Proyecto “Sistema de resúmenes estadísticos climáticos de regiones definidas en la República Mexicana” TH 1302.1

Informe final

Coordinación de Hidrología
Subcoordinación de Hidrometeorología

Julio Sergio Santana
Efraín Mateos Farfán
Indalecio Mendoza Uribe

1 ÍNDICE

2 RESUMEN EJECUTIVO

El sistema resultante del presente proyecto, permite realizar un conjunto de operaciones estadísticas sobre una base de datos para una región definida en el territorio de la República Mexicana. Las regiones típicas que se manejan son: cuencas, estados, y regiones arbitrarias definidas por un usuario. La base de datos es el CLICOM, en la cual el Servicio Meteorológico Nacional (SMN), mantiene la información de sus estaciones climáticas distribuidas a lo largo y ancho del país. Las operaciones estadísticas que se pueden obtener son: tablas de valores medios por años y meses, histogramas de frecuencias, curvas distribución de probabilidades, curvas de períodos de retorno y, para el caso de la precipitación, tabla de curvamasas por años. El sistema está formado por tres módulos a saber: la interfaz con el usuario mediante una página *Web*, la máquina computacional y la base de datos.

3 OBJETIVOS

Proveer de herramientas interactivas, en línea, para el procesamiento y análisis estadístico de la información observada, proveniente de alguna base de datos climática o meteorológica como podría ser alguna red de estaciones climáticas distribuidas a lo largo y ancho del país. Para su análisis, las estaciones o puntos de malla de la base de datos, podrán ser agrupadas en regiones geográficas preestablecidas, tales como cuencas o estados, o bien, en regiones arbitrarias definidas en línea por el usuario. El sistema tendrá como entrada una página Web, que habilitará al usuario distintas opciones interactivas para formular su consulta, y que como respuesta a esa interacción, le entregará como resultado las salidas de un conjunto de procesos, escritos en lenguajes computacionales adecuados para la tarea, tales como R, Python, Grads, etc.

4 ANTECEDENTES

México cuenta con alrededor de 5916 estaciones climáticas con información registrada, en algunos casos, desde 1951. Esta información es administrada por el SMN, mediante la base de datos CICLOM. El IMTA ha desarrollado sistemas que permiten consultar esta información cruda, como el Eric III, que además en algún grado permite la visualización de la información mediante gráficos, tales como mapas y series de tiempo, y que provee de una estadística básica de la información, a saber, promedios y desviaciones estándar por mes para el período seleccionado. Aunque el citado sistema

ha sido una herramienta valiosísima, carece de algunas facilidades que potenciarían su utilidad, al extender sus capacidades para la producción de diferentes resúmenes estadísticos, amén del asunto de la actualización de la información y del diseño de una nueva interfaz con el usuario para hacerlo más asequible a las personas comunes. Esta es la principal motivación para el sistema que se ha desarrollado en este proyecto. La base de datos de CLICOM no es la única que provee información climática del país. Existen otras redes de estaciones climáticas, como la de las Fundaciones PRODUCE, por ejemplo, además de otras bases de datos, que igualmente pudieran constituir la fuente de información de los desarrollos planteados en el presente proyecto.

La base de datos CLICOM, en su estado actual, reside en formatos que no son directamente utilizables por los distintos sistemas y/o lenguajes de análisis y procesamiento estadístico de la información. Por otra parte, los mecanismos para su actualización son heterogéneos, ya que la información de algunas de las estaciones se actualiza de manera automática y oportuna, mientras que la de algunas otras, se hace de manera prácticamente manual y en los momentos en que esa información se hace disponible por los operadores en campo. Atendiendo a esos dos problemas, el proyecto está pensado para establecer un conjunto de procedimientos para construir una “copia” de la base de datos, en un formato adecuado para la extracción de las secciones de información que requieran los procedimientos de procesamiento y análisis estadístico desarrollados también como parte del proyecto. Del mismo modo el proyecto atiende al desarrollo de procedimientos, justamente para extraer secciones de la base de datos y para transparentar al usuario las actualizaciones de la base de datos CLICOM, original.

Por su parte, la puerta de entrada visible para el usuario, al conjunto de servicios desarrollados como parte de este proyecto, es mediante una página *Web*. La función de este sitio es la de ayudarle al usuario a configurar o formular de una manera sencilla e interactiva, consultas a la base de datos de estaciones climáticas, y la de presentarle como respuesta la información requerida en formatos textuales o gráficos, con diseños que permitan comprenderla de manera rápida. Por consiguiente, el sistema involucra procedimientos para la formulación de consultas a la base de datos, que van más allá de lo que se conoce como *queries* en las bases de datos tradicionales, pues permiten no solamente solicitar secciones o fragmentos de la base de datos, considerando las coordenadas geográficas y de tiempo, sino también solicitar la ejecución de operaciones de procesamiento y análisis estadístico sobre los datos seleccionados. Se atiende también al desarrollo de un módulo central, encargado de despachar las consultas generadas por el usuario a los programas, sistemas o agentes apropiados para satisfacer la consulta. Dependiendo de la naturaleza de la respuesta, este mismo módulo se encarga de invocar los procedimientos necesarios para presentarle la respuesta final al usuario, ya sea ésta, textual o gráfica.

Finalmente la maquina computacional representa el núcleo central del proyecto lo constituye la integración de un conjunto de rutinas para el procesamiento estadístico de los datos climáticos provenientes de las estaciones cuya información se mantiene en la base de datos CLICOM. La definición de este conjunto de rutinas, se desprende de la

determinación de las operaciones estadísticas mayormente solicitadas por especialistas en las áreas de meteorología y climatología. Cada operación entrega como producto, un texto, que contiene un dato numérico o categórico, o un conjunto de ellos, arreglados como una tabla, o un gráfico, que muestra visualmente los resultados de la operación. Este núcleo incluye asimismo, las rutinas para las selecciones de regiones geográficas. Aparte de contar con áreas geográficas preestablecidas, tales como cuencas y estados, se provee al usuario con la posibilidad de definir sus propias regiones geográficas y guardarlas para su posterior uso.

5 METODOLOGÍA

El ataque a los problemas planteados por el presente proyecto se ha hecho básicamente en cuatro frentes a saber:

1. TRATAMIENTO DE LA BASE DE DATOS CLIMÁTICA Y/O METEOROLÓGICA. En general, la bases de datos climáticas y meteorológicas, residen en formatos que no son directamente utilizables por los distintos sistemas y/o lenguajes de análisis y procesamiento estadístico de la información. Por otra parte, los mecanismos para su actualización son heterogéneos, ya que la información de algunas de las estaciones se actualiza de manera automática y oportuna, mientras que la de algunas otras, se hace de manera prácticamente manual y en los momentos en que esa información se hace disponible por los operadores en campo. Atendiendo a esos dos problemas, se desarrollará un conjunto de procedimientos para construir una “copia” de la base de datos, en un formato adecuado para la extracción de las secciones de información que requieran los procedimientos de procesamiento y análisis estadístico desarrollados en otro de los apartados de esta metodología. Del mismo modo se desarrollarán los procedimientos, justamente para extraer secciones de la base de datos y para transparentar al usuario las actualizaciones de la base de datos original.

2. INTERACCIÓN CON EL USUARIO. La puerta de entrada visible para el usuario, al conjunto de servicios desarrollados como parte de este proyecto, será un sitio Web. La función de este sitio será la de ayudarle al usuario a configurar o formular de una manera sencilla e interactiva, consultas a la base de datos de estaciones climáticas, y la de presentarle como respuesta la información requerida en formatos textuales o gráficos, con diseños que permitan comprenderla de manera rápida. Por consiguiente, se desarrollarán procedimientos para la formulación de consultas a la base de datos, que van más allá de lo que se conoce como *queries* en las bases de datos tradicionales, pues permitirán no solamente solicitar secciones o fragmentos de la base de datos, considerando las coordenadas geográficas y de tiempo, sino también solicitar la ejecución de operaciones de procesamiento y análisis estadístico sobre los datos seleccionados. Se desarrollará también un módulo central,

encargado de despachar las consultas generadas por el usuario a los programas, sistemas o agentes apropiados para satisfacer la consulta. Dependiendo de la naturaleza de la respuesta, este mismo módulo se encargará de invocar los procedimientos necesarios para presentarle la respuesta final al usuario, ya sea ésta, textual o gráfica. En este mismo tenor, aquí se determinará por razones de eficiencia, qué parte de las tareas descritas se se llevarán a cabo en el servidor y qué parte en el cliente.

3. PROCESAMIENTO ESTADÍSTICO Y GEOMÉTRICO/GEOGRÁFICO. El núcleo central del proyecto lo constituye la integración de un conjunto de rutinas para el procesamiento estadístico de los datos climáticos provenientes de las estaciones o puntos de malla, cuya información se mantiene en la base de datos climática y/o meteorológica. La definición de este conjunto de rutinas, se desprenderá de la determinación de las operaciones estadísticas mayormente solicitadas por especialistas en las áreas de meteorología y climatología. Cada operación podrá entregar como producto, un texto, que contenga un dato numérico o categórico, o un conjunto de ellos, arreglados como una tabla, o un gráfico, que muestre visualmente los resultados de la operación. Aunque no forman parte directamente del procesamiento estadístico, las rutinas para las selecciones de regiones geográficas se incluyen en este apartado. Aparte de contar con áreas geográficas preestablecidas, tales como cuencas y estados, se proveerá al usuario con la posibilidad de definir sus propias regiones geográficas y guardarlas para su posterior uso. Aunque, la captura de estas áreas y su almacenamiento en la máquina cliente, es tarea del frente de interacción con el usuario, el uso de esa información para el recorte de la información de la base de datos queda en el frente que se define aquí

4. INTEGRACIÓN E INTERACCIONES ENTRE COMPONENTES. Este frente se focaliza en la definición y desarrollo de las interfaces entre los distintos componentes del proyecto. Cada componente, al momento de ejecutarse, ya sea en el servidor o en el cliente, se convierte en un proceso que recibe ciertas entradas y produce ciertas salidas. Generalmente, las salidas de algún proceso sirven como entradas de algún otro proceso. Por consiguiente, se definen aquí los formatos y las formas de esas salidas/entradas de los procesos. Por formatos se entiende aquí la estructura de la información que es comunicada entre los procesos, y por formas, la manera en la que se desarrolla esa comunicación. Estas definiciones y su implementación constituyen las articulaciones del esqueleto en el que se fundamenta el proyecto. Es decir, el funcionamiento articulado de los componentes del proyecto, como un todo, depende de las definiciones hechas en este apartado y en sui implementación.

6 RESULTADOS

A continuación se listan de manera breve los principales resultados del presente proyecto, que coinciden con los **entregables** del mismo.

1. Conjunto de rutinas y/o procedimientos en distintos lenguajes de programación, para la extracción, manipulación y organización de la información climática contenida en las bases de datos climáticas y meteorológicas y para el procesamiento geográfico/geométrico y estadístico de la misma.

2. Sitio Web, que sirve como interfaz para que un usuario común pueda formular interactivamente una consulta a la información contenida en las bases de datos climáticas y meteorológicas y que se satisface en términos de llamados al conjunto de rutinas descrito en el punto anterior.

Con el fin de abundar en el detalle de los resultados, se han añadido al presente documento tres anexos en las secciones 9, 10 y 11. Los primeros dos anexos corresponden al sitio Web y el tercer anexo, al conjunto de rutinas.

7 CONCLUSIONES

Los distintos componentes del presente sistema resultan útiles para el análisis de la información histórica guardada en las bases de datos de observaciones meteorológicas y climáticas.

Para aprovechar cabalmente las bondades del presente desarrollo, es muy conveniente instalar el sitio Web en un servidor accesible por todo el público. De este modo, mediante las estadísticas de uso del sitio, se podrá corroborar la utilidad de los presentes desarrollos.

Cabe mencionar que partes importantes del presente desarrollo se están usando ya en los procedimientos de validación de modelos que el IMTA ha entregado al Servicio Meteorológico Nacional como parte del proyecto TH 1318.3.

8 AGRADECIMIENTOS

Se agradece el apoyo del Ing. Delker Emmanuel Martínez Ocampo en el proyecto, particularmente en lo que se refiere a la elaboración de la interfaz al usuario, y al Dr. Obed Pérez Cortés, por su colaboración en el desarrollo de rutinas estadísticas.

9 ANEXO 1: Diseño y Desarrollo del Sistema

9.1 Arquitectura del Sistema

A continuación se muestran las principales características estructurales y funcionales del sistema desarrollado aquí:

1. **Interfaz al usuario mediante una página *Web*.** Dos de las grandes ventajas de presentarle al usuario un sistema mediante una página *Web*, son que el sistema es *omnipresente*, en la medida que *Internet* lo es, y también que las actualizaciones, tanto de la información subyacente, como del sistema mismo son inmediatas y transparentes para el usuario. De este modo no se tendrá que generar un nuevo sistema año con año para incorporar la nueva información generada por las estaciones.
2. **Selección interactiva de regiones de interés.** El sistema ofrece diferentes formas amigables para la selección de regiones de interés dentro de las cuáles estará el conjunto de estaciones a considerar. Estas formas van desde regiones predefinidas, como pueden ser cuencas, entidades federativas, etc., hasta la selección interactiva mediante polígonos y rectángulos directamente indicados en un mapa sensible. Estas regiones, se pueden guardar por el usuario, y ser realimentadas al sistema, cuando así lo desee. Desde el punto de vista interno, el sistema cuenta con un algoritmo eficiente para la inclusión o exclusión de estaciones dentro de la región seleccionada.
3. **Distintos tipos de análisis estadísticos para la región seleccionada**. El sistema tiene la capacidad de hacer distintos tipos de análisis estadísticos para la región seleccionada
4. **Utilización de las herramientas computacionales adecuadas**. Uno de los énfasis de este proyecto es el uso de las herramientas computacionales adecuadas; esto es, a cada herramienta seleccionada se le dedicará a ejecutar las tareas en las que mejor es su desempeño. Así, la interfaz al usuario se ha desarrollado con herramientas tales como el los lenguajes PHP, JavaScript y HTML5, al tiempo que el manejo estadístico de la información se hace principalmente mediante el lenguaje R. Para esto se han establecido los canales adecuados de comunicación entre cada uno de los elementos que conforman el sistema.
5. **Facilidad de adaptar este sistema a otros países o regiones del mundo**. Aunque el presente sistema se plantea para México, su adaptación a otros países que cuenten con un sistema de estaciones climáticas distribuidas a lo largo y ancho de su territorio será sencillo, toda vez que se defina un formato adecuado para la base de datos con la información de las estaciones en cuestión.

6. **La arquitectura del sistema**. Esta se plantea, como se muestra en la figura 1, compuesta de los elementos:
    a. **Base de Datos.** Está conformada por la información histórica de las estaciones climáticas. Esencialmente es una réplica de la base de datos CLICOM, salvo porque se ha cambiado el formato a uno más adecuado para el sistema desarrollado.
    b. **Máquina Computacional.** Es un conjunto de procesos, escrito en varios lenguajes, como R, php, Python, Perl y Bash, que se encargan de satisfacer las peticiones del usuario, que le son entregadas desde la interfaz al usuario. Para ello, hace consultas a la base de datos y efectúa las operaciones matemáticas requeridas sobre la información. Los resultados de sus operaciones son entregadas a la interfaz al usuario para su presentación final.
    c. **Interfaz Gráfica al Usuario.** Ésta tiene la forma final de una página *Web* que permite al usuario introducir sus peticiones y recibir en forma inteligible los resultados de sus solicitudes.

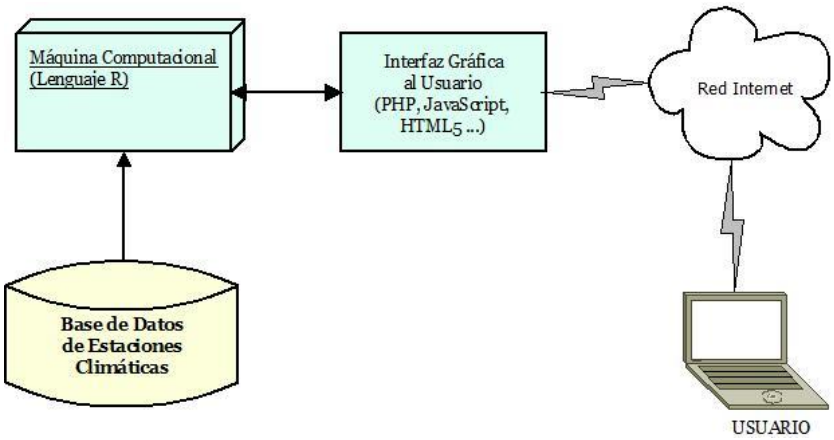


Figura 1. Arquitectura del sistema

En las siguientes secciones se exponen de manera individual los módulos que componen la arquitectura del sistema: Interfaz Gráfica al usuario y Maquina Computacional.

9.2 Interfaz Gráfica con el Usuario

El proceso de desarrollo de un conjunto de servicios como parte de un sitio web, siendo esta una puerta de entrada visible para usuarios, se realizó en base a especificaciones necesarias tales que ayudarán al usuario a configurar o formular de una manera sencilla e interactiva, consultas a la base de datos de estaciones climáticas, y la de

presentarle como respuesta la información requerida en formatos textuales, tablas o gráficos, con diseños que le permitirán comprenderla de manera rápida. Para esto se procedido a buscar las tecnologías Web más convenientes.

Como lenguaje base se tiene php, que constituye el núcleo de la interfaz, puesto que con él se logra la integración de otros lenguajes y procesos que se ocupan para ejecutar Scripts, almacenar archivos de texto con coordenadas, leer archivos de texto con coordenadas, procesar comandos para mandar a llamar procesos, y la generación de una página como resultado de las operaciones generadas por el usuario.

