la Cartografía y Posicionamiento Global (GPS).

El trabajo que aquí se presenta permite al lector entender los principios básicos referentes al cálculo y georeferenciación de puntos geográficos utilizando cálculos básicos de geometría elíptica. Aunque no es un trabajo especializado en el tema de georeferenciación, considero que presenta las bases para generar interés y trato de presentar la utilización de mapas web de manera práctica y útil para un lector con conocimientos básicos de geometría y cálculo. Los mapas proporcionados por las compañías Google y Microsoft son una gran herramienta de libre acceso las cuales utilizamos para presentar geográficamente los resultados estimados de Densidad de Probabilidad. Se da una introducción sencilla al uso de mapas tanto estáticos como dinámicos, así como las funciones principales para mapear coordenadas geográficas en coordenadas de la imagen de los mapas antes mencionados.

Por último, creo que con este trabajo presentamos una solución de buena calidad y con bases matemáticas para el estudio de la incidencia de las Descargas Atmosféricas en la región del Occidente, abordando un interés real planteado por la empresa CFE.

7.2. Trabajos futuros

1. Hasta el día de hoy, ya se tienen almacenados eventos atmosféricos por alrededor de 3 años. Se tiene la propuesta de aplicar técnicas de predicción de eventos tales como los modelos ocultos de Markov.
2. Probar con otras técnicas de estimación de la FDP como el método de k-vecinos mas cercanos.
3. Se tiene planeado con la CFE-DCO la instalación de mas antenas ubicadas en lugares estratégicos, esto para corroborar la correcta medición de las DA's censadas.
4. El Centro Nacional de Control de Energía (CENACE) tiene interés en poder tener mapas georeferenciados con la localización de sus instalaciones y poder observar en tiempo real el comportamiento de fenómenos atmosféricos que pudieran presentar un riesgo para la integridad de sus equipos.

Apéndice A

Algoritmos de cálculo geodésico

El planeta Tierra no es una figura geométrica perfecta. Las primeras suposiciones acerca de la forma de la Tierra suponían que tenía forma esférica y no estaban muy lejos de la realidad. Hoy en día, la aproximación más usada utiliza una elipsoide de revolución descrita por las siguientes ecuaciones paramétricas:

$$x = a \operatorname{Cos}(\theta) \tag{A.1}$$

$$y = b \operatorname{Sen}(\theta) \tag{A.2}$$

donde a y b son el semieje horizontal y vertical de nuestra elipsoide, como se puede ver en la Figura A.1.

En el sistema de coordenadas WGS84 [Loweth97] se define $a = 6378137$ y $b = 6356752.3142$ como semiejes mayor y menor respectivamente para la representación del planeta. Recordemos que el sistema WGS84 es el utilizado para la georeferenciación de puntos geográficos implementados en los dispositivos GPS.

Mediante identidades trigonométricas es posible calcular una circunferencia equivalente con lo cual se facilitan los cálculos referentes a distancias geodésicas. Primeramente definamos un punto M con componentes latitud y longitud que definen su posición sobre

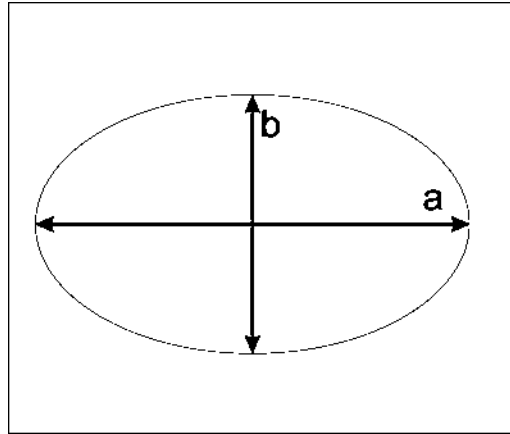


Figura A.1: Elipsoide y sus semiejes mayor (a) y menor (b)

la superficie de la Tierra, además tendremos un radio vector que es la distancia entre el centro de la Tierra y el punto M , con estas dos sencillas definiciones podemos deducir las siguientes componentes equivalentes de Latitud:

Latitud Geocéntrica ω : Es el ángulo que forma el radio vector con el plano del ecuador (Figura A.2).

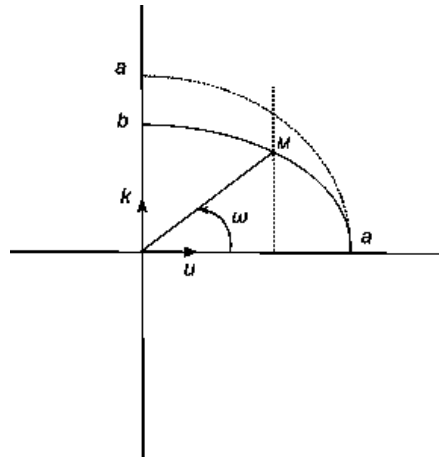


Figura A.2: Latitud Geocéntrica Ω

Latitud Geográfica o Geodésica φ : Es el ángulo que forma la vertical del punto

M con el plano horizontal, como se observa en la Figura A.3. Este tipo de coordenada es el utilizado en el sistema WGS84 utilizado para la georeferencia en dispositivos GPS y para hacer referencia a puntos geográficos en los mapas que hemos utilizado a lo largo de este trabajo (Google Maps y Bing Maps de Microsoft).

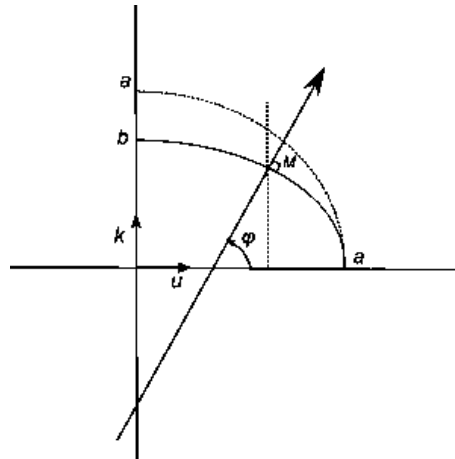


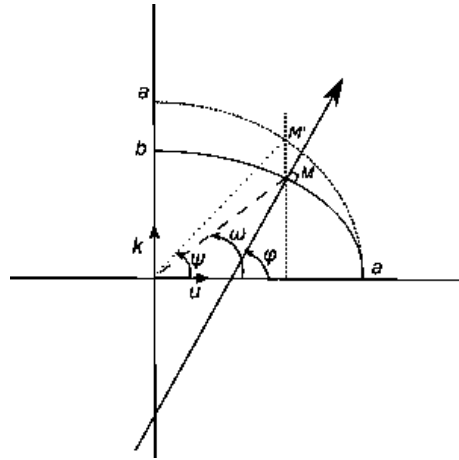
Figura A.3: Latitud Geodésica

Latitud Reducida o Excéntrica Ψ : Para definirla se traza una semicircunferencia de radio igual al semieje horizontal y se pasa por M una perpendicular al semieje mayor a que corta a la semicircunferencia en M' . Si unimos M' con el centro de la elipse vemos que esta línea corta al plano con un ángulo ψ que es la latitud reducida mostrada en la Figura A.4.