Se emplea también la tecnología Ajax del lenguaje JavaScript para la elaboración de aplicaciones interactivas manteniendo comunicación asíncrona. Ajax es una técnica de desarrollo Web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios, mientras que se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que se refleja directamente en la interactividad, velocidad y usabilidad en las aplicaciones.

Se utiliza también la tecnología Jquery del lenguaje JavaScript para la simplificación en la interacción con páginas dinámicas. JQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica Ajax a páginas web. Al igual que otras bibliotecas, JQuery, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Se utilizó además la tecnología de Google Maps, que es el nombre de un servicio gratuito de Google, mediante un servidor de aplicaciones de mapas en la Web. Ofrece imágenes de mapas desplazables, así como fotos satélites del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle Google Street View. Ofrece, asimismo, la posibilidad de que cualquier propietario de una página Web integre muchas de sus características a su sitio. En particular, aquí se empleó para la implementación de un área grafica donde se pudiera seleccionar una sección en la que se aplicarían operaciones estadísticas, se realizó un análisis y estudio de la API de Google Maps para inserción de Mapas en un Sitio Web mediante JavaScript, así como sus respectivas funciones para la elaboración de Polígonos, poli líneas, y rectángulos. Los polígonos son objetos cerrados; se crean con la clase google.maps.Polygon y se requiere de dos argumentos dentro del objeto google.maps.PolygonOptions:

Path: requiere de varios puntos que indican la latitud y longitud usando la clasegoogle.maps.LatLng.
Map: el mapa donde se va a mostrar el polígono, usando la clase google.maps.Map. Esta clase, contiene dentro varios eventos que pueden ser utilizados para la interacción con el usuario, un ejemplo por decir alguno es ‘rightclick’ que puede realizar algún

despliegue de información o cualquier cosa que se pueda realizar en una función de JavaScript.

Se Investigó además sobre la extracción de información de vértices para formar polígonos y sus respectivos eventos y funciones para mostrar o no mostrar paneles de Google Maps.

Se analizaron archivos KML y KMZ con el objetivo de poder mostrar Capas (Layer) de polígonos en mapas de Google Maps, así como su elaboración desde el Programa instalable, Google Earth para poder manejarlos en un sitio Web.

KML es un lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones. Los ficheros KML a menudo suelen distribuirse comprimidos como ficheros KMZ.Un fichero KML especifica una característica (un lugar, una imagen o un polígono) para Google Earth. Contiene título, una descripción básica del lugar, sus coordenadas (latitud y longitud) y alguna otra información.

Posteriormete se realizaron pruebas en el programa Google Earth donde se trazaron polígonos y se guardaron como layers o capas en formato KMZ y se alojaron en el Sitio de Google sites, para poder desde ahí manipularlos en una página Web y presentarlos en un mapa (en este caso un mapa de la República Mexicana).

También se investigó el máximo de capacidad que puede soportar Google Maps en un archivo KMZ, ya que si el tamaño sobrepasa los 3MB no muestra el Layer.

Adentrándonos en la investigación se encontró que las propiedades de los archivos kml y kmz son para mostrar capas en la API de Google Maps y se descubrió que estos solo funcionan como capas pero no se puede dar otra funcionalidad como poder recuperar sus coordenadas. Por ese motivo se buscó la implementación de la creación de polígonos definidos desde JavaScript para la utilización de estos como capas de mapa del sitio web dándole la capacidad de interacción con el usuario.

Ya reconocidas las tecnologias y lenguajes que se utilizarón se decidio trabajar el sitio web para los navegadores Google Chrome y Mozilla Firefox para esto se realizó un método en JavaScript para poder identificar el navegador que se está utilizando al acceder al sitio web y poder realizar las operaciones de acuerdo al navegador y así poder mostrar los resultados de manera correcta.

Para esto se implementaron métodos en JavaScript para definir varios polígonos que en primer instancia cubrieran el mapa de la República Mexicana, teniendo mapas en dos formas: la primera, seccionando el mapa por estados, y la segunda es seccionándolo por cuencas hidrológicas.

Buscando el resultado que se deseaba nos encontramos con el problema de obtener los vértices de estos polígonos desde una bodega de datos y así no tener almacenados

los puntos de las coordenadas dentro del código de JavaScript ya que no es factible tenerlos de esta manera, aunque su función es buena. De esta manera nos dimos a la tarea de iinvestigar en primer lugar sobre archivos xml para la utilización en almacenamiento de coordenadas de polígonos para su lectura desde JavaScript. Encontrando que XML es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

XML no ha nacido sólo para su aplicación para internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Como segunda opción se investigó sobre el manejo de archivos JSON para la interacción con JavaScript para el posible uso de estos archivos como bodega de datos para las coordenadas utilizadas para crear polígonos desde JavaScript.

JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX.

Una de las ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON.

En JavaScript, un texto JSON se puede analizar fácilmente lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

En la práctica, los argumentos a favor de la facilidad de desarrollo de analizadores o del rendimiento de los mismos son poco relevantes, JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia cuando la fuente de datos es explícitamente de fiar.

Procedimos a extraer cuencas y estados con sus vértices correspondientes de un archivo kmz utilizando Google Earth como medio de extracción de estos, para

insertarlos en el formato JSON para la manipulación en JavaScript y formar los polígonos en el mapa de la República Mexicana.

Teniendo lo anterior como base, decidimos utilizar dos archivos JSON como bodega de datos para el uso exclusivo de las coordenadas para la creación de los polígonos que se muestran en la API de Google Maps, un archivo para las coordenadas de los Estados de la República Mexicana y otro para las coordenadas de las Cuencas Hidrológicas. Esto con el fin de reducir el código en los archivos de JavaScript y hacer ligeros estos mismos dándole interacción a eventos, donde al seleccionar cada polígono aparece un tooltip mostrando información sobre el polígono.

Realizando de esta manera parte del sitio web se redujo el código hasta en un 70 % haciéndolo con mejor orden y mayor estabilidad. La ventaja es que el formato de los datos es sencillo y la manera de obtenerlos en un archivo Javascript es muy práctica, dándole un soporte compacto y mostrando de manera agradable y funcional los datos de una forma gráfica y entendible.

Se crearon tooltips mediante el framework infobox de Javascript para mostrar información en globos estilo bocadillo de los Estados, así como de las cuencas hidrológicas dando una buena apariencia.

Un tooltip (también llamada descripción emergente) es una herramienta de ayuda visual, que funciona al situar el cursor sobre algún elemento gráfico, mostrando una ayuda adicional para informar al usuario de la finalidad del elemento sobre el que se encuentra. Los tooltip son una variación de los globos de ayuda y es un complemento muy usado en programación y diseño, dado que proporcionan información adicional sin necesidad de que el usuario la solicite de forma activa.

Teniendo lo anterior Investigamos sobre el manejo de sesiones en el Lenguaje php ya que es parte esencial de este proyecto. El soporte para sesiones en PHP consiste en una forma de preservar cierta información a través de accesos subsiguientes. Esto habilita la construcción de aplicaciones más personalizadas e incrementa el atractivo de un sitio web. Un visitante que accede a su sitio web se le asigna un id único, también llamado id de sesión. Éste es almacenado en una cookie en la parte del cliente o se propaga en el URL.

Después se buscó la Simplificación de código para no tener un uso excesivo en los recursos del Ordenador y así evitar que se hiciera lento, ya que cuando se extraen los vértices mediante php para ser guardados en un archivo de texto puede ser que tarde bastante porque genera en algunas ocasiones hasta 13000 ciclos para dibujar un polígono. Esto se simplificó de una manera en la que se hicieron archivos que tienen los vértices de cada cuenca, así como archivos que tienen los vértices de los Estados de la República Mexicana y para no generarlos cada vez que se selecciona una cuenca, o en su caso un Estado, se copia este archivo y se pega en un nuevo directorio, conservando al principio el nombre del usuario que este en sesión y

separado por un guión bajo el nombre de la cuenca o el Estado. Esta acción se lleva acabo programando un evento en Php para acceder a la ruta de los archivos, copia el archivo .txt y lo pega en la ruta deseada, antes cambia el nombre (en este caso agregando el nombre del usuario al principio), esto nos ayuda reduciendo en un 80% el tiempo de espera de cada creación de archivos txt.

Posteriormente se hizo un análisis de la base de datos Wendy, que es la base de datos de observaciones de estaciones climáticas en la República Mexicana y que representa el paso de la información textual provista por la base de datos CLICOM, administrada por el Servicio Meteorológico Nacional, hacia el manejador de bases de datos MySQL. Se revisó como funciona esta base de datos y se generaron ciertas consultas para ver su estructura y funcionalidad. En esta revisión se observó que un campo estaba mal colocado ya que nunca se estaba utilizando, y estaba contemplado en la mayoría de las tablas que componen esta. Al eliminarse este campo se rescató un espacio de memoria de 600 Mb aligerando la Base de Datos. La estructura de la base de datos se da en el diagrama de la Fig. 2.

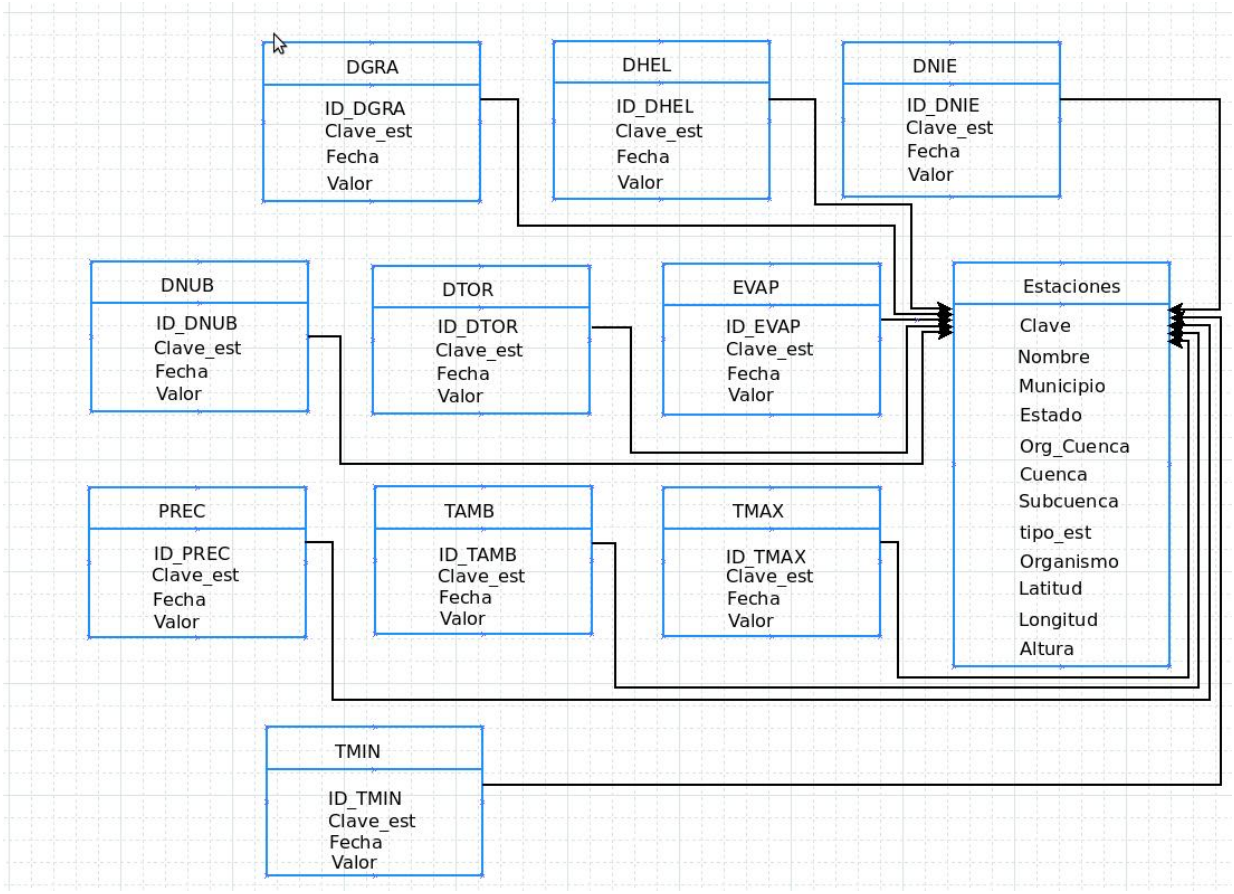


Figura 2. Estructura de la base de datos

En la figura 2,

DGRA = Día Granizo.
DNUB = Día Nublado.
DTOR = Día Tormenta.
EVAP = Evaporización.
PREC = Lluvia acumulada en 24 horas.
TAMB = Temperatura a la hora de la Observación (8:00 am).
TMAX = Temperatura Máxima.
TMIN = Temperatura Mínima.
Estaciones = Estaciones de monitoreo.

Posteriormente al análisis de la base de datos se generó un método en Php que realiza una conexión a la base de datos WENDY el cual extrae datos de un determinado rango de tiempo que el usuario específica y devuelve la suma del valor de las precipitaciones de este rango, mostrando el resultado de la operación en una tabla.

Con MySQL que es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Utilizamos el lenguaje de consulta estructurado o SQL (structured query language) que es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

Lo anterior, sirvió como una prueba para realizar posteriormente operaciones más complejas en las cuales se puedan obtener resultados de mayor funcionalidad para el proyecto, como la realización de esa y otras operaciones pero también con un agregado donde se puedan obtener datos tales como, las estaciones climatológicas que se encuentran dentro de un área determinada y esto en conjunción con la página web donde se seleccionan dichas áreas nos pueden arrojar resultados muy buenos y útiles para enviarlos mediante procesos y generar operaciones estadísticas que nos devuelvan algunas tablas, Imágenes con gráficas o algún texto en específico.

En JavaScript se realizaron métodos para obtener las coordenadas de ciertos polígonos en este caso de estados, cuencas, coordenadas de polígonos personalizados y Coordenadas de polígonos predefinidos, en el caso de los últimos, estos son hechos por el Usuario. Las coordenadas se obtienen mediante la tecnología AJAX enviando mediante url, un Id de operación, en el caso de los estados, cuencas y polígonos predefinidos se envía el nombre del archivo de texto específico para poder realizar una rutina que lea este archivo y pueda obtener las respectivas coordenadas, en el caso de los polígonos personalizados dónde se pasan por url para ser recibidas y utilizadas en un SCRIPT echo en Php para ejecutar comandos de consola. Esto fue realizado como prueba para la realización de operaciones, las cuales se mandan mediante procesos en terminal utilizando Php los cuales se reciben en otros procesos

para realización de operaciones estadísticas, las cuales devuelven resultados en tablas, imágenes con gráficas y textos, que son útiles para mostrar al usuario los resultados de las operaciones deseadas.

Para mostrar los polígonos predefinidos se realizó un método en JavaScript para obtener las coordenadas y mostrar los polígonos que se cargan desde un archivo de texto en un mapa y así poder realizar la operación deseada por el usuario, este es de mucha ayuda puesto que el usuario puede generar un polígono en algún otro programa, y guarda las coordenadas en un archivo de texto y puede subir el archivo en la página y desde ahí ésta lo puede mostrar en un mapa para ayuda visual del usuario.

Se realizó la implementación de un archivo procesos.ini para generar un titulo en la página y un menú dinámico el cual fuera fácil de modificar por el usuario. Para esto en el archivo .ini se buscó la manera de poder ejecutar comandos, colocando una etiqueta que se lee como título del ítem del menú “Operaciones”, adelante un signo “**=**” y después se coloca el comando que se desee ejecutar. Al ser procesados los comandos obtenemos resultados gráficos de validaciones, que de manera visual los podemos ver en 5 imágenes diferentes: Grafico de Pronósticos, Grafico de observaciones, Diferencia: Observado—Modelo, Diagrama de Taylor y Dispersión.

Ejemplo del archivo procesos.ini:

[Titulo]
Titulo=PÁGINA DE PRUEBA PARA VALIDACIÓN DE PRONÓSTICOS
[palabras_clave_Menú]
Prueba de textos=./aaa.sh
Precipitacion General=./Redirector2.sh
Precip. Gro 10/09/2013=./Redirector2.sh GRO 10
Precip. Gro 15/09/2013=./Redirector2.sh GRO 15
Precip. Ver 10/09/2013=./Redirector2.sh VER 10
Precip. Ver 15/09/2013=./Redirector2.sh VER 15

Estas líneas generan en nuestra página web ítems dentro del menú de “Operaciones” que posteriormente el usuario puede ocupar para realizar alguna operación estadística que este desee, antes seleccionando un área en el mapa de la República Mexicana.

Se realizaron pruebas con archivos del Lenguaje R, modificando su contenido para mandar distintas salidas en php y así este pudiera interpretar las posibles salidas como son: Texto, Imágenes y 4 diferentes tipos de tablas. Las tablas son:

1.- Con encabezado y con lomo.
2.- Solo con encabezado.
3.- Solo con lomo.
4.-Sin encabezado y sin lomo.

La página generada desde las operaciones enviadas de datos se ordena de acuerdo al contenido y en el orden que se mandan las secciones de los datos.

Se realizó una breve investigación sobre Expresiones Regulares en el lenguaje php ya que son de muy buena utilidad y practicidad en el reconocimiento de expresiones que se tienen definidas y necesitan compararse, en este caso las expresiones de identificación del lenguaje discreto creado para identificación de Textos, Imágenes y Tablas.

Se realizó un respaldo de cada modificación de la página de los meses de trabajo. Para asegurarnos de que la página web este en las óptimas condiciones. Se realizó un diagrama de flujo del producto realizado, incluyendo la integración de lenguajes y procesos externos, que se muestra en la Fig. 3.

Dado que el sistema es fácilmente modificable por los administradores, se desarrolló un manual el cual guía tanto al usuario como al administrador, paso a paso para el correcto uso y especificación del sitio web; este manual se presenta en el anexo 2 (capítulo 10) del presente documento.

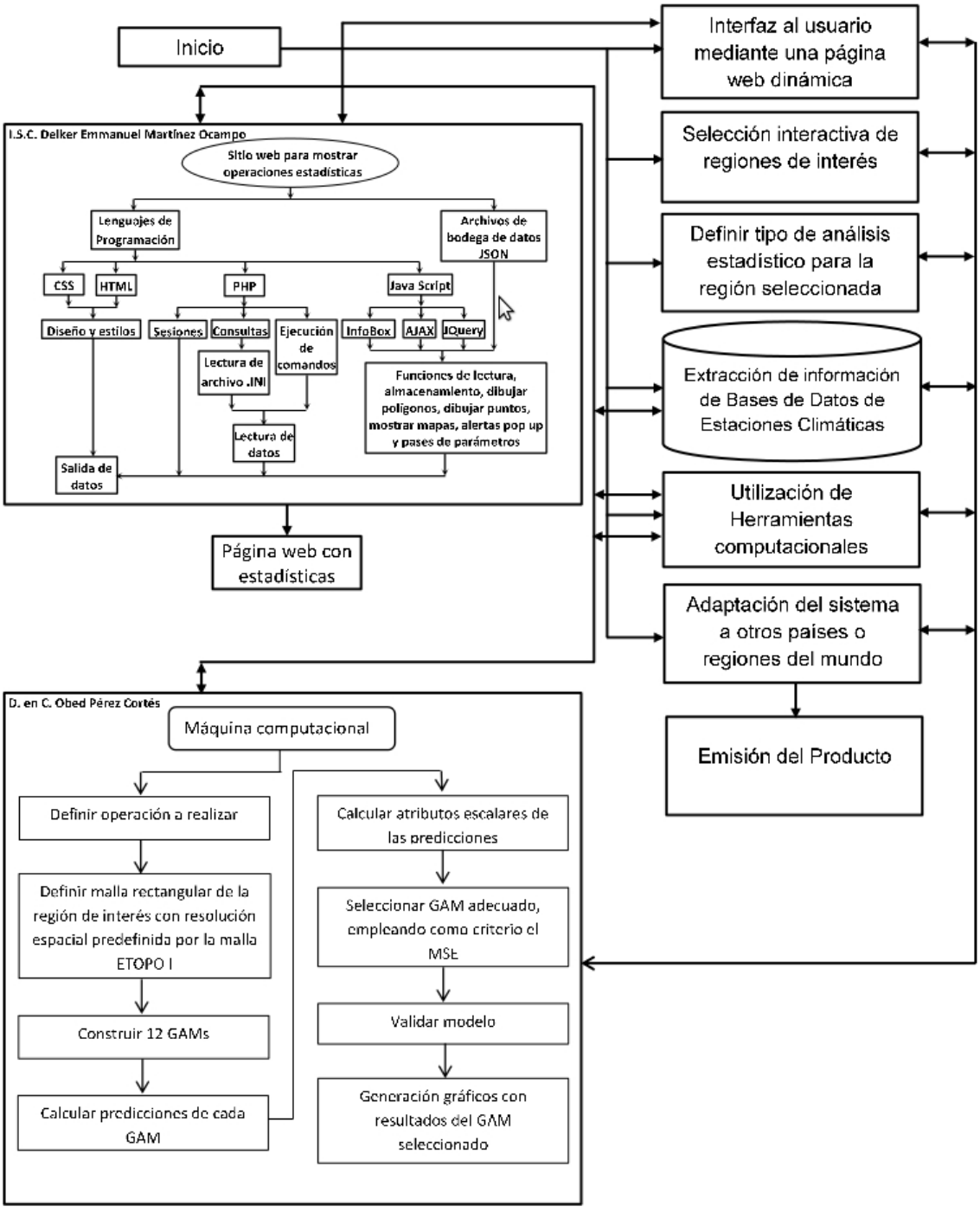


Figura 3. Detalle operacional del sistema

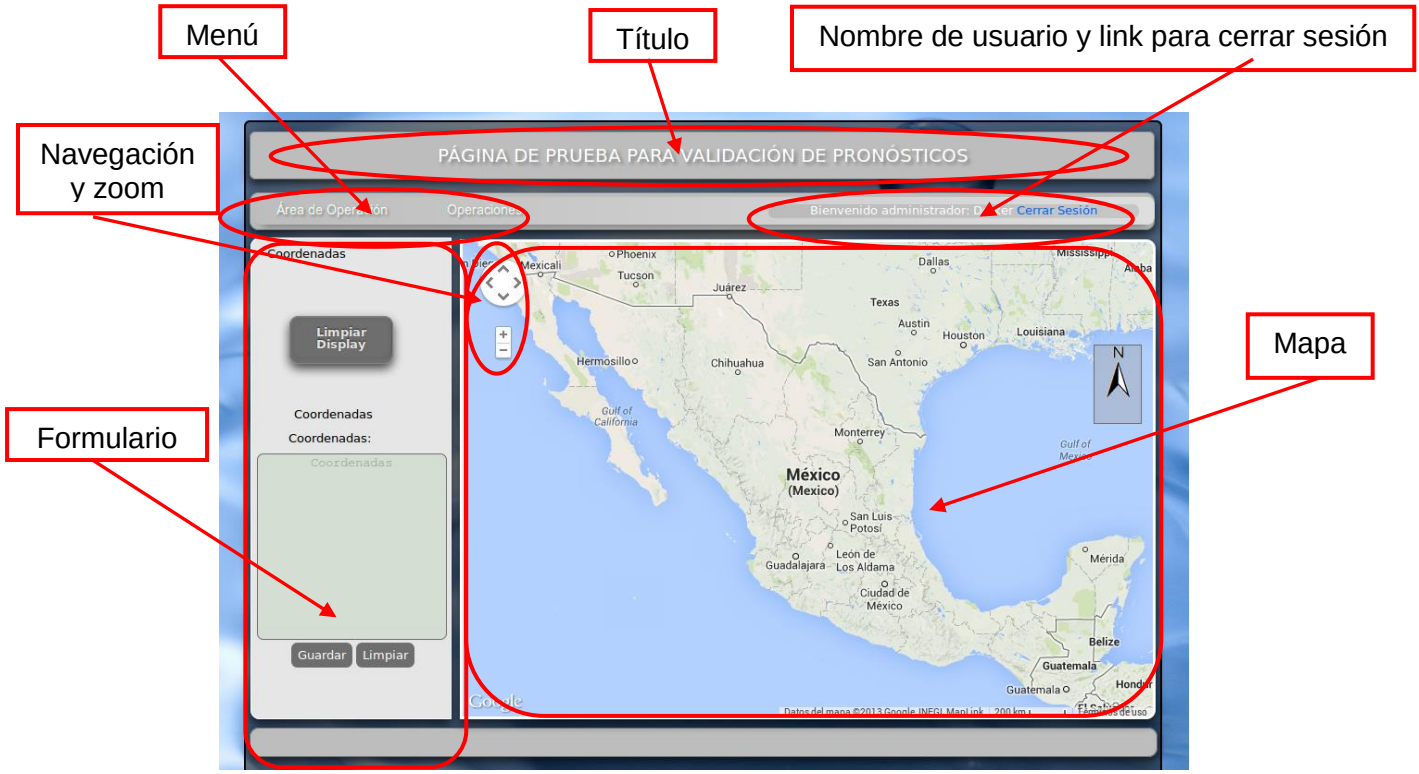
10 ANEXO 2: Interfaz con el Usuario: Manual de DEGEOPRO v. 1.0

Se ha denominado a esta parte del sistema como DEGEOPRO, que significa Despachador Geográfico de Procesos. Su funcionamiento es como se presenta a continuación.

DESCRIPCION GENERAL.

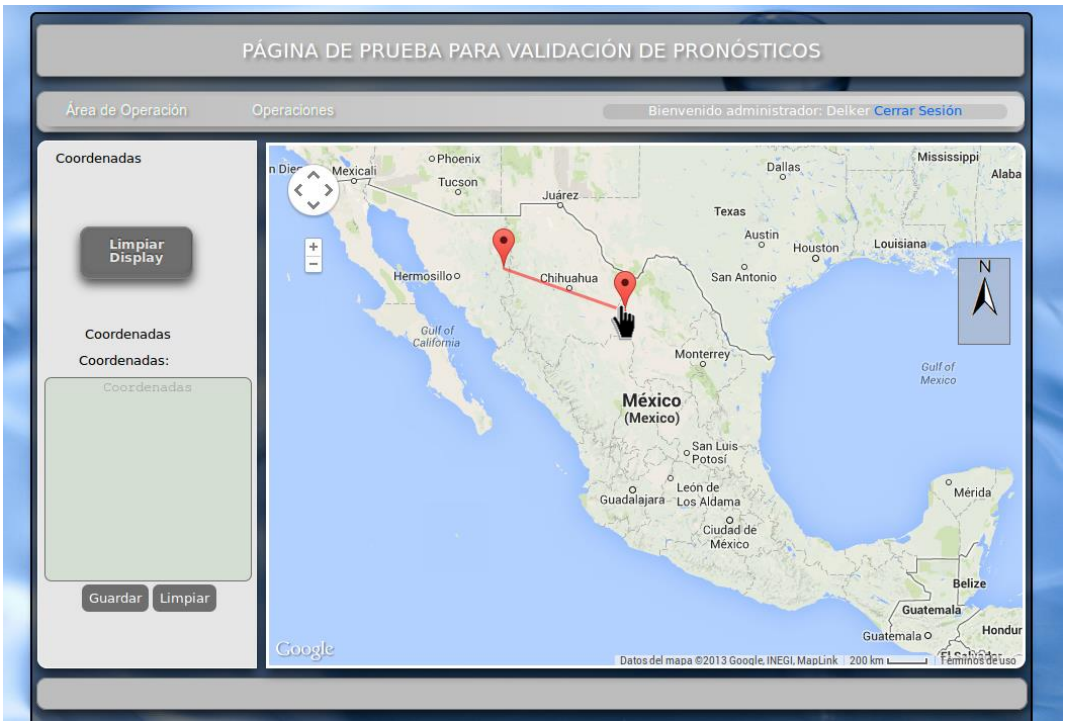
Al acceder a la ruta de la página web encontraremos el formulario mostrado en la imagen de arriba.

Para tener acceso a la página es necesario ingresar un nombre de usuario en el cuadro de texto, (este puede ser un nombre cual sea) y dar clic en el botón “Iniciar sesión”.



Al iniciar sesión se muestra en la pantalla esta imagen en donde en la parte superior se muestra el título de la página, cabe mencionar que este título lo ingresa el usuario desde un archivo .INI que más adelante explicaremos. En la parte de abajo una barra con los menú de Área de Operaciones y Operaciones, este último también se define desde el archivo .INI. Al lado derecho se muestra el nombre del usuario y un link para cerrar sesión.

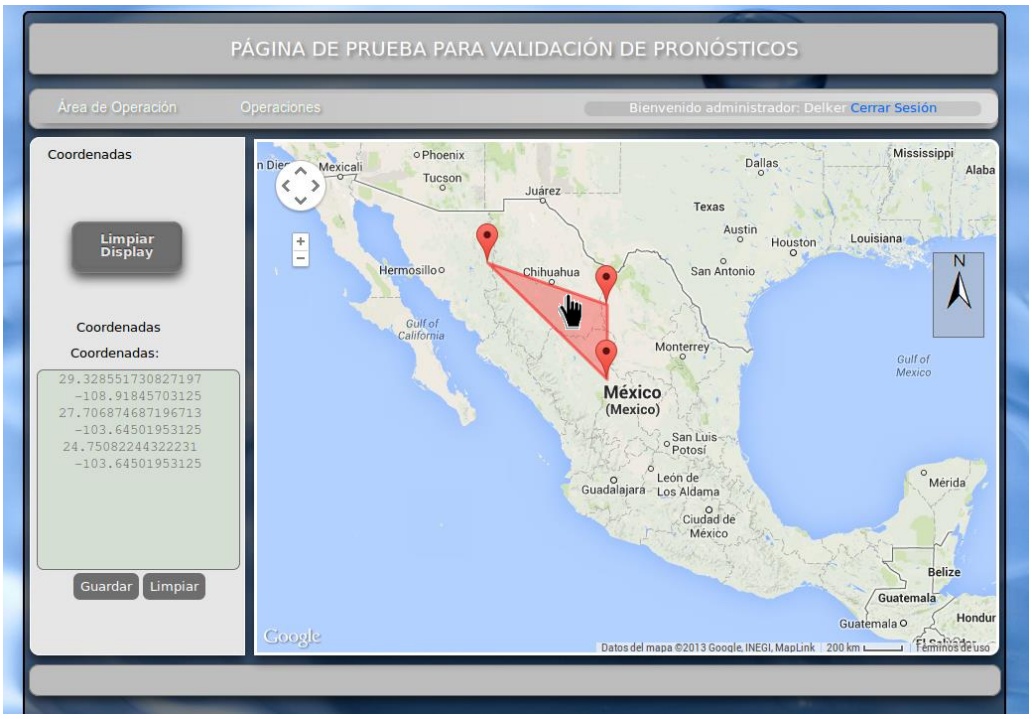
En la sección siguiente se muestra por default el mapa de la República Mexicana con la opción de generar un polígono de manera personalizada, esta sección cuenta con un panel de navegación y zoom para poder hacer más personalizado el polígono. En la sección del lado izquierdo se encuentra un formulario que captura las coordenadas. Y si el usuario lo desea, puede guardar estas en un archivo de texto (.txt), así como también, contiene un botón para limpiar el display donde se generan los polígonos.



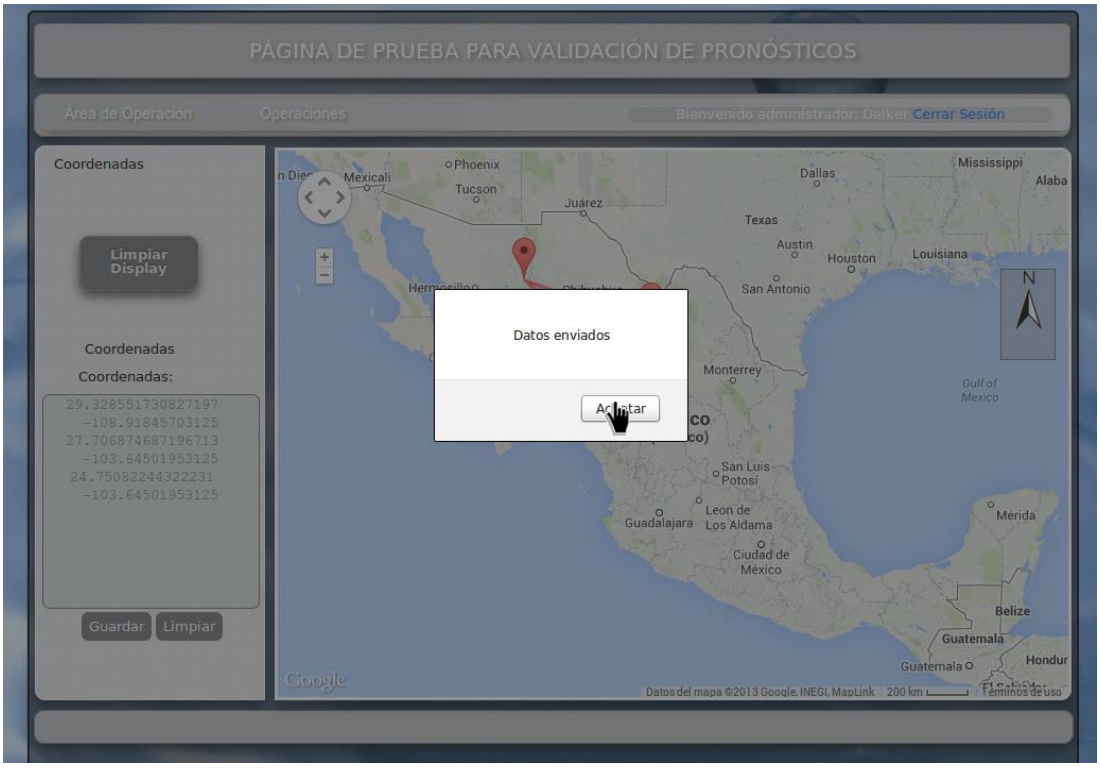
Para generar un poligono personalizado damos click izquierdo sobre el mapa y se van colocando puntos.

Ya colocados los puntos deseados se ve generando un polígono.

Al dar click derecho fuera del polígono este cambia de color para habilitar la posibilidad de extraer las coordenadas de sus vértices.