Los tres tipos de coordenadas definidos se relacionan entre sí mediante las siguientes ecuaciones [E. J. Krakiwsky73]:

$$\text{Tan}(\Psi) = \frac{b}{a}\text{Tan}(\Omega) \quad (\text{A.3})$$

$$\text{Tan}(\Psi) = \frac{a}{b}\text{Tan}(\Omega) \quad (\text{A.4})$$

Figura A.4: Latitud Reducida ψ

$$\text{Tan}(\Omega) = \frac{b^2}{a^2} \text{Tan}(\omega) \quad (\text{A.5})$$

Con la ecuación A.3 podemos calcular las coordenadas reducidas equivalentes a las coordenadas geodésicas correspondientes a nuestros puntos geográficos sobre la superficie de la Tierra y de esta manera facilitar el cálculo geodésico, estos cálculos son la base para el cálculo de geodésicas ya que con estas equivalencias podemos pasar de un sistema de coordenadas sobre una elipse a calcular un equivalente de coordenadas sobre un círculo. Estas ecuaciones son necesarias para la aplicación de los algoritmos presentados en el Anexo A.

A.1. Fórmula Directa de Vincenty

En el artículo publicado por Theodore Vincenty titulado " Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations " [Vincenty75] en Abril de 1975, se propone una solución al cálculo de geodésicas. El algoritmo propuesto por Vincenty de manera directa resuelve el problema de calcular un segundo punto con coordenadas geográficas

$\langle \varphi_2, \lambda_2 \rangle$, recibiendo como parámetros un primer punto con coordenadas geográficas $\langle \varphi_1, \lambda_1 \rangle$, la distancia geodésica s que existe entre los dos puntos, y el ángulo α_1 de salida del punto 1 hacia el punto 2.

Así mismo, Theodore Vincenty propone un algoritmo inverso, el cual resuelve el problema de calcular la distancia geodésica entre dos puntos geográficos, recibiendo como argumentos las coordenadas geográficas de los dos puntos sobre la superficie terrestre. El Algoritmo 6 calcula un segundo punto p_2 en coordenadas geográficas $\langle \varphi_2, \lambda_2 \rangle$ dado el primer punto p_1 con coordenadas $\langle \varphi_1, \lambda_1 \rangle$, la distancia geodésica s entre los dos puntos y el ángulo α_1 que se forma entre la línea geodésica y una línea imaginaria que apunta al norte geográfico.

A.2. *Fórmula Inversa de Vincenty*

La formulación inversa del algoritmo de Vincenty para el cálculo de distancia geodésica s entre dos puntos geográficos p_1 y p_2 con coordenadas $\langle \varphi_1, \lambda_1 \rangle$ y $\langle \varphi_2, \lambda_2 \rangle$ se implementa en el Algoritmo 7

El algoritmo inverso de Vincenty calcula la distancia geodésica s entre los puntos p_1 y p_2 , así como los ángulos de salida $\langle \alpha_1 \rangle$ y $\langle \alpha_2 \rangle$ para ir de p_1 a p_2 y de p_2 a p_1 respectivamente.

VINCENTYDIRECTO($\varphi_1, \lambda_1, s, \alpha_1$)

- 1 $f = \frac{a-b}{a}$
- 2 $\tan(\psi_1) = (1-f)\tan(\varphi_1)$ aplicando la identidad de la ecuación A.3.
- 3 $\cos(\psi_1) = 1/\sqrt{1+\tan^2(\psi_1)}$ por la identidad trigonométrica $\tan^2(x) + 1 = 1/\cos^2(x)$
- 4 $\sin(\psi_1) = \tan(\psi_1)\cos(\psi_1)$ por la identidad trigonométrica $\tan(x) = \sin(x)/\cos(x)$
- 5 $\sigma_1 = \text{atan2}(\tan(\psi_1), \cos(\alpha_1))$
- 6 $\sin(\alpha) = \cos(\psi_1)\sin(\alpha_1)$
- 7 $\cos^2(\alpha) = 1 - \sin^2(\alpha)$ por la identidad trigonométrica $\cos^2(x) + \sin^2(x) = 1$
- 8 $u^2 = \cos^2(\alpha) \frac{a^2 - b^2}{b^2}$
- 9 $A = 1 + u^2/16384\{4096 + u^2[-768 + u^2(320 - 175u^2)]\}$
- 10 $B = u^2/1024\{256 + u^2[-128 + u^2(74 - 47u^2)]\}$
- 11 $\sigma = s/b.A$ (1er aproximación) , $\sigma' = 2\pi$
- 12 **mientras** $\text{abs}(\sigma - \sigma') > 10^{-12}$
- 13 $\cos(2\sigma_m) = \cos(2\sigma_1 + \sigma)$
- 14 $\Delta\sigma = B.\sin(\sigma)\{\cos(2\sigma_m) + B/4[\cos(\sigma)(-1 + 2\cos^2(2\sigma_m)) - B/6\cos(2\sigma_m)(-3 + 4\sin^2(\sigma))(-3 + 4\cos^2(2\sigma_m))]\}$
- 15 $\sigma' = \sigma$
- 16 $\sigma = s/b.A + \Delta\sigma$
- 17 $\varphi_2 = \text{atan2}(\sin(\psi_1)\cos(\sigma)\cos(\alpha_1), (1-f)\sqrt{\sin^2(\alpha) + (\sin(\psi_1)\sin(\sigma) - \cos(\psi_1)\cos(\sigma)\cos(\alpha_1))^2})$
- 18 $\lambda = \text{atan2}(\sin(\sigma)\sin(\alpha_1), \cos(\psi_1)\cos(\sigma) - \sin(\psi_1)\sin(\sigma)\cos(\alpha_1))$
- 19 $C = \frac{f}{16\cos^2(\alpha)[4+f(4-3\cos^2(\alpha))]}$
- 20 $L = \lambda - (1-C).f.\sin(\alpha\{\sigma + C.\sin(\sigma)[\cos(2\sigma_m) + C.\cos(\sigma)(-1 + 2\cos^2(2\sigma_m))]\})$
- 21 $\alpha_2 = \text{atan2}(\sin(\alpha), -\sin(\psi_1)\sin(\sigma) + \cos(\psi_1)\cos(\sigma)\cos(\alpha_1))$
- 22 **regresar** $p_2(\varphi_2, \lambda_1 + L)$

Algoritmo 6: Fórmula Directa de Vincenty

VINCENTYINVERSO($\varphi_1, \lambda_1, \varphi_2, \lambda_2$)

1 $L = (\lambda_2 - \lambda_1)$

2 $\psi_1 = \text{atan}((1 - f)\tan(\varphi_1))$

3 $\psi_2 = \text{atan}((1 - f)\tan(\varphi_2))$

4 $\lambda = L$ primera aproximación

5 **repetir**

6 $\sin(\sigma) = \sqrt{(\cos(\psi_2)\sin(\lambda))^2 + (\cos(\psi_1)\sin(\psi_2) - \sin(\psi_1)\cos(\psi_2)\cos(\lambda))^2}$

7 $\cos(\sigma) = \sin(\psi_1)\sin(\psi_2) + \cos(\psi_1)\cos(\psi_2)\cos(\lambda)$

8 $\sigma = \text{atan2}(\sin(\sigma), \cos(\sigma))$

9 $\sin(\alpha) = \cos(\psi_1)\cos(\psi_2)\sin(\lambda)/\sin(\sigma)$

10 $\cos^2(\alpha) = 1 - \sin^2(\alpha)$

11 $\cos(2\sigma_m) = \cos(\sigma) - 2\sin(\psi_1)\sin(\psi_2)/\cos^2(\alpha)$

12 $C = f/16\cos^2(\alpha)[4 + f(4 - 3\cos^2(\alpha))]$

13 $\lambda' = \lambda$

14 $\lambda = L + (1 - C)f\sin(\alpha)\{\sigma + C\sin(\sigma)[\cos(2\sigma_m) + C\cos(\sigma)(-1 + 2\cos^2(2\sigma_m))]\}$

15 **hasta** $|\lambda - \lambda'| > 10^{-12}$

16 $u^2 = \cos^2(\alpha)(a^2 - b^2)/b^2$

17 $A = 1 + u^2/16384\{4096 + u^2[-768 + u^2(320 - 175u^2)]\}$

18 $B = u^2/1024\{256 + u^2[-128 + u^2(74 - 47u^2)]\}$

19 $\Delta\sigma = B.\sin(\sigma)\{\cos(2\sigma_m) + B/4[\cos(\sigma)(-1 + 2\cos^2(2\sigma_m)) - B/6\cos(2\sigma_m)(-3 + 4\sin^2(\sigma)(-3 + 4\cos^2(2\sigma_m))]\}$

20 $s = b.A.(\sigma - \Delta\sigma)$

21 $\alpha_1 = \text{atan2}(\cos(\psi_2)\sin(\lambda), \cos(\psi_1)\sin(\psi_2) - \sin(\psi_1)\cos(\psi_2)\cos(\lambda))$

22 $\alpha_2 = \text{atan2}(\cos(\psi_1)\sin(\lambda), \sin(\psi_1)\cos(\psi_2) + \sin(\psi_2)\cos(\psi_1)\cos(\lambda))$

23 **regresar** s, α_1, α_2

Algoritmo 7: Fórmula Inversa de Vincenty

Apéndice B

APACHE TOMCAT sobre WINDOWS

La antena del proyecto SISTORM se encuentra instalada en una computadora personal con SO Windows XP y cabe señalar que al día de hoy, no hay soporte de otro SO para el DRIVER de la tarjeta de adquisición de datos de la antena y por este motivo se tomó la decisión de instalar un servidor APACHE-TOMCAT en WINDOWS, para la máquina que funciona de servidor del proyecto.

Para instalar un servidor APACHE-TOMCAT en WINDOWS es necesario tener instalado previamente el JDK o JRE actualizado y será de importancia conocer la ruta donde se instaló dicho paquete de JAVA. Lo siguiente es descargar el servidor APACHE-TOMCAT para nuestro SO Windows desde la página del desarrollador [APA11]. En nuestro caso decidimos utilizar la versión 6.0.26 para WINDOWS ya que será requerida mas adelante al momento de configurar el servidor. Existen diferentes paquetes de descarga dependiendo de las necesidades que tengamos, en nuestro caso, para desarrollo y realizar pruebas sobre el funcionamiento del servidor, descargamos el archivo comprimido ZIP para windows versión CORE del listado BINARY DISTRIBUTIONS dentro del apartado de DOWNLOADS.

El siguiente paso es descomprimir el archivo ZIP y colocar su contenido en el directorio donde se encuentran los archivos de programas, en nuestro caso particular en la ruta C:/Program Files/ y le damos los permisos necesarios de lectura y escritura sobre la carpeta.

El paquete que descargamos contiene entre otros los siguientes directorios de importancia:

1. /bin - Arranque, cierre y otros scripts ejecutables.
2. /temp - Archivos temporales.
3. /conf - Archivos XML y los correspondientes DTD para la configuración de apache-tomcat.
4. /logs - Archivos de registro (log) de apache-tomcat.
5. /webapps - Directorio que contiene las aplicaciones web.
6. /work - Almacenamiento temporal de ficheros y directorios.

El siguiente paso y uno de los más importantes es declarar la variable de entorno `JAVA_HOME` con la ruta del JDK o `JRE_HOME` en el caso de JRE, instalado previamente, es necesario reiniciar la computadora después de declarar dicha variable para que surtan efecto los cambios. En la figura B.1 se muestra la variable declarada en nuestra máquina en particular.

Si realizamos correctamente los pasos anteriores, nos dirigimos a la ruta donde

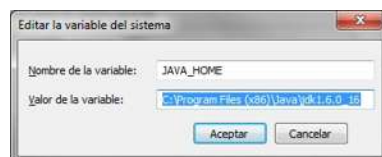


Figura B.1: Ejemplo de declaración de variable.

descomprimos los archivos del apache-tomcat y en la subcarpeta /bin ejecutamos el archivo startup.bat en windows y esto nos arrancará nuestro servidor.