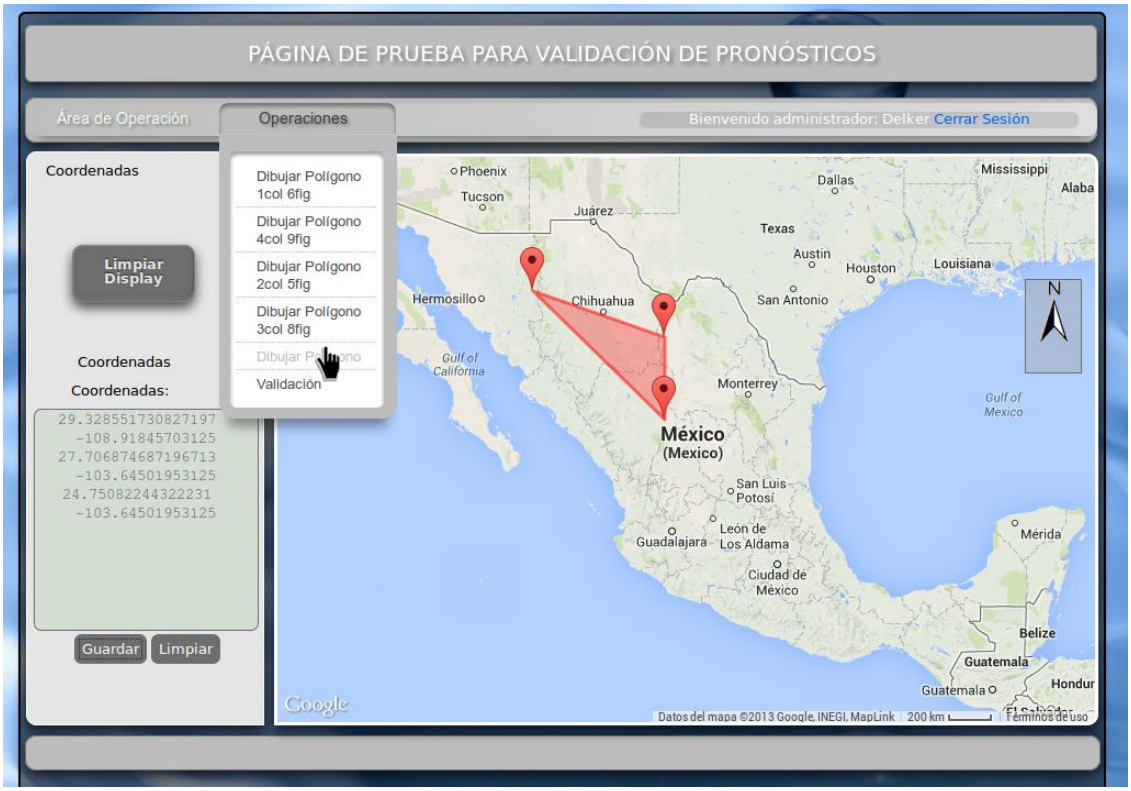


Al volver a dar click izquierdo dentro del polígono este vuelve a su color y extrae las coordenadas en una caja de texto, cabe mencionar que al dar click izquierdo en cualquier punto, estos se van eliminando, o se pueden arrastrar a otro lado para modificar el polígono.

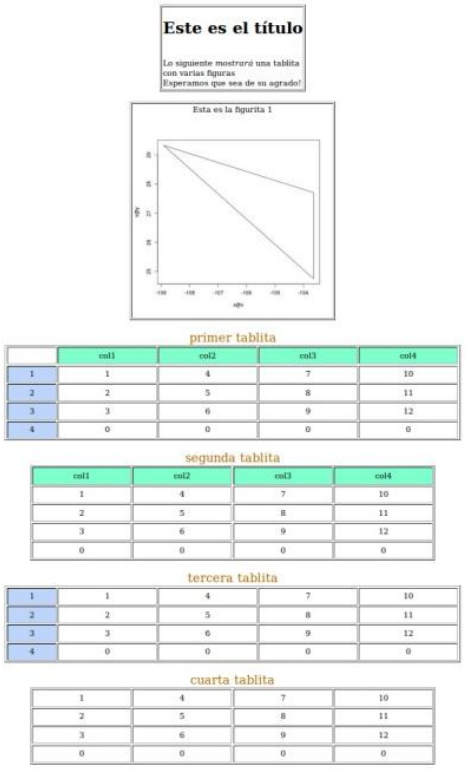


Al dar click en el botón “Guardar” se muestra un letrero de ”Datos enviados”, esto es que las coordenas que forman el polígono han sido enviadas a un archivo txt, damos click en Aceptar.

Al estar seleccionado el polígono, podemos ir al menú “Operaciones”

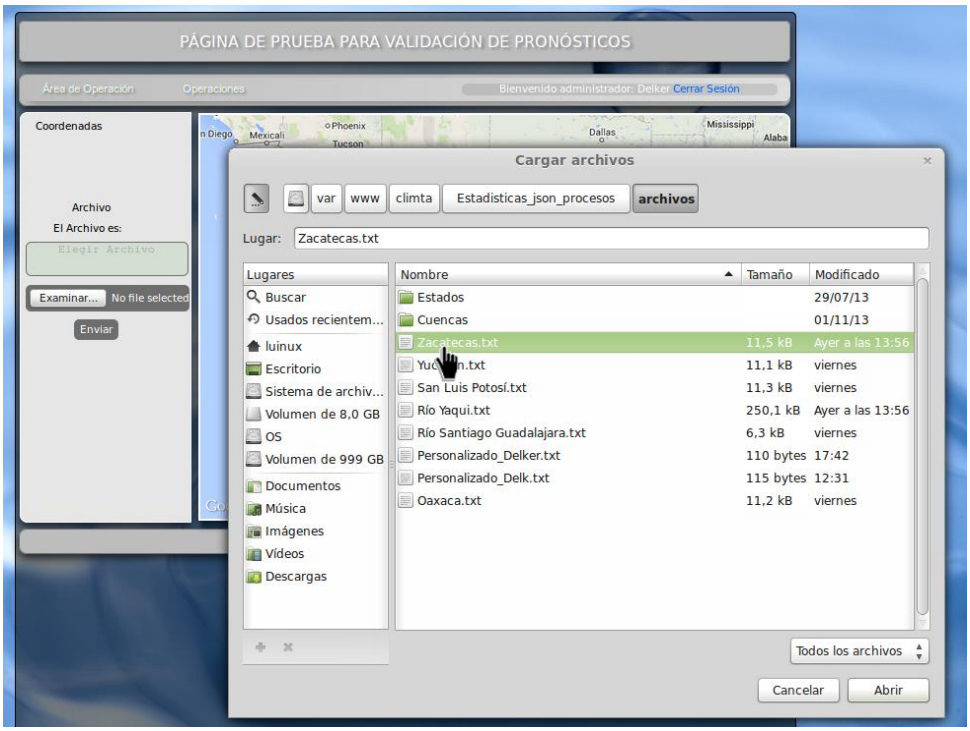


Al dar click en la operación que se desee, en este caso “Dibujar Polígono”.



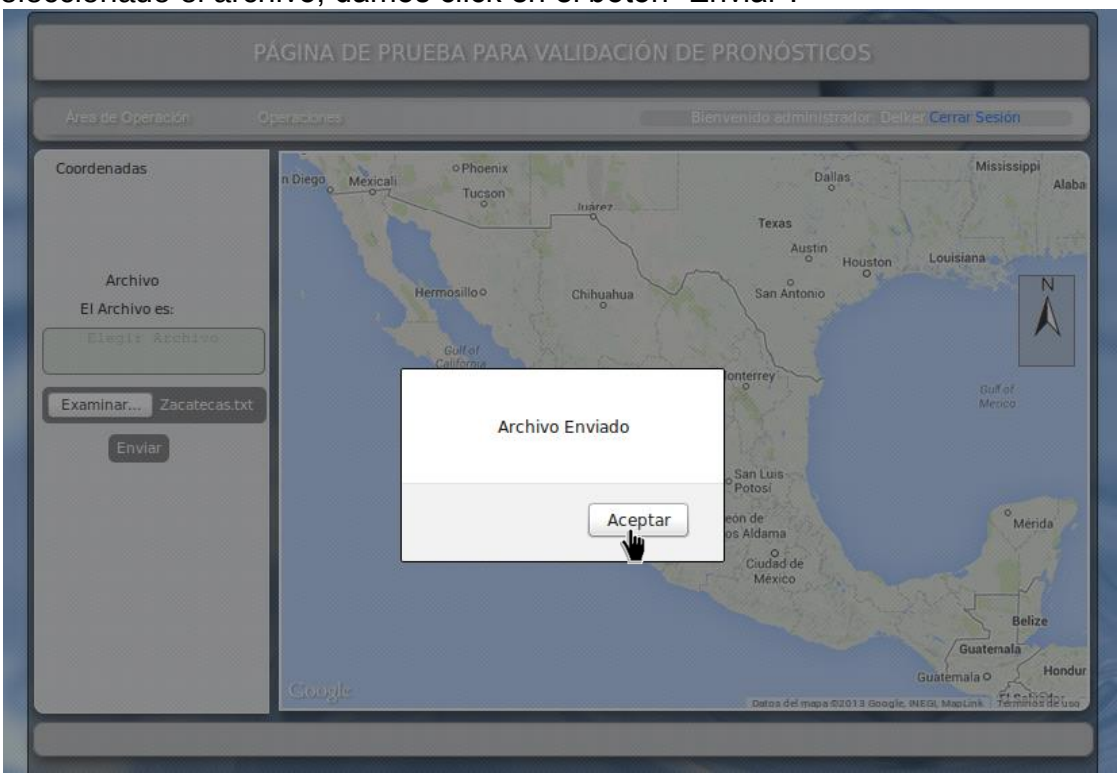
Se mostrará el resultado en una pestaña diferente.

Al dar click en el ítem “Predefinido” del menú “Área de Operaciones”, se mostrara un panel izquierdo donde se podrá elegir un archivo de texto con coordenadas de un polígono para ser mostrado en el mapa.



Al dar click en “examinar”, nos mostrara una ventana donde podremos elegir el archivo a subir, para mostrar el polígono.

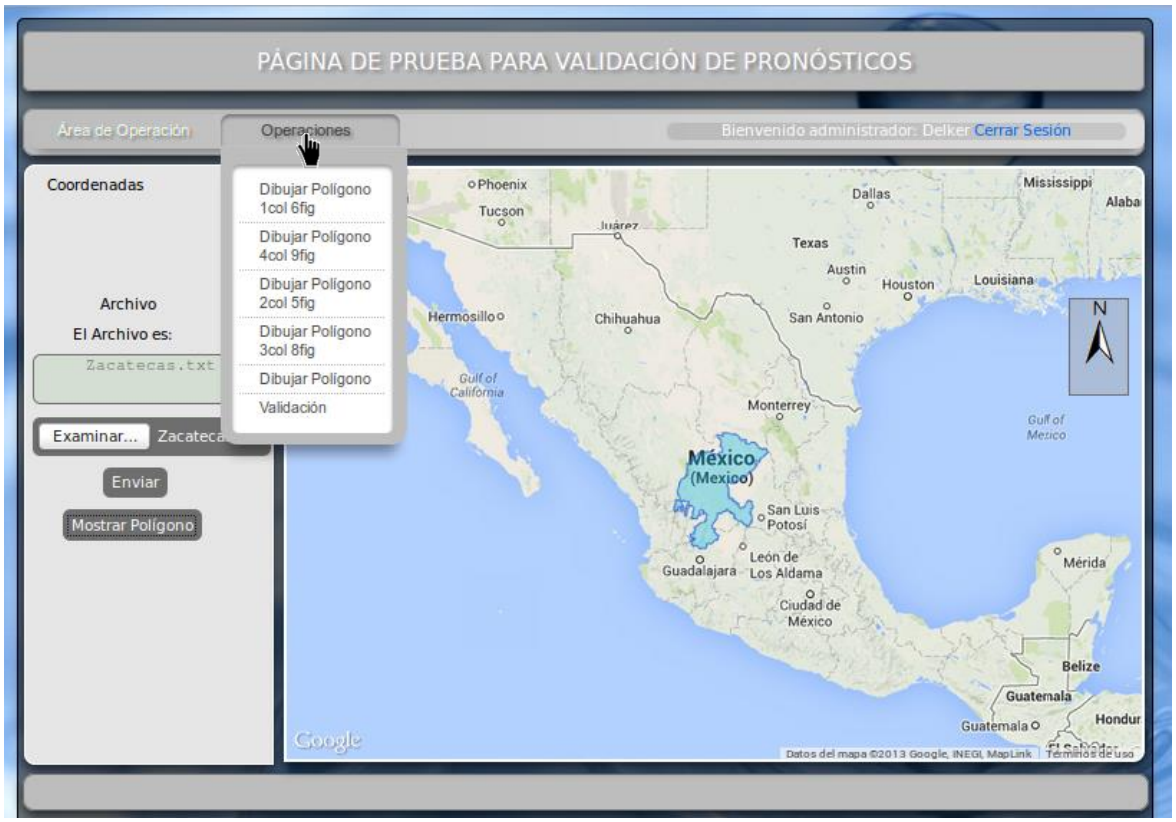
Ya seleccionado el archivo, damos click en el botón “Enviar”.



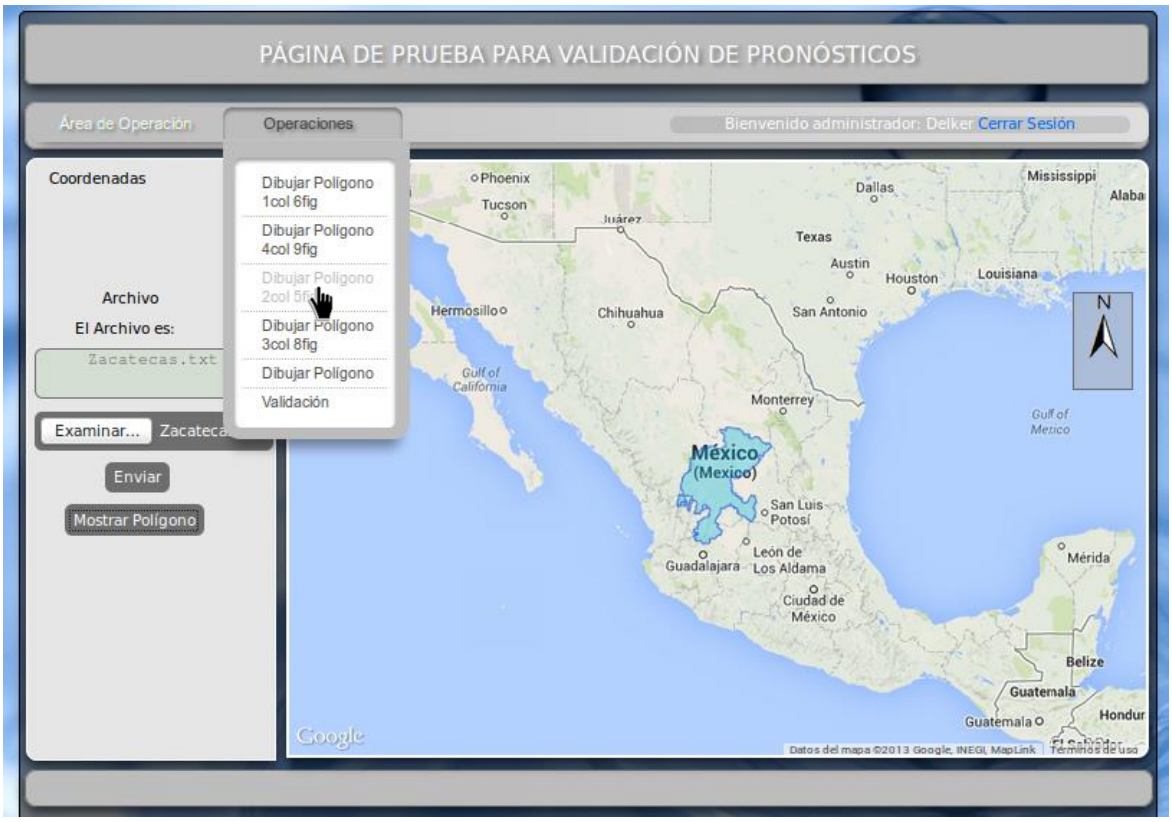
Se mostrará una ventana con el texto “Archivo Enviado” y damos click en “Aceptar”.

Si el archivo es cargado correctamente, se mostrara un botón con el texto “Mostrar Polígono”.

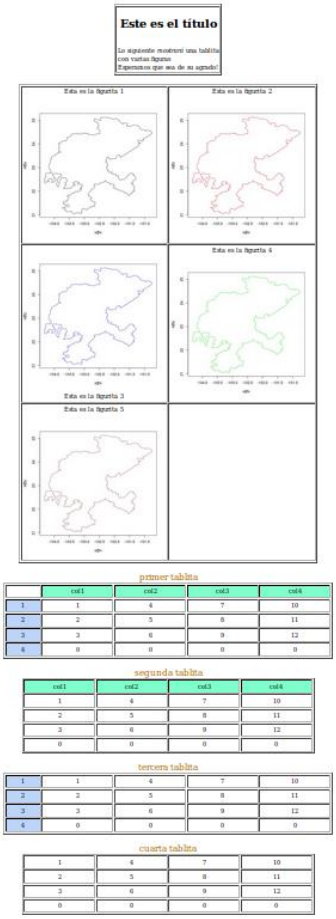
Al hacer click en el botón “Mostrar Polígono”, se visualizará en el mapa el polígono predefinido.



Al mostrarse el polígono, podemos ir al menú “Operaciones”.



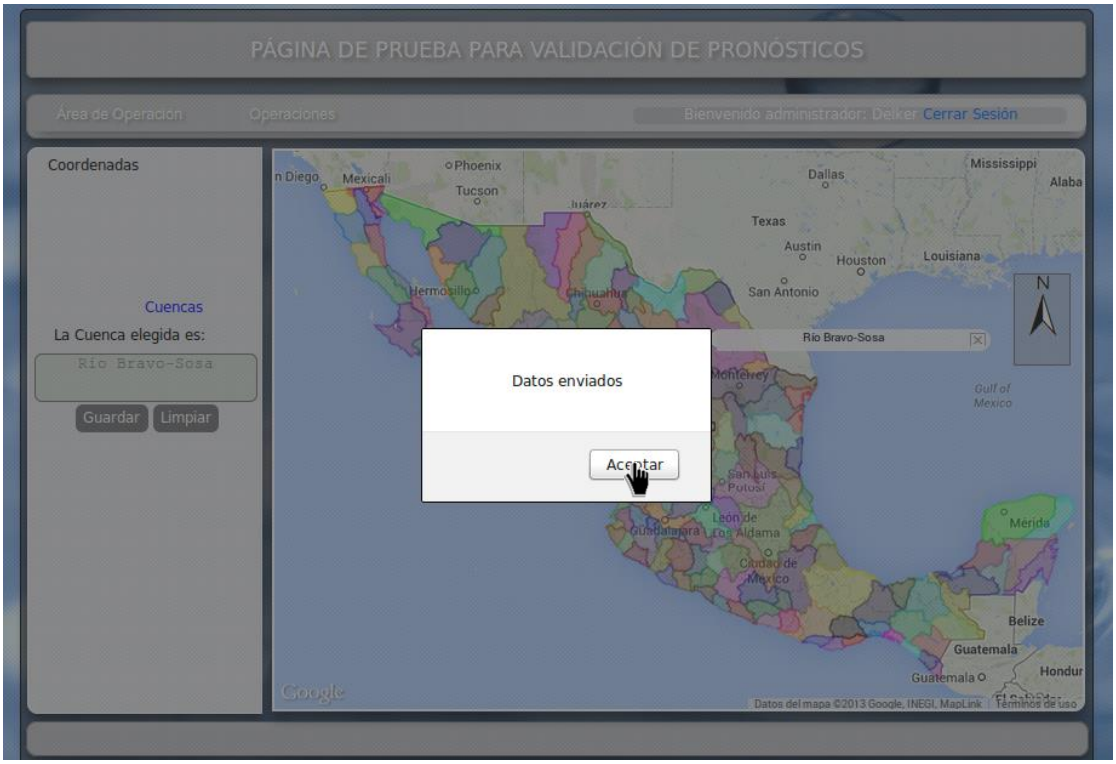
Al dar click en la operación que se desee, en este caso Dibujar Polígono 2col 5fig.



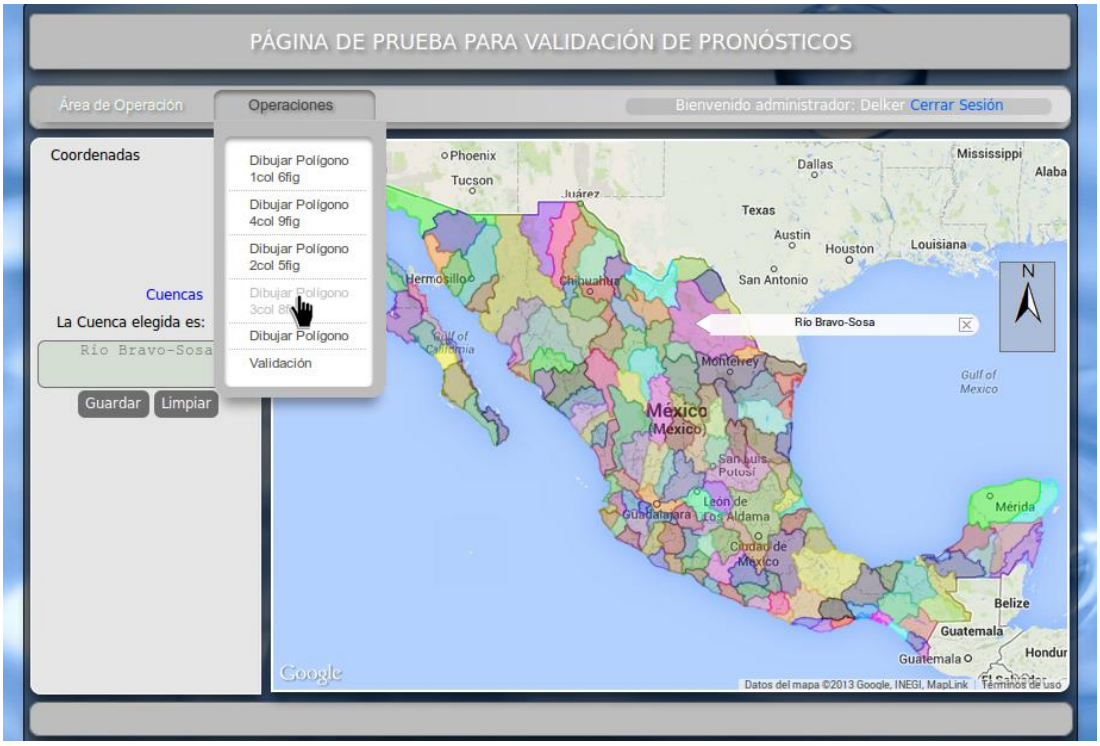
Se mostrará el resultado en una pestaña diferente.

Al dar click en el ítem “Cuencas” del menú “Área de Operaciones”, se mostrara un mapa de la República Mexicana con cuencas. En el panel izquierdo se muestra un formulario para que el usuario si lo desea guarde las coordenadas de cualquier cuenca.

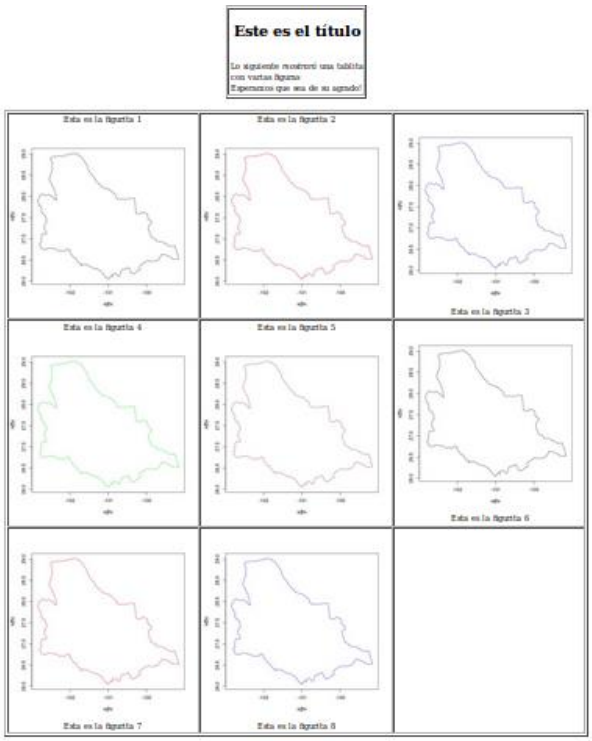
Al dar click sobre una cuenca, se mostrara en un globo el nombre de esta, también se mostrara en el cuadro de texto del formulario del panel izquierdo.



Al dar click en el botón “Guardar” se muestra un letrero de ”Datos enviados”, esto es que las coordenas de la cuenca seleccionada han sido enviadas a un archivo txt, damos click en Aceptar.



Al dar click izquierdo en la operación que se desee, en este caso Dibujar Polígono 3col 8fig.



primer tablita

	col1	col2	col3	col4
1	1	4	7	10
2	2	5	8	11
3	3	6	9	12
4	0	0	0	0

segunda tablita

col1	col2	col3	col4
1	4	7	10
2	5	8	11
3	6	9	12
0	0	0	0

tercera tablita

1	1	4	7	10
2	2	5	8	11
3	3	6	9	12
4	0	0	0	0

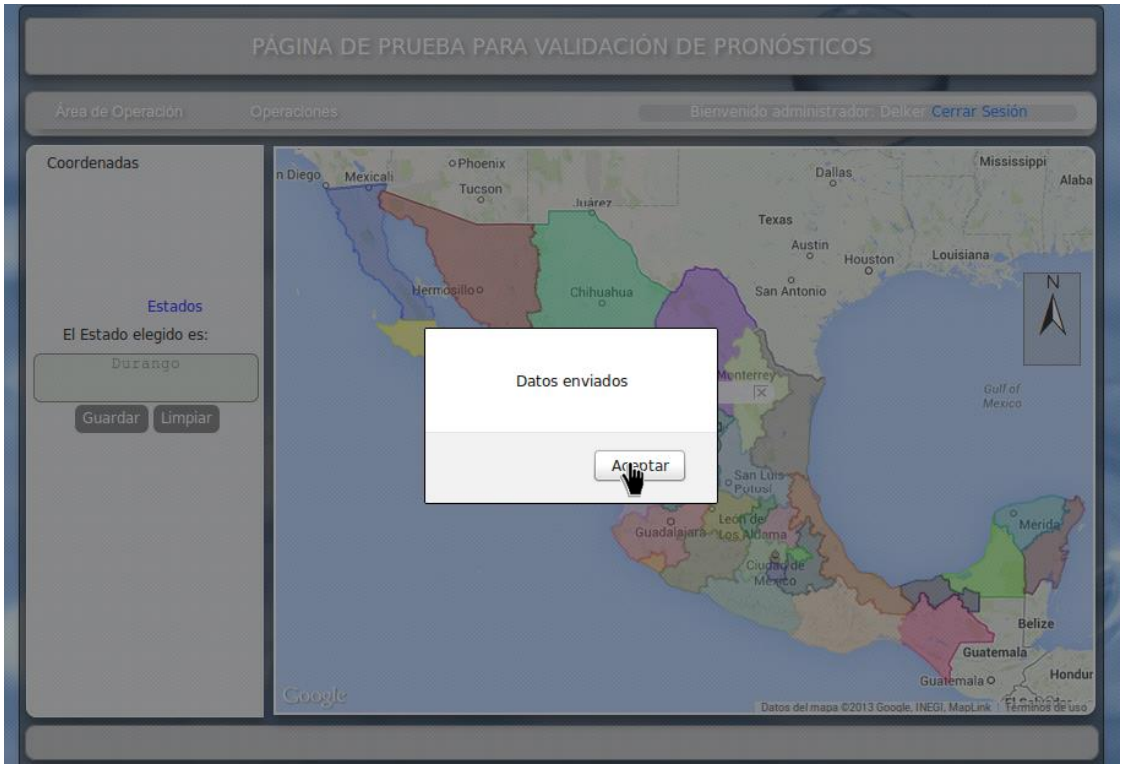
cuarta tablita

1	4	7	10
2	5	8	11
3	6	9	12
0	0	0	0

Se mostrará el resultado en una pestaña diferente.

Al dar click en el ítem “Estados” del menú “Área de Operaciones”, se mostrara un mapa de la República Mexicana con los estados que la componen. En el panel izquierdo se muestra un formulario para que el usuario si lo desea guarde las coordenadas de cualquier estado.

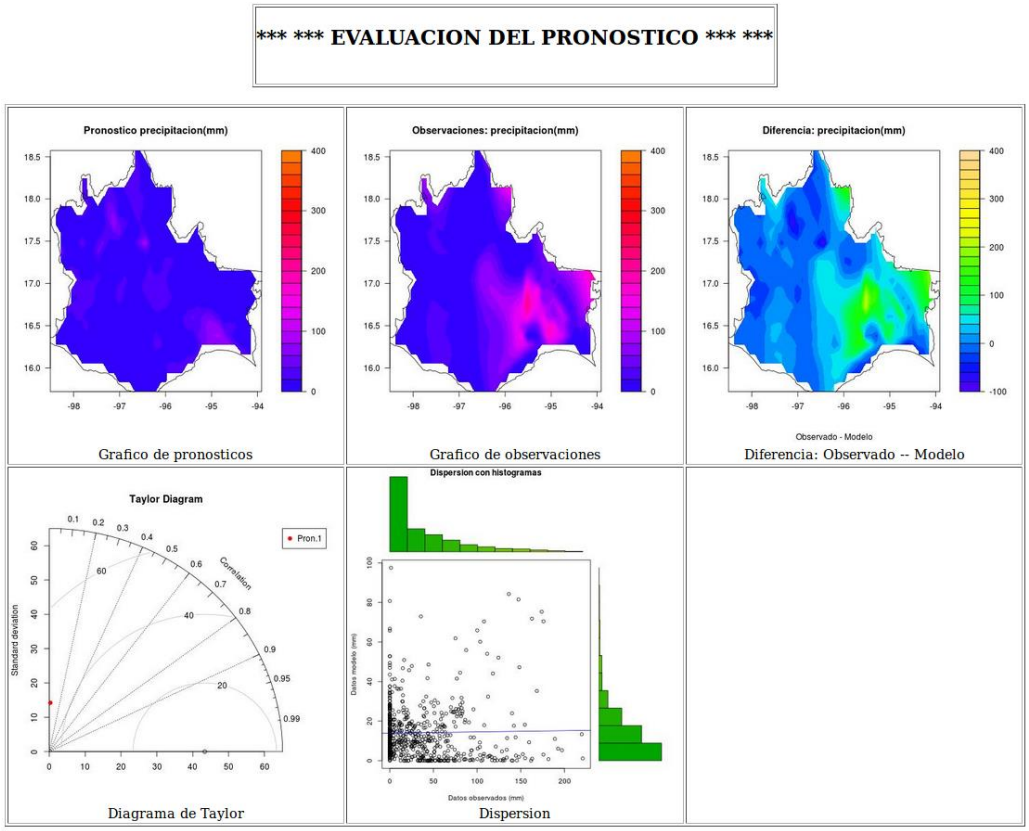
Al dar click sobre algún estado, se mostrara en un globo el nombre de este, también se mostrará en el cuadro de texto del formulario del panel izquierdo.



Al dar click en el botón “Guardar” se muestra un letrero de ”Datos enviados”, esto es que las coordenas de la cuenca seleccionada han sido enviadas a un archivo txt, damos click en Aceptar.

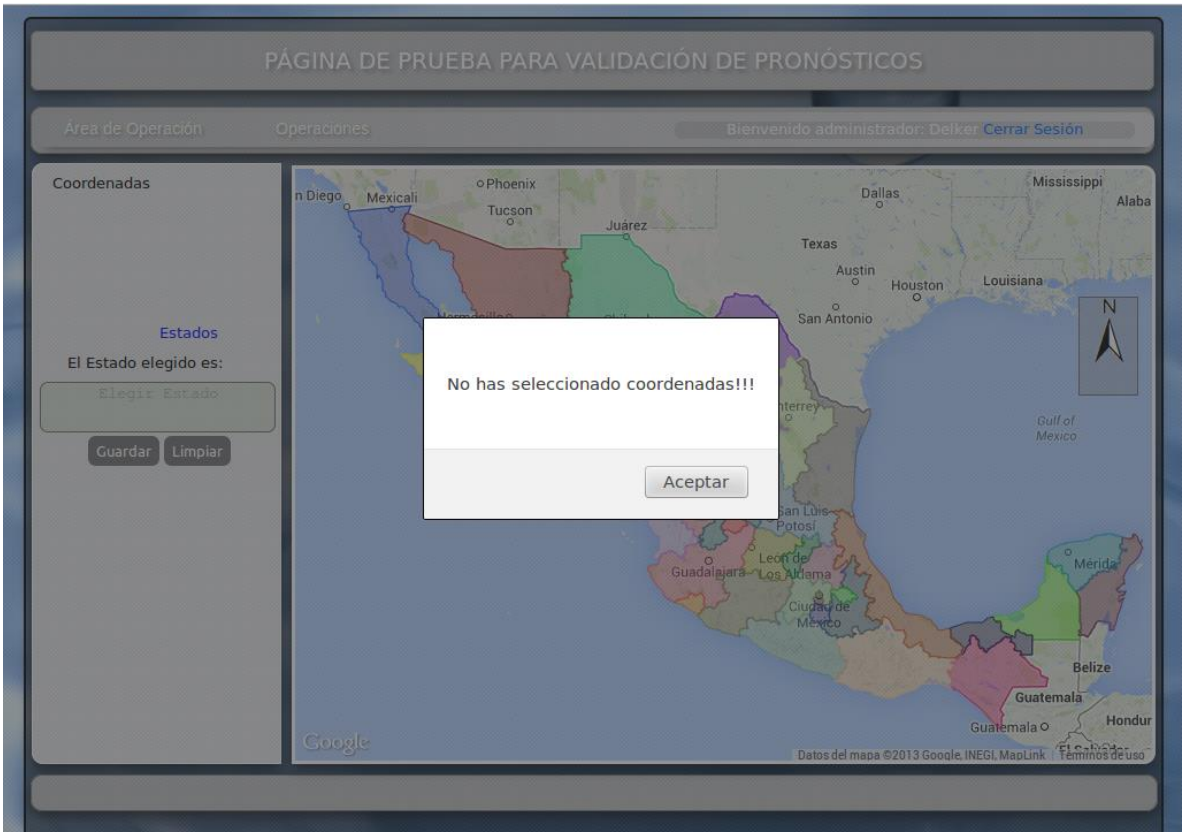
Al dar click sobre algún estado, en este caso Oaxaca.

Al dar click izquierdo en la operación que se desee, en este caso Validación.



Se mostrará el resultado en una pestaña diferente.

NOTA:
Si no se ha seleccionado un área (polígono) de trabajo, llámese Estado, cuenca, Predefinido o Personalizado y se desea hacer una operación, en todos los casos se mostrara el mensaje “No has seleccionado coordenadas”



ARCHIVO DE INICIO.

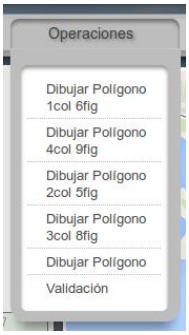
En la página web contamos con un archivo llamado procesos.ini del cual es esencial que el usuario conozca su estructura, para que lo pueda manipular dándole el uso correcto, ya que de este dependen el “titulo” que será mostrado y las “operaciones” que se realizarán en la página web.

Con una capacitación previa a la utilización, el usuario tendrá a bien conocer ciertos comandos que le serán de ayuda para poder estructurar operaciones que le servirán, ya sea para generar ciertas gráficas, textos, o tablas de apoyo para su página resultante. En el archivo .ini se tienen líneas de comandos, las cuales son de mucha importancia, puesto que en ellas se describen las operaciones que el usuario desee realizar y separadas de un signo “=” está escrito un comando que será interpretado para realizar ciertas operaciones.

```
1 [Titulo]
2 Titulo=PÁGINA DE PRUEBA PARA VALIDACIÓN DE PRONÓSTICOS
3 [palabras_clave]
4 Dibujar Polígono 1col 6fig=./MultiplesSalidas.R 1 6
5 Dibujar Polígono 4col 9fig=./MultiplesSalidas.R 4 9
6 Dibujar Polígono 2col 5fig=./MultiplesSalidas.R 2 5
7 Dibujar Polígono 3col 8fig=./MultiplesSalidas.R 3 8
8 Dibujar Polígono=./MultiplesSalidas.R
9 Validación=./Redirector.sh p24.nc EstacionesOAX20130922pp.txt
```

Tal como se observa, en la línea 2 tenemos la etiqueta “Titulo” que después del signo “=” tiene escrito el título de la página y que al ser leído al cargarse la página, será mostrado en la parte superior de esta.