Para comprobar si realizamos correctamente estos pasos, abrimos nuestro navegador web y escribimos localhost:8080 ya que el puerto 8080 es el utilizado por default, y debemos poder observar en la pantalla una página de bienvenida de apache-tomcat como se muestra en la figura B.2.

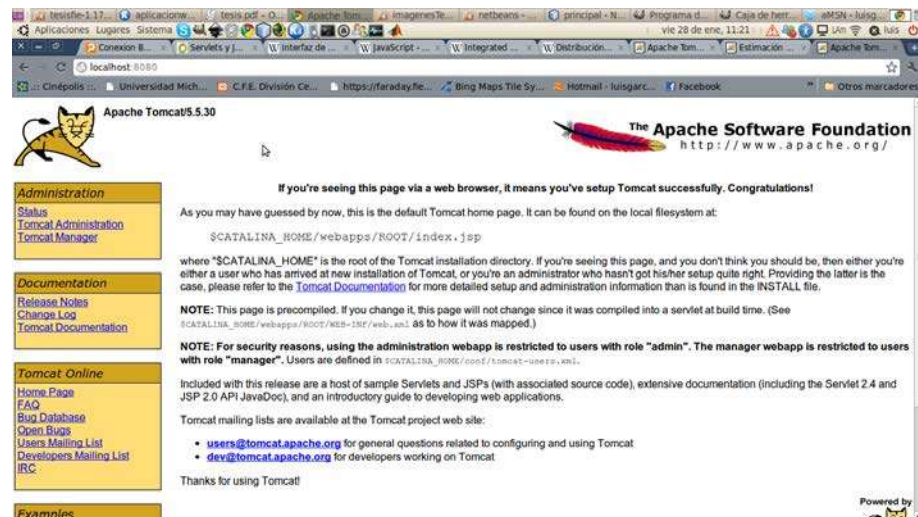


Figura B.2: Página de Bienvenida de APACHE-TOMCAT

Apéndice C

Diseño de Aplicación WEB para desplegar mapas

Uno de los intereses de la CFE-DCO es el poder observar los resultados gráficos de la estimación de la FDP a través de una aplicación WEB. Para esto, JAVA tiene la característica de proporcionar una alta interacción con aplicaciones WEB. Se propuso desarrollar una aplicación WEB que contará con una página para visualizar la estimación de la FDP dentro de las fechas seleccionadas por el usuario, todo esto graficado sobre mapas dinámicos de Google.

Nosotros creamos y guardamos las imágenes georeferenciadas de la estimación de la FDP en la máquina servidor y después las “pegamos” sobre los mapas estáticos. Para esto es necesario crear nuevos archivos de imagen cada vez que se solicita una nueva estimación de la FDP; estos archivos se crean en la máquina servidor. De entrada, JAVA limita la manipulación de los archivos del servidor a través de una página WEB desde una máquina cliente por cuestiones de seguridad; esto significa que no podemos modificar o alterar los recursos que no pertenecen a la máquina donde se ejecuta la aplicación. Nosotros proponemos desarrollar nuestra aplicación WEB utilizando JSP para la creación de una aplicación cliente-servidor la cual se ejecutaría en la máquina servidor y desplegaría los resultados en la máquina cliente.

C.1. Java Server Pages (JSP)

JSP es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML, o Javascript. JSP puede considerarse como una manera alternativa y simplificada de construir servlets, es por ello que una página JSP puede hacer todo lo que un servlet puede hacer, y viceversa. Podemos pensar en un SERVLET como un APPLETT que corre en el lado del servidor, teniendo de esta manera la facilidad de interactuar con los recursos del mismo y nos permite trabajar sobre diferentes tipos de servidores de aplicaciones tales como APACHE y GLASSFISH. El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del SERVLET a generar. Este SERVLET será el que genere el documento (típicamente HTML) que se presentará en la pantalla del navegador del usuario.

De esta forma podemos cargar nuestra aplicación mediante JSP y ejecutarlo del lado del servidor. Además, si los datos referentes a las DA son almacenados en una base de datos, por ejemplo MySQL, el proyecto de aplicación podría definirse gráficamente como se muestra en la Figura C.1. Podemos observar en la Figura que tanto la consulta a los datos como la ejecución de la aplicación JSP se ejecutarán en la máquina servidor, y sólo se desplegarán los mapas con la información de la FDP en las máquinas cliente que se conecten a la aplicación. Los clientes pueden residir sobre diferentes plataformas tales como MacOS, Windows o Linux.

Hasta este punto del trabajo de tesis sólo hemos hecho mención de los mapas estáticos tanto de la compañía Microsoft como de Google, pero otro tipo de mapas ampliamente desarrollados y utilizados (aún mas que los mapas estáticos) son los dinámicos, por ejemplo los utilizados en el servicio de mapas llamado Google Maps, ofrecido por la compañía Google. Google Maps es una

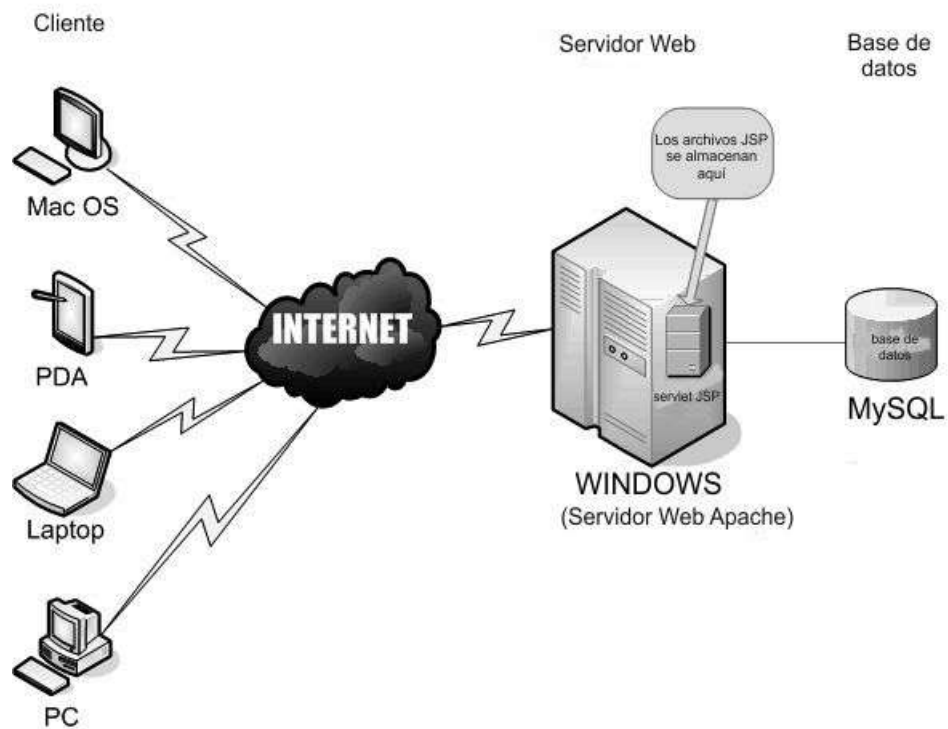


Figura C.1: Proyecto de aplicación WEB con JSP

servicio WEB de libre acceso que nos permite, entre otras cosas, localizar lugares o puntos geográficos de diferentes maneras. Por ejemplo, dando las coordenadas geográficas del lugar, o dando una descripción del lugar de interés. Google ofrece la documentación necesaria para implementar este tipo de servicios mediante su API en lenguaje JAVASCRIPT. Nosotros hicimos uso de los ejemplos de código a los que se tiene libre acceso en la documentación de Google, pero la parte referente a la creación de la imagen de la FDP y la georeferenciación de la misma sigue siendo implementación propia. En resumen, solo utilizamos los mapas proporcionados como fondo de nuestra imagen de FDP y como referencia geográfica de los eventos detectados. Lo que proponemos es crear nuestras imágenes de FDP con los métodos implementados en las secciones anteriores y “pegar” esta imagen sobre los mapas proporcionados por Google o Microsoft. En la Figura C.2 se muestra uno de los ejemplos que Google publica [OVE11]

en su documentación para JAVASCRIPT, este ejemplo nos muestra cómo fijar una imagen, en formato JPG o PNG entre otros, sobre un mapa definiendo las coordenadas geográficas donde se desea fijar la imagen. De esta manera, la aplicación escala y desplaza la imagen de tal manera que coincida con el mapa. Si hacemos un acercamiento sobre el mapa, la aplicación escala la imagen y si nos desplazamos sobre el mapa la aplicación implementa el desplazamiento de la imagen automáticamente.

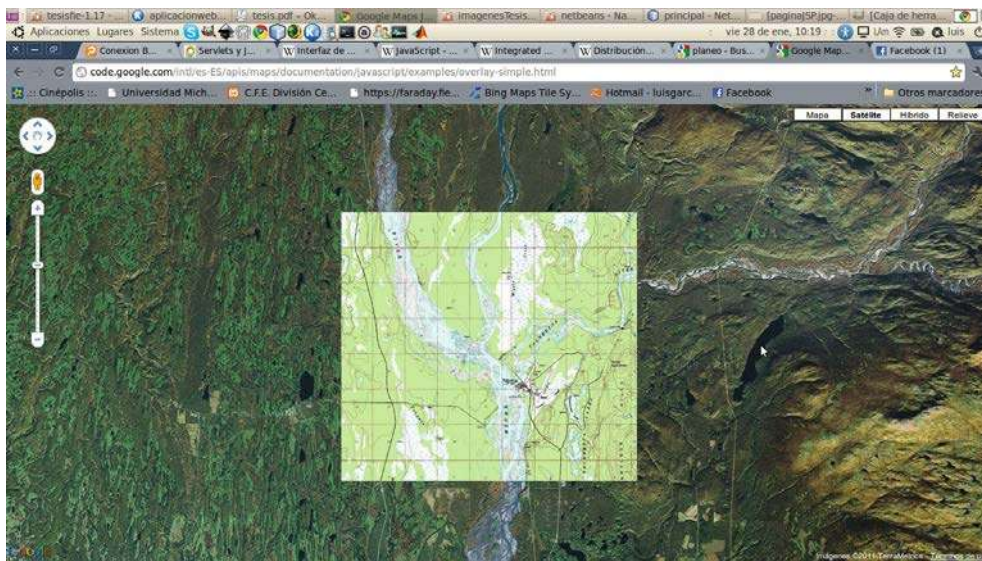


Figura C.2: Ejemplo proporcionado por Google para colocar una imagen fija sobre los mapas

C.2. Desarrollo de aplicación WEB utilizando NETBEANS

Para desarrollar la aplicación utilizamos la herramienta gratuita llamada NETBEANS, la cual es un IDE (Integrated Development Environment) que nos permite desarrollar aplicaciones en diversos lenguajes de programación tales como JAVA, C, PHP, entre otros y además nos ayuda en el desarrollo de aplicaciones WEB tales como JSP. La mejor opción es descargar el paquete de descarga de NETBEANS IDE que ya incluye los servidores GLASSFISH y APACHE TOMCAT [NET11]. En la Figura C.3 se tienen los paquetes de descarga que

están disponibles actualmente para su descarga, así como el tamaño de cada paquete. Una vez instalado el paquete de desarrollo en el sistema operativo de

Paquetes de descarga de NetBeans IDE							
Tecnologías *	Java SE	JavaFX	Java	Ruby	C/C++	PHP	All
NetBeans Platform SDK	•	•	•				•
Java SE	•	•	•				•
JavaFX		•					•
Java Web y EE			•				•
Java ME			•				•
Java Card™ 3 Connected			—				—
Ruby				•			•
C/C++					•		•
Groovy			•				•
PHP						•	•
Servidores incluidos							
GlassFish Server Open Source Edition 3.0.1			•	•			•
Apache Tomcat 6.0.26			•				•
	Download	Download	Download	Download	Download	Download	Download
	Libre, 54 MB	Libre, 116 MB	Libre, 173 MB	Libre, 88 MB	Libre, 35 MB	Libre, 31 MB	Libre, 266 MB

Figura C.3: Paquetes de descarga para NETBEANS

nuestra elección, abrimos el IDE, seleccionamos del menú FILE la opción NEW PROJECT y aquí la opción de WEB APPLICATION como podemos observar en la Figura C.4. Al dar clic en la opción NEXT nos solicitará definir el nombre