A partir de la línea 4 se colocan las etiquetas que serán las operaciones que el usuario desee. Después del signo “=” tiene escritos los comandos que conforman dichas operaciones. Estas serán mostradas dentro del menú “Operaciones”.



EL LENGUAJE DE SALIDAS.

Dentro de la manipulación de las operaciones que se realizan en la página web, contamos con un lenguaje de salidas que el Usuario debe saber para que esta pueda identificar de manera ordenada y así realizar las operaciones pertinentes.

Ya que sin este lenguaje, la página resultante será incapaz de realizarse, puesto que está preparada para recibir **como identificadores de imágenes el siguiente texto estructurado:**

- ::DG “Número de columnas”:: (Indica que inicia un bloque de imágenes y el número de columnas que se deseen para su ordenamiento al ser mostradas).
- Nombre de la imagen.jpg (puede ser .gif ó .png).
- Nota de píe.
- Nota de encabezado.
- Nombre de la imagen.jpg (puede ser .gif ó .png).
- ::EDG:: (Indica que termina el bloque de imágenes).

Como identificadores de texto el siguiente texto estructurado:

- ::TX:: (Indica que inicia un bloque de texto).
- Puede tener las líneas de texto que se deseen.
- ::ETX:: (Indica que termina el bloque de texto).

Como identificadores de tabla el siguiente texto estructurado:

- ::TB, ”Separador (carácter, espacio o coma)”, ”Indicador de Encabezado(T, F)”, “Indicador de lomo (T, F)”,”Nombre de la tabla”:: (Indica que inicia un bloque de tabla, con o sin encabezado y con o sin lomo).
- Información de la tabla separada por el carácter colocado después de “TB”.
- ::ETB:: (Indica que termina el bloque de tabla).

11 ANEXO 3: Máquina computacional y el conjunto de operaciones estadísticas

Uno de los módulos que conforman la arquitectura del proyecto es la denominada *maquina computacional*. Ésta es un conjunto de códigos desarrollados en el lenguaje R que permiten realizar las operaciones solicitadas por el usuario. Para realizar su función debe interactuar con la *interfaz gráfica al usuario* y la *base de datos* (los otros dos módulos que forman el sistema, ver figura 1); el primero indica qué operaciones, y en qué región, según lo solicitado por el usuario, mientras que el segundo proporciona los datos que permiten la construcción de modelos estadísticos o según se requiera.

Visto como un sistema de entrada-salida, la máquina computacional tiene como entrada un archivo con valores correspondientes a información registrada por estaciones climatológicas ubicadas en la región de interés del usuario de alguna variable de interés (temperatura, precipitación, etc.). La salida es un archivo con valores que representan alguna variable de la estadística descriptiva o bien de la estadística inferencial (predicciones de un modelo aditivo generalizado), según haya sido solicitado por el usuario. Este archivo de salida es “proporcionado” a la interfaz gráfica al usuario con la finalidad de que sea procesada para ser presentada de manera adecuada. En el caso de que el usuario solicite una variable de la estadística descriptiva la salida será un número o un conjunto de números organizados en tablas o según sea conveniente. Si el usuario requiere de valores inferidos por un modelo estadístico, entonces tanto los archivos de entrada como salida de la máquina computacional poseen una estructura semejante y puede pensarse como una matriz de dos dimensiones de 4 columnas y *n* filas. En las tres primeras columnas se almacena información relativa a la ubicación geográfica de las estaciones climatológicas, en el caso del archivo de entrada, y en el de salida los datos de éstas representan un punto geográfico, y en la cuarta se registra el valor correspondiente de la variable de interés.

Como ha sido mencionado, la máquina computacional desarrollará cálculos de la estadística descriptiva e inferencial, sin embargo, en lo que resta de esta sección se abordará la parte de la estadística inferencial. En particular, esta componente de la máquina computacional la conforman los denominados *modelos aditivos generalizados*, los cuales representan una alternativa a una interpolación. De manera que se abordará parte de la teoría involucrada, la metodología empleada y el código en el lenguaje R que los genera. La diferencia más importante entre un modelo aditivo generalizado y una interpolación es que estimaciones e inferencia del primero se basan en la teoría de máxima verosimilitud mientras que el segundo, por lo general, en criterios como minimizar el error cuadrático medio.

De manera que el resto de esta sección se divide en las siguientes subsecciones: introducción, teoría, metodología; y el código en R de la máquina computacional.

11.1 Introducción

Los modelos aditivos generalizados pertenecen a la categoría de modelos lineales, para entenderlos es necesario revisar brevemente los modelos lineales y los modelos lineales generalizados. Los modelos estadísticos lineales son herramientas matemáticas que se emplean en la industria, en la ciencia, en el comercio, etc. Sus aplicaciones van desde el análisis de datos de internet hasta la búsqueda de asociaciones entre marcadores genéticos y enfermedades. De manera particular, se utilizan también para la predicción de variables climatológicas.

Los LM (acrónimo de *linear model*)son modelos estadísticos en los que la respuesta de una variable dependiente es modelada como la suma de un *predictor lineal* cuyo error de estimación posee media igual a cero. El predictor lineal depende de algunas variables independientes y algunos parámetros desconocidos. La característica principal de los modelos lineales es que el predictor depende linealmente de las variables independientes. Para realizar inferencias a partir de dichos modelos se asume que la respuesta de la variable dependiente posee una distribución normal.

Por otra parte, los modelos GLM (acrónimo de *generalized linear model*) permiten modelar una variable dependiente a través de una función monótona del predictor. Al mismo tiempo, la restricción sobre la distribución normal de la variable dependiente se relaja permitiendo que posea una distribución de otro tipo (como exponencial). Las inferencias hechas a partir de estos modelos se basan en teoría de probabilidad.

Finalmente, un GAM (acrónimo de *generalized additive model*) es un GLM en el cual una parte del predictor se especifica en términos de la suma de funciones suaves de las variables independientes. La forma algebraica de dichas funciones es desconocida así como el grado de *suavidad* apropiado. Para emplear los GAMs se requiere de algunas extensiones de los métodos empleados en los GLM.

11.2 Teoría

Para entender en qué consiste los GAM, es necesario revisar primeramente los LM y los GLM; comenzaremos con los LM. El objetivo de los LM es estudiar una variable dependiente *y* (que puede representar un evento físico) en términos de otra variable independiente *x*, a través de una relación matemática cuya media es μ y su desviación estándar es σ; se asume que la distribución de la variable *y* es normal. Las variables independientes (o covariables o predictores) son multiplicados por un coeficiente y posteriormente sumados lo que resulta en un predictor lineal, que adicionalmente

proporciona un estimado de los valores ajustados de la variable *y*. En términos matemáticos, con un LM se busca que:

$$\mu = \beta_0 + \beta_1 x + \beta_2 x_2 + \cdots \qquad \text{asumiendo } y = N(\mu, \sigma) \qquad (1)$$

Por otra parte los GLM permiten que la distribución de la variable *y* se extienda a la familia de las distribuciones exponenciales, además incorpora una función *g(.)* que relaciona la media μ (el valor estimado de los valores ajustados) con el predictor lineal βX (usualmente denotada por η). De modo que la expresión general de los GLM es: $g(\mu) = \beta X$.

Considere nuevamente los modelos LM. En éstos se asume que la distribución de la variable *y* es normal, que la varianza es la misma para todas las observaciones y que existe una relación directa entre el (los) predictor (es) y el valor esperado μ, es decir $\mu = \beta X$. De hecho los LM es un GLM de una variable con distribución normal y función *g(.)* igual a la identidad.

A manera de mostrar la generalización que representan los GLM, considere un conjunto de datos cuya distribución es tipo Poisson cuya media es μ (la varianza posee el mismo valor numérico que la media). Para este tipo de distribución la función *g(.)* es el logaritmo natural, de modo que en este caso la media se determina como lo indica la ecuación 2.

$$\mu = e^{\beta 0 + \beta 1 x 1 + \beta 2 x 2 + \cdots} \qquad (2)$$

Dicho lo anterior, si a los GLM se modifica la restricción sobre la forma algebraica que puede adquirir el predictor a permitir una dependencia no lineal, entonces el GLM se convierte en un GAM. En términos matemáticos los GAM se expresan como:

$$\mu = E(y) \quad g(\mu) = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots \qquad (3)$$

Una de las diferencias de los GAM respecto a los GLM es el uso de funciones suaves por medio de las que se establecerá la relación del predictor con la variable *y*. Note que también existe una función que relaciona la media (el valor estimado de los valores ajustados) con el predictor. Para los GAM también se asume que la distribución de la variable *y* es de la familia exponencial.

Un GAM es un modelo lineal generalizado con un predictor lineal construido por la suma de funciones covariables *suaves*. Una GAM es una extensión de los modelos aditivos de variables respuesta que pertenecen a la familia de distribuciones

exponenciales. La estructura algebraica puede expresarse como lo indica la ecuación 4.

$$g(\mu_i) = X_i^* \theta + f_1(x_{1i}) + f_2(x_{2i}) + \cdots \qquad (4)$$

siendo: $g(\mu_i) = E(Y_i)$, a su vez $Y_i \sim$ función de distribución exponencial.

En la ecuación 4 Y_i es la variable dependiente, X_i^* es una fila de la matriz de diseño correspondiente a las covariable que definen las componentes paramétricas del modelo, θ es el vector de coeficientes de regresión, x_i son las covariables y f_i son las funciones de las covariables. Las funciones de distribución que pertenecen a la familia exponencial son discretas (Bernoulli, binomial, Poisson) o absolutamente continuas (beta, Chi-cuadrado, Dirichlet, normal, etc). GAM permite que la relación de la variable dependiente con las variables independientes sea flexible, además al especificar al modelo en términos de *funciones suaves* en lugar de relaciones explícitas, es posible evitar modelos difíciles de manejar. No obstante esta flexibilidad viene acompañada de dos problemas: es necesario representar de alguna manera dichas *funciones suaves* y elegir "qué tan suaves deben ser."

El problema de estimar un GAM, es el problema de de determinar los parámetros de las funciones suaves y los coeficientes del modelos que maximizan la probabilidad penalizada, una vez que se ha elegido una base para las funciones suaves, junto con las mediciones pertinentes de la "variabilidad" de las funciones suaves. En la práctica el problema de maximizar la probabilidad penalizada se resuelve al determinar de manera iterativa los mínimos cuadrados cuyos pesos son re-calculados, mientras que los parámetros que determinan el "grado de suavidad" se estiman usando el criterio de validación cruzada.

Los GAMs pueden representarse usando funciones *spline* construidas por métodos de regresión penalizados y el grado de *suavidad* de las funciones f_i se determina a partir de las variables independientes empleando validación cruzada.

Es pertinente mencionar que los modelos aditivos son estimados por medio de mínimos cuadrados penalizados mientras que los GAM son estimados al maximizar la probabilidad penalizada.

FUNCIONES SUAVES DE UNA VARIABLE: FUNCIONES SPLINE

La regresión lineal es una de las técnicas estadísticas más empleadas en diversas áreas de la ciencia. Típicamente la regresión lineal consiste de una variable respuesta (variable independiente *Y*) cuyo valor esperado se modela en términos de un predictor lineal (covariables), una función paramétrica formada por un conjunto de variables independiente (covariables) que determinan el valor de la variable Y. Se asume que el predictor depende linealmente de los parámetros que lo definen.

En muchos casos los modelos lineales no se ajustan a los datos observados, por ello típicamente se extienden los modelos lineales agregando un predictor formado por un polinomio. Sin embargo, los modelos polinomiales son inadecuados en muchas situaciones. Este es el caso cuando se desean ajustar datos que corresponden a un fenómeno físico, el comportamiento de éstas en una región pueden no guardar relación alguna con lo que ocurre en otra región. En este sentido, los polinomios así como cualquiera otra función matemática poseen la característica opuesta, es decir, su comportamiento en una región pequeña determina su comportamiento en cualquier punto.

Las funciones spline son diferentes a las funciones suaves como polinomios, sinusoides, logarítmicas, etc. Esto se debe a que un spline esta formada por segmentos de polinomios unidas en los denominados *puntos de unión* (knots) y son continuas así como sus derivadas. Las spline han sido introducidas en la estadística como interpoladores, en este sentido, en muchos problemas estadísticos los datos disponibles poseen errores. Por lo tanto, en este caso, es deseable crear una función spline que "pase cerca de los puntos de interés" y que no tenga que pasar forzosamente por éstos. Esto se conoce como splines suavizantes y está relacionado con el problema de ajuste de curvas.

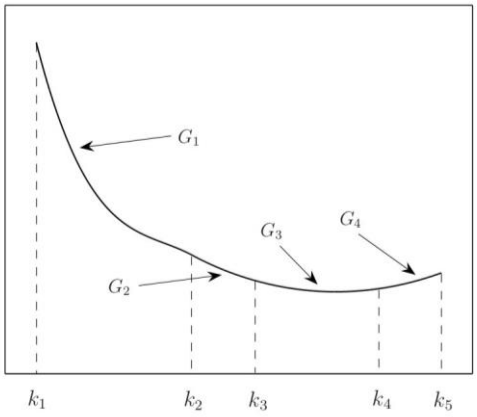


Figura 2. Spline formada por 4 segmentos de polinomios de orden 3 y 5 puntos de unión.

Una función spline (o simplemente spline) de grado *s* está formado por un conjunto de segmentos polinomios. Los segmento de polinomios (todos de orden *s*) se unen en los denominados *puntos de unión* k_m, $m = 1,2, \ldots k\$$. En la figura 2 se representa una spline formada por 4 segmentos de polinomios de orden 3 (*s=3*) con 5 puntos de unión. Por conveniencia se dará por hecho que el conjunto de puntos de unión $\{k_m\}_{m=1}^{k}$ está organizado ascendentemente, de modo que es posible escribir la spline como:

$$\mathrm{g}(x)=\mathrm{G_m}(x) = c_{0m}+c_{1m}\mathcal{X}+c_{2m}\mathcal{X}^2+\ldots+c_{\mathrm{S}m}\mathcal{X}^{\mathrm{S}}, \qquad \mathcal{K}_m \leq x < \mathcal{K}_{m+1} \qquad (5)$$

Los segmentos $G_m(x)$ se unen en los nodos satisfaciendo condiciones continuidad así como su *(s-1)*-ésima derivada, es decir:

$$\mathrm{G}_m^{(d)}(\mathcal{K}_{m+1})= \mathrm{G}_{m+1}^{(d)}(\mathcal{K}_{m+1}), d = 0,\ldots \mathrm{s}-1, m = 1, \ldots \mathcal{K}-2 \qquad (6)$$

Los parámetros que definen la spline son:

- El grado de la función polinomial, *s*.
- El número de puntos de unión, *k*.
- La posición de los puntos de unión, $\{k_m\}_{m=1}^{k}$.
- El número de coeficientes libres de la spline es $k+s-1$, si se desea satisfacer la ecuación 6; para aclarar este punto, note que cada uno de los $k-1$ segmentos de polinomios tienen $s+1$ coeficientes, y que las *s* condiciones de continuidad en cada punto de unión introduce $(k-1)s$ constricciones, lo cual resulta en $(k-1)(s+1)-(k-2)s = k+s+1$ coeficientes libres. De modo que una spline de grado s con k puntos de unión y $s-1$ derivadas continuas se "encuentra" en un espacio de $k+s-1$ dimensiones.

Un conjunto de splines muy empleado son los denominados *spline naturales*, los cuales se definen a partir de una condición. Si $s+1$, el orden de la spline, es par, entonces g es una spline natural si satisface la siguiente condición: g es un polinomio de orden $(s+1)/2$ fuera del intervalo $[K_1, K_k]$. Si se satisface esta condición y simultáneamente se satisface la ecuación 4, el número de parámetros libres se reduce a $k-2$.

La definición de una spline como se especificó por medio de la ecuación 5 es conveniente una vez que todos sus coeficientes son conocidos. Sin embargo, es más simple definir una spline de grado *s* con puntos de unión $\{k_m\}_{m=1}^{k}$ como una combinación lineal de unas \emph{funciones base}. Éstas forman un conjunto de splines linealmente independientes de grado *s* con puntos de unión $\{k_m\}_{m=1}^{k}$.

Típicamente se emplean dos bases para representar las spline: potencias truncadas y B-spline. Las primeras se manipulan fácilmente pero carecen de *estabilidad numérica*. Las segundas poseen "buenas propiedades numéricas" pero su manipulación analítica se vuelve compleja excepto en el caso especial en que los puntos de unión estén separados de manera equidistante.

SPLINES CÚBICAS

La importancia de las splines cúbicas radica en que por medio de éstas la función de interpolación que se construya será la "más suave" para cualquier conjunto de datos.

Dado un conjunto de puntos de unión existen diversas formas equivalentes de escribir la función spline. Una base que resulta de un análisis muy general de las funciones tiene como elementos las funciones: $b_1(x)=1$, $b_2(x)=x$, y $b_{i+1}(x)=R(x,x_i^*)$ (se hará referencia a esta base como CSW) para $i=1..,q-2$ en la que:

$$\mathrm{R}(x,z)=\left[(z-\tfrac{1}{2})^2-\tfrac{1}{12}\right]\left[(x-\tfrac{1}{2})^2-\tfrac{1}{12}\right]/4-\left[(|x-z|-\tfrac{1}{2})^4-\tfrac{1}{2}(|x-z|-\tfrac{1}{2})^2+\tfrac{7}{240}\right]/24 \qquad (7)$$

Si se utiliza esta base para construir a la función g, entonces la i-ésima fila de la matriz de diseño está dada por:

$$X_i=\left[1,x_i,R(x_ix_1^*)+R(x_ix_2^2)+\cdots R\left(x_ix_{q-2}^2\right)\right] \qquad (8)$$

Y la variable independiente es $y=X\beta+\epsilon$ y puede estimarse por medio de la técnica de mínimos cuadrados.