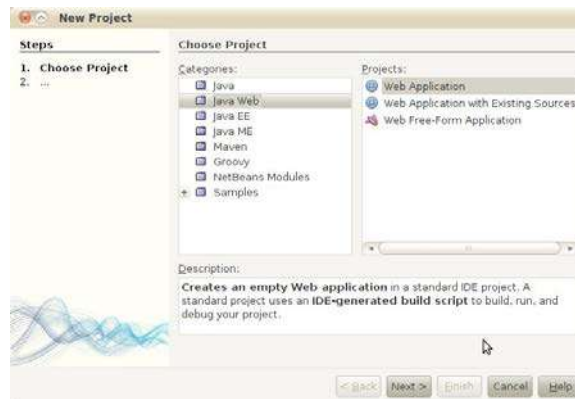


Figura C.4: Selección de tipo de proyecto para NETBEANS

del proyecto que estamos creando, como podemos observar en la Figura C.5. Una vez definido el nombre del proyecto damos clic nuevamente sobre la op-

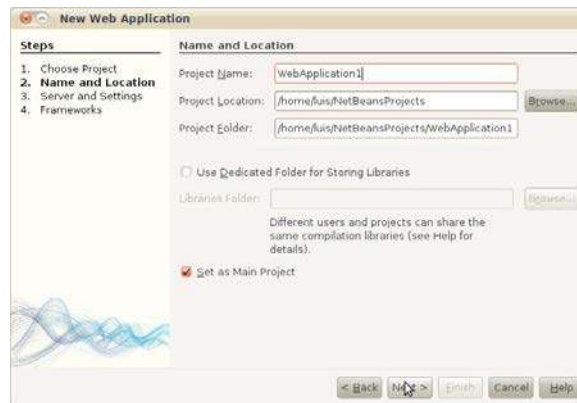


Figura C.5: Selección del nombre de la aplicación WEB con NETBEANS

ción NEXT y tendremos que elegir el servidor para nuestra aplicación (Figura C.6), si descargamos el paquete de desarrollo NETBEANS que recomendamos que ya incluye los servidores disponibles tendremos la opción de seleccionar entre GLASSFISH y TOMCAT, siendo este último el seleccionado para nuestra aplicación particular. Finalmente damos clic sobre la opción FINISH para terminar de definir las propiedades que necesitamos para nuestro proyecto, esto nos generará un proyecto que contiene los archivos necesarios para desarrollar nuestro proyecto, entre ellos un archivo JSP el cual definirá la página principal generada por default para nuestro proyecto.

El archivo index.jsp que NETBEANS genera por default para nuestra aplicación WEB contiene código HTML que al ejecutarlo muestra el clásico ejemplo de “Hola Mundo” en nuestro navegador WEB. Nosotros vamos a necesitar dos archivos primordiales para nuestra aplicación WEB. El primero será este archivo index.jsp donde obtendremos los parámetros necesarios para generar nuestra imagen de la FDP, y el segundo archivo será un archivo JAVA que procesará la información proporcionada por el cuestionario JSP, este archivo JAVA generará la imagen de la FDP, creará un archivo JPG de la FDP y además implementará los mapas dinámicos de Google para fijar sobre ellos nuestro archivo con la FDP. El cuestionario JSP lo podemos crear modificando el contenido del

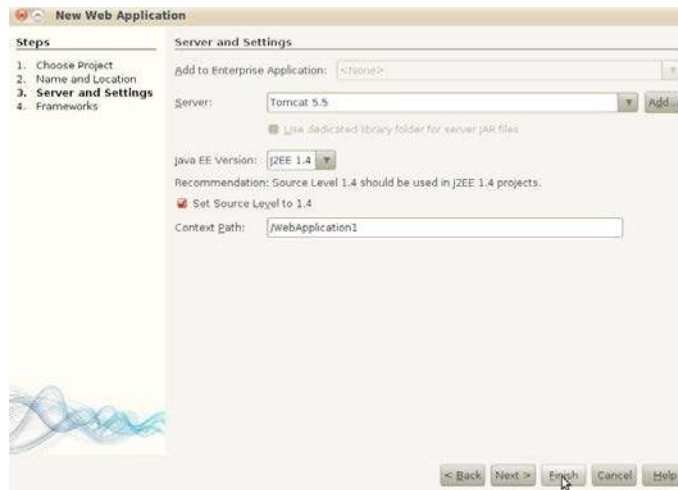


Figura C.6: Selección del servidor de la aplicación WEB con NETBEANS

archivo `Index.jsp` que NETBEANS generó por default al crear nuestro proyecto de aplicación WEB. Podemos editar este archivo JSP desde el mismo editor que contiene NETBEANS, el cual contiene algunas paletas con herramientas que facilitan el diseño de nuestras páginas WEB, por ejemplo podemos agregar botones o campos de texto con tan solo arrastrar los elementos de la paleta al proyecto que estamos diseñando. Nosotros cambiamos el código que se genera por default para tener una página como la que se muestra en la Figura C.7. Para crear una página como la que se muestra en la Figura C.7 solo es necesario investigar sobre el desarrollo de formularios en HTML, Internet ofrece una ininidad de información al respecto. En la Figura C.7 podemos observar que al final del formulario generado tenemos el último de los elementos del formulario, este es un botón “Enviar Consulta” al cual se debe dar clic una vez que se han llenado todos los campos para realizar el cálculo de la FDP. Con este botón se recopilará la información para crear la imagen de la FDP. Debemos especificar al principio de nuestro código HTML, en el archivo `index.jsp`, el nombre del archivo JAVA que procesará la información recopilada por el cuestionario, esto se hace con las siguientes etiquetas en código HTML:

Figura C.7: Página principal del proyecto SISTORM

```
<FORM ACTION="ejecuta" METHOD="POST">
```

Con esta instrucción definimos que el archivo “ejecuta.java” procesará la información recabada por el formulario “index.jsp”, y “POST” es el método por el cual se van a transferir las variables del formulario a la ejecución del código del archivo “ejecuta.java”. Para generar el archivo “ejecuta.java”, sugerimos utilizar el asistente para generar un archivo servlet con NETBEANS, ya que de esta manera se configurará automáticamente la conexión entre el formulario “index.jsp” y el servlet “ejecuta.java”, los pasos a realizar son los siguientes: Dentro de NETBEANS seleccionamos nuestro proyecto “WebApplication1”, y damos clic derecho sobre la carpeta “Source Packages”, después seleccionamos la opción “New” para crear un nuevo archivo de los cuales seleccionaremos la opción “Servlet...” lo cual nos permitirá crear este archivo dentro de la carpeta definiendo el nombre del archivo, en nuestro caso particular lo llamaremos “ejecuta” y tendrá extensión .java. En la Figura C.8 se puede observar la aplicación

de los pasos descritos anteriormente para la creación del archivo servlet nombrado “ejecuta.java”.