Considere la figura 3 que muestra el perfil de cada elemento de la base CSW en el intervalo [0,1], siendo $x_1=1/6$, $x_2=3/6$ y $x_3=5/6$, $\beta_1=1$ $(i=1,2,3,4)$ y también la función que se construye sumado todos los elementos de la base.

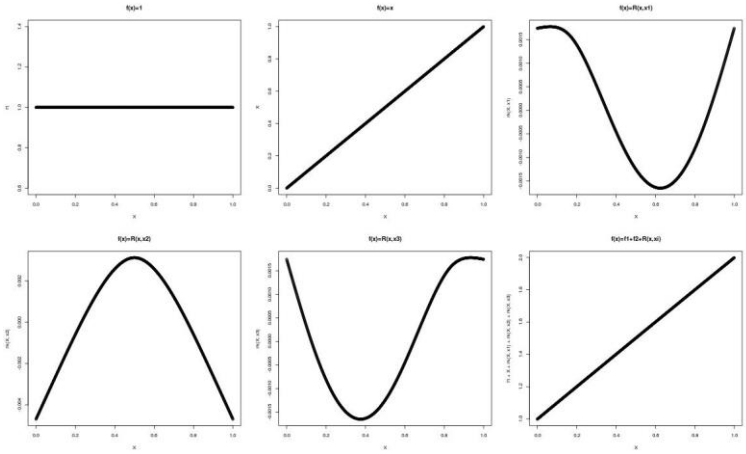


Figura 3. Representación de los elementos de la base expuesta en esta sección junto con la función que se construye a partir de ésta cuando los coeficientes $\beta_i=1$.

Al modificar apropiadamente los coeficientes β_i es posible modificar el perfil de la función g tal y como se muestra en la figura 4; en esta figura $\beta_1=4$.

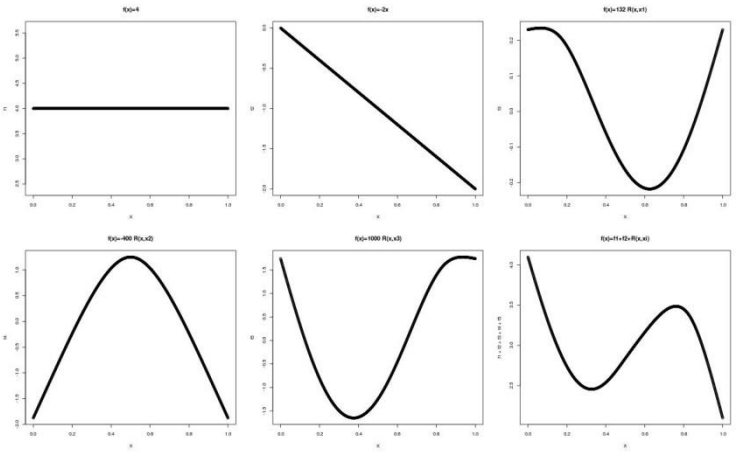


Figura 4. Representación de los elementos de la base expuesta en esta sección junto con la función que se construye a partir de ésta cuando los coeficientes β_i son diferentes entre sí y diferentes de 1.

A manera de ejemplo de aplicación de la base CSW, considere la figura 5 que muestra una nube de datos que representan: en el eje de las ordenadas la temperatura promedio y en el eje de las abscisas el "tiempo normalizado" (cada punto del eje de las abscisas corresponde un día del mes de enero de 2010 registrado por la estación INIFAP, al que se le ha sustraído 1 y dividido entre 31). Al implementar un modelo lineal con estos datos, se encuentra que la recta de ajuste está dada por la ecuación $y = 16.77 - 3x$ (siendo x el tiempo normalizado e y la temperatura) cuyo MSE es 161.7; la figura 6 muestra de manera gráfica éste resultado.

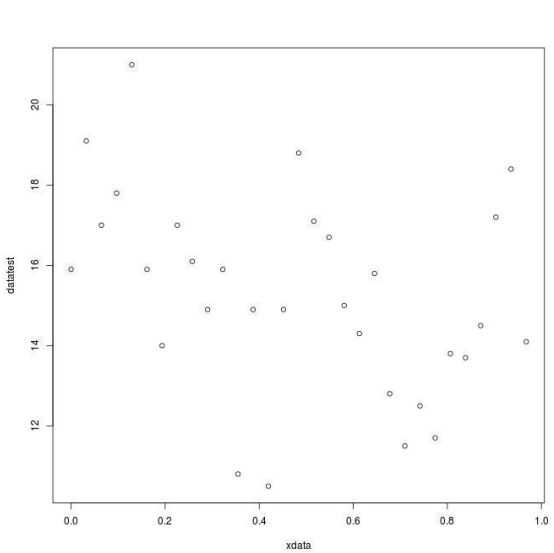


Figura 5. Nube de datos que representan la temperatura promedio

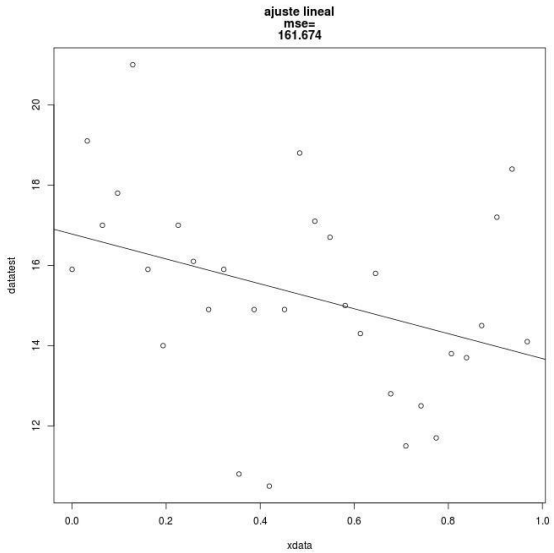


Figura 6. Recta de ajuste lineal correspondiente a los datos mostrados

durante el mes de enero de 2010 en la figura 5 registrada por la estación INIFAP.

Ahora bien, si se elige un modelo basado en splines cúbicas y en particular se emplea la base CSW, la curva que se obtiene cambia radicalmente cualitativa y cuantitativamente. Los puntos de unión seleccionados son $\{k_m\}_{m=1}^{k}$ =0.01, 0.11, 0.21, 0.31, 0.41, 0.51, 0.61, 0.71, 0.81, 0.91 y los coeficientes resultantes son: 17.2, -3.9, 12998.2, 89143.6, 9665.3, 105570.4, -20276.2, 121169.4 13247.8, 78763, 4520.1, 108038.2. Con relación a la parte cuantitativa el MSE de este modelo es 65.5, es decir, representa apenas el 40% del MSE correspondiente al ajuste lineal. La representación de la spline y los datos se muestran en la figura 7. De modo que en este ejemplo, el modelo basado en splines describe mejor a la temperatura promedio registrada por la estación INIFAP en enero de 2010.

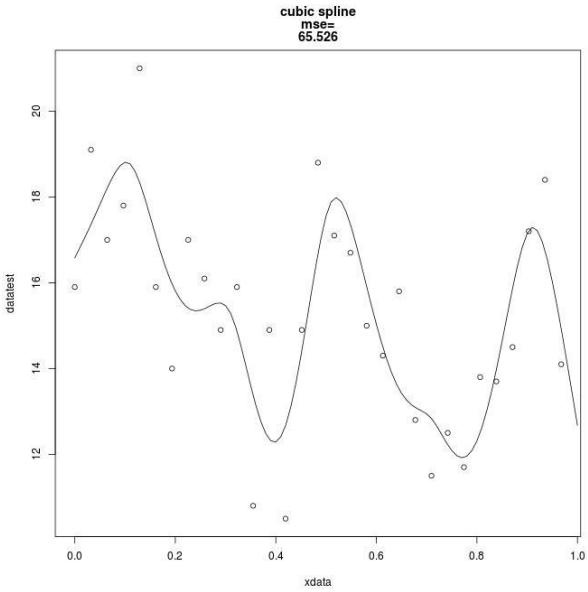


Figura 7.Spline de ajuste correspondiente a los datos mostrados en la figura 5.

Si bien la spline cubica se ajusta “bien” a los datos, el grado de “suavidad” de la curva está determinada por el número de puntos de unión $(+2)$ el cual es arbitrario.

SPLINES CÚBICAS PENALIZADAS

Las técnicas empleadas para determinar el “suavizado” en un modelo que usa splines se basa en dos aproximaciones: splines de regresión y splines suavizantes. Las splines de regresión se definen empleando un número pequeño, pero cuidadosamente seleccionado, de puntos de unión (knots), lo cual garantiza que la función sea suave. La ubicación de los puntos de unión en el dominio de la curva a ser estimada juega un papel crucial, esto es porque en las regiones de mayor flexibilidad de la función “real” deben asignarse más puntos de unión que en otras regiones.

Las spline suavizantes surgen como un problema de optimización. Schoenberg define la “suavidad” de una curva a través del cuadrado de de la *d*-ésima derivada de la curva.

La solución al criterio de *la suma de cuadrados de residuos penalizados* es una spline de orden $2d-1$ con un punto de unión en cada punto de diseño. No obstante, el número de parámetros que deben ser estimados es tan grande como el número de puntos de diseño, lo cual se traduce en altos requerimientos de cómputo.

Las splines penalizadas (p-spline) pueden considerarse como una aproximación que busca un compromiso entre las spline de regresión y suavizantes. Su idea básica consiste en representar la curva g por medio de una spline sobre-estimada, y controlar la "suavidad" sustrayendo términos penalizados de la función de probabilidad. El número de puntos de unión es mucho menor que el número de observaciones, de modo que los p-splines son más eficientes que las spline suavizantes desde el punto de vista computacional.

De modo que en lugar de ajustar el modelo minimizando la cantidad $\left||y-X\beta|\right|^2$ (como se hace en las otras splines) se busca minimizar $\left||y-X\beta|\right|^2+\lambda\int[f''(x)]^2dx$ en el que el integrando "penaliza" aquellas secciones "muy curvas". El grado de suavidad se controla por medio del parámetro λ. Si $\lambda\to\infty$ entonces g es una línea recta, por otra parte si $\lambda=0$ entonces g es una curva no penalizada.

Dado que la función g es lineal respecto a los parámetros β_i, la penalización puede escribirse como: $\int[g''(x)]^2dx=\beta^TS\beta$, siendo S una matriz de coeficientes conocidos. De hecho al emplear la base CSW $S_{i+2,j+2}=R(x_i^*,x_j^*)$ para $i=1,\dots q-2$ y las primeras dos columna de S son cero.

De manera que, el problema de regresión con spline penalizadas se traduce en minimizar la cantidad:

$$\left||y-X\beta|\right|^2+\lambda\beta^TS\beta, \qquad (9)$$

respecto a β. El problema de determinar el grado de "suavidad" del modelo se convierte ahora en el problema de determinar el parámetro λ; por el momento se determinará β dado una valor de λ. De hecho el valor de β que minimiza la cantidad 9 es:

$$\hat{\beta}=(X^TX+\lambda\,S)^{-1}X^Ty \qquad (10).$$

La mayoría de los modelos que emplean p-splines consideran un parámetro de penalización como medida de "qué tan burda" es la curva g. No obstante estos modelos presentan limitaciones relativos a la correcta especificación de los límites de "suavidad".

SPLINES CÚBICAS

A fin de estimar g es necesario poder representar la ecuación 5 como un modelo lineal. Esto se puede realizar eligiendo una *base*, esto es, definiendo un espacio de funciones del cual g (o una aproximación de ella) sea elemento. Elegir una base implica elegir un conjunto de funciones que se da por hecho que se conoce: si β_i es el coeficiente de la b_i-esima base entonces $f(x)$ está dado por la ecuación 11..

$$f(x) = \sum_i b_i(x)\beta_i \qquad (11)$$

Suponga que g pertenece al conjunto de polinomios de orden 4, de modo que la base $b_i(x)$ estaría dada por:$b_1(x) = 1$, $b_2(x) = x$, $b_3(x) = x^2$, $b_4(x) = x^3$ y $b_5(x) = x^4$, por lo que $f(x)$ se expresaría como:

$$f(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3 + \beta_5 x^4, \qquad (12)$$

por lo que la variable dependiente y_1 se convierte en

$$y_i = \beta_1 + \beta_2 x_i + \beta_3 x_i^2 + \beta_4 x_i^3 + \beta_5 x_i^4 + \varepsilon_i \qquad (13)$$

A manera de ejemplo, en la figura 8 se muestra el gráfico de los elementos de la base propuesta, así como el perfil de la función resultante.

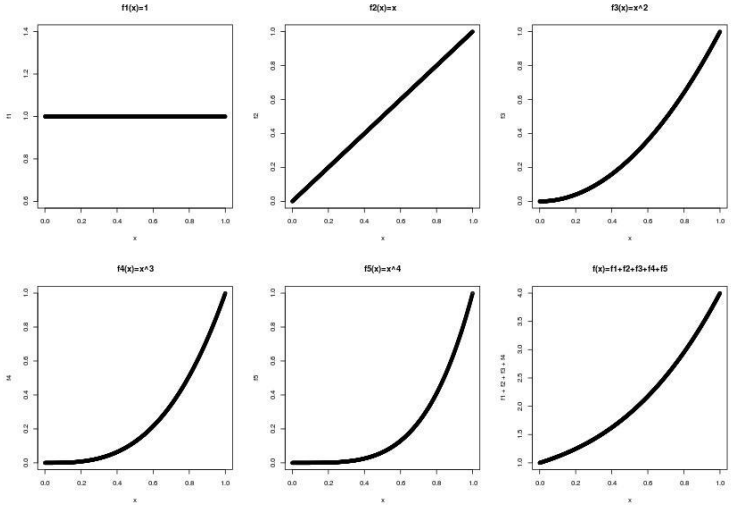


Figura 8. Representación de los elementos de la base formada por los polinomios de orden menor que 4 junto con la función que se construye a partir de ésta cuando los coeficientes son iguales a 1 ($\beta_i = 1$).

Ahora bien, si modificamos los coeficientes de cada elemento de la base podremos construir una gráfica con un perfil diferente al de cualquiera de las bases, como muestra de ello considere la figura 9.

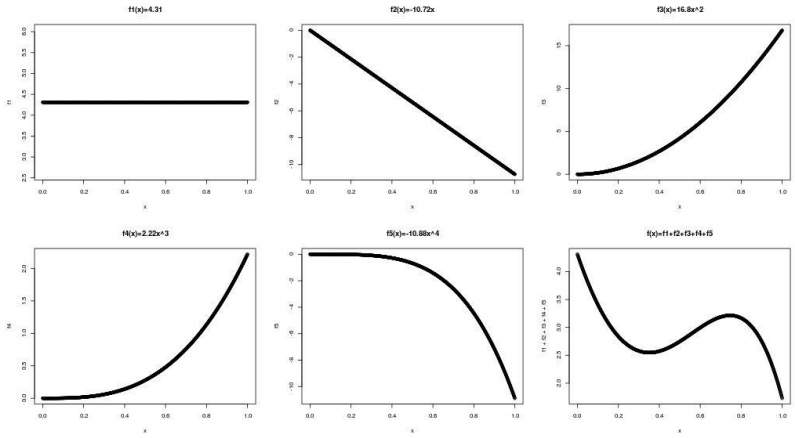


Figura 9. Representación de los elementos de la base formada por los polinomios de orden menor que 4, cada uno con un coeficiente diferente, y la función resultante.

La base formada por los polinomios son muy útiles en casos en los que se desea investigar las propiedades de la función g en la vecindad de un punto específico, no obstante cuando se desea investigar en todo el dominio de éstas base se presentan algunos problemas. Una de las aplicaciones más importantes de las spline es el de interpolación; de hecho los spline son las funciones más suaves para ésta labor.

Con relación a la construcciones de funciones ($f(x_i) = y_i$) que permitan interpolar un conjunto de puntos (x_i, y_i), en ocasiones es más útil *suavizar* que interpolar. En este sentido las spline son la mejor opción para construir funciones suaves que pasen cerca de los puntos de interés. Dicho con otras palabras, en lugar de construir una función $f(x_i) = y_i$ puede resultar más conveniente considerar $g(x_i)$ como un parámetro libre que conforma la spline y estimarlo de manera tal que

$$\sum_i [y_i - f(x_i)]^2 + \lambda \int f''(x)^2 dx, \qquad (14)$$

sea mínimo. En esta ecuación λ es un parámetro que puede ajustarse el peso relativo que se le confiere a los puntos que se desean suavizar.

Al construir una splines que suaviza, se encuentra que éstas tienen tantos parámetros libres como datos que se desean suavizar. No obstante, en la práctica, el número de parámetros libres es lo suficientemente grande de tal modo que la spline resultante es "más suave" que el sugerido por el número de grados de libertad. De hecho muchos grados de libertad se suprimen mediante la penalización (más adelante se explicará este concepto). Para construir una función suavizante de una covariable usando splines, un gran número de parámetros libres no representa un problema serio, sin embargo lo es cuando el número de covariables es más de uno.

Una manera de conservar las propiedades que hacen atractivas a las spline al mismo tiempo que se mantiene al mínimo los requerimientos de tiempo de cómputo, es utilizando las splines de regresión penalizadas.En su versión más simple, se debe construir una base de splines (con sus respectivas penalizaciones) para un conjunto de datos mucho menor al que se desea analizar, y después usar esta base (más las penalizaciones) para modelar el conjunto original de datos. Los valores de las covariables en el conjunto de datos pequeño deben organizarse de tal manera que abarquen los valores extremos del conjunto original de datos.