El archivo “ejecuta.java” contiene el método `processRequest` generado au-

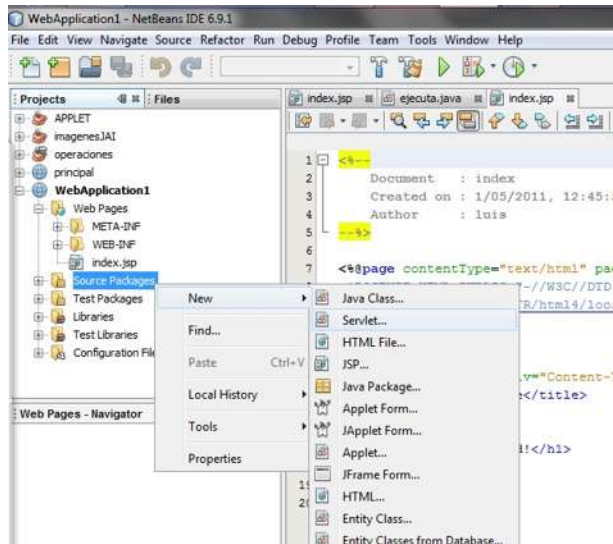


Figura C.8: Creación del servlet nombrado “ejecuta.java”

tomáticamente por NETBEANS que recibe como argumentos dos objetos, uno del tipo `HttpServletRequest` y otro del tipo `HttpServletResponse`. El método `processRequest` es el encargado de ejecutar el código JAVA que definamos dentro del método `processRequest`. Esta parte del código se ejecuta al dar clic sobre el botón de “Enviar Consulta” del cuestionario presentado en la Figura C.7. La manera en que se obtiene la información recabada por el cuestionario `index.jsp` es mediante el siguiente código:

```
String cadena1 = request.getParameter("dia").toString();
```

Lo que obtenemos con este código es el contenido del elemento del cuestionario llamado “dia”, que en nuestro caso particular se trata de un campo de texto. Este valor se transforma al tipo de dato necesario, en nuestro caso particular lo convertimos en una cadena de texto la cual la guardamos en la variable llamada

“cadena1”. De esta forma podemos obtener toda la información necesaria para implementar la estimación de la FDP y crear la imagen de la misma.

Hasta este punto, podemos generar ya la imagen de la FDP con la información obtenida del cuestionario `index.jsp`, la imagen de la FDP la almacenamos en un archivo PNG mediante la siguiente instrucción en JAVA:

```
ImageIO.write(imagenBuffer, "png", ruta_archivo);
```

donde `imagenBuffer` es el objeto de tipo “`BufferedImage`” que contiene a la imagen de la FDP creada con la aplicación de los algoritmos propuestos en la Sección 5.8 de este trabajo. Con el segundo parámetro del método `ImageIO.write` especificamos el tipo de archivo de imagen donde queremos guardar nuestra imagen y el último parámetro será la ruta donde se guardará el archivo creado. Ya teniendo almacenado el archivo PNG que contiene la imagen de la FDP, lo siguiente es implementar los mapas dinámicos de Google para “fijar” nuestra imagen de la FDP sobre el mapa. Como mencionamos anteriormente, Google da libre acceso a la documentación necesaria para implementar sus mapas dinámicos y nosotros usamos como guía los ejemplos a los que se tiene en la página de la documentación para Google Maps con Javascript. En particular hicimos uso del ejemplo en el cual se fija un archivo de imagen JPG o PNG sobre los mapas definiendo las coordenadas geográficas latitud y longitud donde se fijará la esquina inferior izquierda de la imagen y la esquina superior derecha de la misma. Dentro del método `processRequest` del SERVLET “`ejecuta.java`” podemos generar código HTML para desplegar una página WEB en nuestro navegador con los resultados generados por el código JAVA, esto pasando como argumento al método “`println`” de la clase “`out`” una cadena de texto con el código HTML que queramos ejecutar. Por ejemplo, con el siguiente código desplegaremos una página WEB que nos informará que la consulta que realizamos no genera ningún resultado:

```
out.println("<html>");
```

```

out.println("<head>");
out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">");
out.println("<title>\<title>");
out.println("</head>");
out.println("<body>");
out.println("<h1 align=\"center\">\</h1>");
out.println("<BODY BGCOLOR=\"#FDF5E6\">");
out.println("<H1 ALIGN=\"CENTER\">Tu consulta no generó resultados</H1>");
out.println("</body>");
out.println("</html>");

```

Esta opción fue muy útil, ya que como mencionamos anteriormente podemos generar nuestros propios mapas utilizando la documentación de los mapas dinámicos de Google para JAVASCRIPT definida por Google. Lo que hicimos fue modificar el código de ejemplo para “fijar” una imagen en formato JPG o PNG sobre los mapas, pasando como argumento la ruta donde se encuentra el archivo con la imagen de la FDP que creamos previamente así como las coordenadas geográficas de la esquina inferior izquierda y esquina superior derecha donde queríamos fijar dicha imagen, el resultado de dicha implementación se muestra en la Figura C.9. Existe mucha información [Svennerberg10] y ejem-

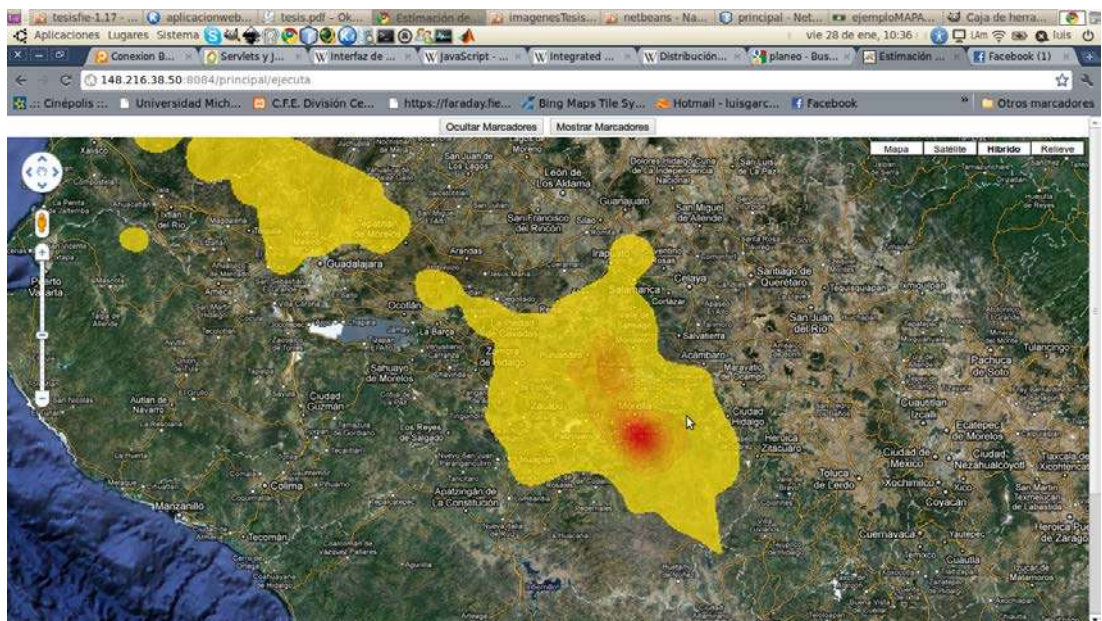


Figura C.9: Aplicación desarrollada para mostrar la FDP sobre los mapas de Google

plos en la WEB, y si todo esto fuera poco, Google nos ofrece libre acceso a

su documentación para los mapas que proporciona para diferentes lenguajes de programación tales como Javascript, PHP o Flash entre otros. Para este trabajo en particular se decidió hacer uso de el lenguaje Javascript para la implementación de los mapas utilizados.

Hasta este momento ya tenemos desarrollada nuestra aplicación WEB y podemos ver en el navegador WEB de nuestra computadora si seleccionamos la opción “run” sobre nuestro archivo “index.jsp”, lo cual nos desplegará el formulario para realizar el cálculo de la FDP y generar la imagen correspondiente, y al dar clic sobre el botón “Enviar Consulta” después de llenar todos los campos de texto debemos obtener un resultado similar al mostrado en la Figura C.9. Por último sólo es necesario dar detalle de cómo instalar un servidor Apache-Tomcat (Apendice A) que aloje nuestra aplicación WEB dentro de la máquina servidor donde está instalada nuestra antena para medir las DA.

C.3. Empaquetado de la aplicación WEB

Una opción práctica y ordenada de mantener nuestra aplicación es empaquetando todos los recursos necesarios para el proyecto en un archivo tipo WAR, de esta forma sólo será necesario manejar un archivo para nuestra aplicación WEB. NETBEANS nos proporciona una opción para empaquetar nuestro proyecto y generar el archivo comprimido. Si damos clic derecho sobre el proyecto (WebApplication1) nos aparecerán algunas opciones entre las que se encuentra CLEAN AND BUILD, esta opción nos facilita la tarea ya que compila todo los archivos necesarios y nos ahorra la tarea de compilar cada uno de ellos por separado generandonos un archivo único comprimido para el proyecto. Este archivo comprimido lo copiamos y lo pegamos en el directorio /webapps y después de esto, podemos ver nuestra aplicación en el navegador web trabajando sobre nuestro servidor APACHE-TOMCAT. En nuestro caso particular, nombramos a nuestro proyecto “WebApplication1” generando un

archivo llamado “WebApplication1.war”, el cual al moverlo al directorio WEBAPPS de nuestra instalación del servidor APACHE-TOMCAT, escribimos localhost:8080/WebApplication1 (donde localhost se reemplaza por la dirección ip de nuestra máquina servidor) y nos despliega la página para generar las consultas mostrada en la Figura C.7.

Referencias

- [APA11] Apache tomcat. <http://tomcat.apache.org/>, 2011. Consulta: Enero 2011.
- [BIN10] Bing maps tile system. <http://msdn.microsoft.com/es-mx/library/bb259689.aspx>, 2010. Consulta: Agosto 2010.
- [BOL10] Boltek home page. <http://www.boltek.com>, 2010. Consulta: Agosto 2010.
- [E. J. Krakiwsky73] E. J. Krakiwsky, y. D. E. W. *Sistemas de Coordenadas en Geodésia*. Fredericton, N.B., Depto. de Ingeniería Topográfica, 1973.
- [FIS10] Conceptos básicos de matemáticas y trigonometría plana. <http://www.fisicanet.com.ar/matematica/trigonometria/ap02trigonometria.php>, 2010. Consulta: Agosto 2010.
- [JAI10] Java advanced imaging api. <http://java.sun.com/javase/technologies/desktop/media/jai/>, 2010. Consulta: Agosto 2010.
- [Loweth97] Loweth, R. P. *Manual of Offshore Surveying for Geoscientist and Engineers*. Chapman and Hall, London, 1^a ed^{ón}., 1997.
- [Miller08] Miller, M. *Googlepedia: the ultimate Google Resource*. Que, New York, 3^a ed^{ón}., nov. 2008.

- [MUL10] Multimap open api v1.2 web services documentation static maps. <http://www.multimap.com/openapidocs/1.2/web-service/staticmaps.htm>, 2010. Consulta: Agosto 2010.
- [NET11] Indice de paquetes de descarga de netbeans. <http://netbeans.org/downloads/index.html>, 2011. Consulta: Enero 2011.
- [OVE11] Código google para sobreposición de una imagen sobre el mapa. <http://code.google.com/intl/es-ES/apis/maps/documentation/javascript/examples/overlay-simple.html>, 2011. Consulta: Enero 2011.
- [Ross96] Ross, S. M. *Stochastic Processes*. WILEY, New York, 2^a edición., 1996.
- [SIS10] Simulación de eventos atmosféricos de nextorm. <http://148.216.38.50:8080/sistorm/view/Simulacion.php>, 2010. Consulta: Agosto 2010.
- [Svennerberg10] Svennerberg, G. *Beginning Google Maps API 3*. Apress, New York, 2^a edición., jul. 2010.
- [Vincenty75] Vincenty, T. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *DMAAC Geodetic Survey Squadron*, 1975.
- [W. Schofield07] W. Schofield, M. B. *Engineering Surveying*. Butterworth - Heinemann, Oxford, 6^a edición., 2007.