Con la base teórica expuesta se procedió a plantear la metodología que permitirá construir los modelos aditivos generalizados.

11.3 Metodología

Con la finalidad de implementar los modelos aditivos generalizados para las variables meteorológicas de interés (temperatura, precipitación, etc.), se estableció la metodología que permite construir GAMs con diferentes parámetros. Se establecieron algunos parámetros que permitirán determinar cuál de todos los GAMs generados es el adecuado.

La metodología a implementar se presenta a continuación:

- Elegir un conjunto de valores correspondientes a alguna variable meteorológica de interés (por ejemplo la precipitación diaria medida en mm ---prec---) y algunos conjuntos de variables independientes (por ejemplo, la ubicación geográfica, temperatura, etc.). A los valores de la variable meteorológica de interés se hará referencia como *variable dependiente*. Al conjunto de variable dependiente más variables independientes se denominará *datos*. Repetir los siguientes pasos para cada variable dependiente.
- Revisar los datos en busca de valores que se ubiquen fuera de un cierto intervalo de "credibilidad".
- Los modelos que se construirán considerarán cambios en 4 parámetros: la función de distribución (**pdf**=N ---Normal--- ó G ---Gamma---), la dimensión de la

base (**k**=50 (1D), 100 (3D), 200 (3D)), la dimensión de la función suave (*f(x)+g(y)+h(z)* ó *f(x,y,z):* **dm**=1 ó 3) y la base (**B**=T ---tp--- ó cr). Se construirán 12 modelos que corresponden a todas las combinaciones posibles de los valores de los parámetros considerados. Los modelos a construir y sus parámetros se listan en la tabla 1.

Tabla 1 Modelos a construir

Modelo	pdf	k	dm	B	Modelo	pdf	k	dm	B	Modelo	pdf	k	dm	B
gm0	N	50	1	T	gm4	N	3	200	T	gm8	G	100	3	T
gm1	N	50	1	cr	gm5	N	3	200	cr	gm9	G	100	3	cr
gm2	N	100	3	T	gm6	G	1	50	T	gm10	G	200	3	T
gm3	N	100	3	cr	gm7	G	1	50	cr	gm11	G	200	3	cr

- Al construir cada modelo deberá registrarse información que formará parte de los resultados. La información consiste en datos numéricos y gráficos. Se empleará el comando “gam.check(gmx)” para extraer los parámetros **edf** (effective degrees of fredom) y **gcv** (generalized cross validation). Estos dos parámetros pueden considerarse como parámetros de control: posiblemente se descarten aquellas variables cuyo **edf** sea menor que la unidad, y por su parte el parámetro **gcv** deberá ser lo más pequeño posible.
- Generar la gráfica “sample quantiles vs theoretical quantiles” empleando el comando: “qqnorm(residuals(gmx))”. Si el perfil de ésta no se ajusta a una recta, entonces la elección de la distribución de la variable independiente es incorrecta.
- Generar la gráfica “residuals vs fitted values” empleando el comando: “plot(fitted(gmx),residuals(gmx))”. La distribución de los puntos de ésta debe ser uniforme
- Con la finalidad de hacer predicciones, construir una matriz con valores de las variables independientes diferentes de lo que se usaron para construir el GAM . Si “pd” es la matriz que contiene dichos valores, entonces las predicciones se realizan por medio del siguiente comando:

  “zz<-predict(gamx,newdata=pd,type='response',se=T)”
- Suponiendo que se desea mostrar una gráfica en el plano (*lon, lat*), y si *zz* es una matriz de dimensiones apropiadas, entonces es posible construirla mediante los siguientes comandos:

  - layout(matrix(c(1,2), nrow=1), widths=c(6,1))
  - par(mar=c(5,5,3,1))
  - image(lon,lat,zz,col=gray(0:32/32),cex.lab=1.5,cex.axis=1.4)
  - contour(lon,lat,zz,add=T)
  - lines(coast$lon,coast$lat,col=1)
  - par(mar=c(1,2,0.5,0.5))
  - image.scale(zz,col=gray(0:32/32),horiz=FALSE)

- Los modelos que asumen una función de distribución Gamma son de la forma: *log(E[Y])=f1(x)+f2(y)+...*, por lo que al calcular los valores de la variable de interés debe realizarse la operación: *exp(Y)*, siendo Y el valor predicho por el GAM.
- Suponiendo que se desea modelar la variable **Prec**, que se cuenta con tres variables independientes (**lat, lon**, y **alt)** y que estos últimos se agrupen en una base de datos denominada **datos**, los comandos que permiten contruir los modelos se listan a continuación:

  - gm0<-gam(Prec~s(lat,k=50)+s(lon,k=50)+s(alt,k=50),data=datos)
  - gm1<-gam(Prec~s(lat,k=50,bs='cr')+s(lon,k=50,bs='cr')+ s(alt,k=50,bs='cr'),data=datos)
  - gm2<-gam(Prec~s(lat,lon,alt,k=100),data=datos)
  - gm3<-gam(Prec~s(lat,lon,alt,k=100,bs='cr'),data=datos)
  - gm4<-gam(Prec~s(lat,lon,alt,k=200),data=datos)
  - gm5<-gam(Prec~s(lat,lon,alt,k=200,bs='cr'),data=datos)
  - gm6<-gam(Prec~s(lat,k=50)+s(lon,k=50)+s(alt,k=50),data=datos, family=gamma(link=log))
  - gm7<gam(Prec~s(lat,k=50,bs='cr')+s(lon,k=50,bs='cr')+s(alt,k=50,bs='cr'),data=datos,family=gamma(link=log))
  - gm8<-gam(Prec~s(lat,lon,alt,k=100),data=datos,family=gamma(link=log))
  - gm9<gam(Prec~s(lat,lon,alt,k=100,bs='cr'),data=datos,family=gamma(link=log))
  - gm10<-gam(Prec~s(lat,lon,alt,k=200),data=datos,family=gamma(link=log))
  - gm11<gam(Prec~s(lat,lon,alt,k=200,bs='cr'),data=datos,family=gamma(link=log))

11.4 Código

En las siguientes páginas se lista el código construido en R; la máquina computacional.

Codigo que contruye 12 modelos y elige aquel que posea el menor MSE obtenido al construir el modelo. Se guarda la grafica summary y se calculan dos promedios: de una malla cuya frontera es cuadrada,
y otra cuya frontera es un poligono convexo.Tambien se genera la grafica que muestra los resultados del modelo seleccionado. El menor MSE se determina al
seleccionar los puntos de la malla de las predicciones coindidente con los puntos de los datos observados y sumarlos.

```
#libreria que permite construir GAMs
library(mgcv)

#libreria que permite el manejo de graficos (geom_*)
library(ggplot2)

#libreria que permite verificar las predicciones de los GAMs
library(verification)
```

```
#libreria que permite determinar de entre un conjunto de puntos, aquellos que forman un poligono convexo que contiene a todos los puntos
library(rgeos)

#libreria que contiene la funcion pip, la cual permite determinar si un conjunto de puntos dados se ubican dentro de un cierto poligono
library(splancs)

#codigo que genera una matriz (C) de NX2, a partir de dos vectores (A,B) de nx1 y mx1 (N=nxm). Cada punto de la matriz representa un punto de una malla
#regular. Si A=(a1,a2,...an) y B=(b1,b2,...bm) entonces c=(a1,b1;a1,b2,...a1,bm;...an,b1,an,b2;...an,bm). C=cmesh(A,B)
source('/home/obed/Documents/codigos/R/GAM/malla.R')

#codigo que permite insertar una 'barra de colores' a un grÃĄfico tipo 'imagen'
source('/home/obed/Documents/codigos/R/GAM/exaples/image.scale.R')

#se lee base de datos ETOPO1
topolat<-
read.table('/home/obed/Documents/codigos/R/GAM/modelos/mapas/lat.txt',header=T,sep='')
topolon<-
read.table('/home/obed/Documents/codigos/R/GAM/modelos/mapas/lon.txt',header=T,sep='')
topoalt<-read.table('/home/obed/Documents/codigos/R/GAM/modelos/mapas/matriz.txt',sep='\t')

#se lee el archivo que contiene los puntos del contorno de Mexico
mexico<-
read.csv('/home/obed/Documents/codigos/R/GAM/exaples/mapas/mexico.csv',sep=',',header=T
)
tmi<-which(mexico$lat!=0)
mexico<-mexico[tmi,]

#%%%%%datos observados CONTRUCCION%%%%%
#Base de datos que contiene los valores de precipitacion observada por estaciones climatologicas del estado de morelos Chiapas-2013
pdatosdf<-
read.table('/home/obed/Documents/codigos/R/GAM/modelos/datosin/chiapasrain001.txt',header
=T,sep='')
datosdf<-pdatosdf[pdatosdf[,4]!=max(pdatosdf[,4]),]
ndatosdf<-names(pdatosdf)
#
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%

#Ubicacion de la carpeta donde se almacenaran las figuras y archivos de texto que representan los resultados
ad<-'/home/obed/Documents/codigos/R/GAM/modelos/ex6results/'

#comandos que permiten construir los GAMs
```

```
comandos<-
read.csv('/home/obed/Documents/codigos/R/GAM/modelos/datosin/comandosgamex6.csv',hea
der=T,sep='\t')

#Se construyen los vectores que contienen las variables 'lon', 'lat' y 'alt' para los cuales se
haran predicciones con los GAMs.
#Tales predicciones se realizaran en puntos coincidentes con la malla de ETOPO1.
#La region sobre la que se realizaran las predicciones es de forma rectangular.
#Se debera proporcionar los valores extremos de dicho rectangulo

lo.inf<-min(datosdf$lon,na.rm=T)
lo.sup<-max(datosdf$lon,na.rm=T)
la.inf<-min(datosdf$lat,na.rm=T)
la.sup<-max(datosdf$lat,na.rm=T)
#proporcionar longitud (limite inferior, limite superior)
lim.mor.lon<-c(lo.inf,lo.sup)
#proporcionar latitud (limite inferior, limite superior)
lim.mor.lat<-c(la.inf,la.sup)
#en la variable i.matriz.lxx se almacena el indice de los vectores 'lon' y 'lat' que contienen los
valores mas cercanos a los deseados
#tome en cuenta que la resolucion de la malla ETOPO1 es de 1 segundo de grado
i.matriz.lon<-array(NaN,2)
i.matriz.lat<-i.matriz.lon
#indice del vector que contiene el limite inferior (longitud)
i.matriz.lon[1]<-which.min(abs(topolon[,1]-lim.mor.lon[1]))
#indice del vector que contiene el limite superior (longitud)
i.matriz.lon[2]<-which.min(abs(topolon[,1]-lim.mor.lon[2]))
#indice del vector que contiene el limite inferior (latitud)
i.matriz.lat[1]<-which.min(abs(topolat[,1]-lim.mor.lat[1]))
#indice del vector que contiene el limite superior (latitud)
i.matriz.lat[2]<-which.min(abs(topolat[,1]-lim.mor.lat[2]))
#vectores
tlon<-topolon[i.matriz.lon[1]:i.matriz.lon[2],1]
tlat<-topolat[i.matriz.lat[1]:i.matriz.lat[2],1]
talt<-topoalt[i.matriz.lon[1]:i.matriz.lon[2],i.matriz.lat[1]:i.matriz.lat[2]]
#Se construye la matriz que contiene los puntos de la malla (lon,lat) sobre la cual se haran las
predicciones del GAM
tmeshxy<-cmesh(tlon,tlat)

# en zz se almacenaran temporalmente las predicciones de los GAMs
zz<-array(NaN,length(tlon)*length(tlat))
# Se construye el data frame en el que se almacenaran las variables independientes y las
predicciones.
# Las diferentes columnas que forman las variables independientes deben coincidir, en orden,
con las de los datos observados (construccion y verificacion)
# El data frame esta organizado como sigue: 3 columnas para las variables independientes
(lon, lat y alt: 1,2 y 3);
# las predicciones de cada modelo se almacenan en una columna 'gm0'-> 4, 'gm1'->5, etc.
```

```
prediction<-
data.frame(lon=tmeshxy[,1],lat=tmeshxy[,2],alt=as.vector(t(talt)),gm0=zz,gm1=zz,gm2=zz,gm3=
zz,gm4=zz,gm5=zz,gm6=zz,gm7=zz,gm8=zz,gm9=zz,gm10=zz,gm11=zz)
# data frame que contiene el error de la estimacion. Su estructura es la misma que el de las
predicciones
prediction.err<-
data.frame(lon=tmeshxy[,1],lat=tmeshxy[,2],alt=as.vector(t(talt)),gm0=zz,gm1=zz,gm2=zz,gm3=
zz,gm4=zz,gm5=zz,gm6=zz,gm7=zz,gm8=zz,gm9=zz,gm10=zz,gm11=zz)


# Se construyen todos los modelos propuestos y se crean los archivos que contienen sus
predicciones
for (gamname in comandos[,1])
{
  #Se construye el comando para crear el GAM usando el archivo "comandosex6.csv"
  togam<-paste(gamname,'<-',comandos[which(comandos$GAM==gamname),2],sep='')
  eval(parse(text=togam))
  #se guarda como un archivo de texto el resumen del gam (GCV, EDF, etc)
#    sink(paste(ad,'summary',gamname,'.txt',sep=''))
#    td<-paste('print(summary(',gamname,'))',sep='')
#    eval(parse(text=td))
#    sink()
#   td<-paste('qq.gam(',gamname,',mar=c(4,5,4,2)+0.1,pch=19,cex.axis=1.5,cex.lab=1.5)',sep='')
#   eval(parse(text=td))
#                                                                                    td<-
paste('plot(fitted(',gamname,'),residuals(',gamname,'),mar=c(4,5,4,2)+0.1,pch=19,cex.axis=1.5,c
ex.lab=1.5)',sep='')
#   eval(parse(text=td))
#   dev.new()
  #Se guarda imagen que contiene los grÃĄficos 'deviance residual vs theoretical quantiles',
'resid vs linearpred', 'histogram of resid' y 'response vs fitted val'
#   jpeg(file=paste(ad,'check',gamname,'.jpeg',sep=''),quality=100,width=800,height=800)
#                                                                                    td<-
paste('gam.check(',gamname,',mar=c(4,5,4,2)+0.1,pch=19,cex.axis=1.5,cex.lab=1.5)',sep='')
#   eval(parse(text=td));dev.off()
  #predicciones del GAM
  td<-paste("zz<-predict(",gamname,",prediction[,1:3],type='response',se.fit=T)",sep='')
  eval(parse(text=td))
  prediction[,gamname]<-zz$fit
  #se buscan y cambian, a cero, valores negativos de las predicciones.
  prediction[which(prediction[,gamname]<0),gamname]<-0
  prediction.err[,gamname]<-zz$se.fit
}
tfprediction<-matrix(NaN,ncol=ncol(prediction),nrow=nrow(prediction)+1)
tfprediction[1,1:3]<-ndatosdf[1:3]
tfprediction[1,4:ncol(prediction)]<-as.character(comandos[,1])
tfprediction[2:(nrow(prediction)+1),]<-as.matrix(prediction,ncol=ncol(prediction))
#se guardan las predicciones en el archivo 'predictiongmx.txt'
write(t(tfprediction),paste(ad,'predictionall.txt',sep=''),ncol=ncol(prediction),sep='\t')
```

```
# Finaliza el codigo que construye todos los GAMs

# Se busca el GAM cuyas predicciones exhiban el menor error, considerando unicamente los puntos coindicentes con las observaciones.
# Los puntos de la malla cuyos errores seran considerados son aquellos mas cercanos a los puntos que corresponden a los datos observados (aquellos que se emplearon para construir el GAM)
# %%%%Primero: se extraen los puntos de las predicciones que se consideraran
# Para ello se multiplican los valores lon y lat tanto de la matriz de pÅ•ediccion como de las observaciones y se 'busca el valor mas cercano'(dicha busqueda es 'punto por punto').
prod.predic<-prediction$lon*prediction$lat
prod.obsvr<-datosdf$lon*datosdf$lat
indices<-array(NaN,nrow(datosdf))
# Se buscan los indices del vector prod.predic que contiene las coordenadas mas cercanas a cada punto geografico de los datos observados (verificacion)
for (ind in 1:nrow(datosdf)){indices[ind]<-which.min(abs(prod.predic-prod.obsvr[ind]))}
# matriz que contiene los valores observados y los pronosticados que se compararan.
# La informacion esta organizada en columnas: 1 datos observados; 2: gm0; 3:gm1; etc
toverify<-matrix(NaN,ncol=(ncol(prediction)-2),nrow=nrow(datosdf))
toverify[,1]<-datosdf[,4]
cont<-2
for (gamname in comandos[,1]){toverify[,cont]<-prediction[indices,gamname];cont<-cont+1}
# %%%%Segundo: se calculan los siguientes parametros MAE,ME,MSE y R (coeficiente de correlacin de Kendall)
# se construye una matriz que posteriormente se guardara por columnas: modelo 1; MAE 2; MSE 3; R 4; MAE*ME*MSE 5. Agregar conforme sea necesarion
table.verify<-matrix(NaN,ncol=6,nrow=(nrow(comandos)+1))
for (h in 2:ncol(toverify))
  {
    vv<-verify(toverify[,1],toverify[,h],baseline='NULL',frcst.type='cont',obs.type='cont')
    table.verify[h,2]<-vv$MAE
    table.verify[h,3]<-vv$ME
    table.verify[h,4]<-vv$MSE
    table.verify[h,5]<-cor(toverify[,1],toverify[,h],method='kendall')
    table.verify[h,6]<-abs(vv$MAE*vv$ME*vv$MSE)
  }
table.verify[2:ncol(toverify),1]<-as.character(comandos[,1])
table.verify[1,2]<-'MAE'
table.verify[1,3]<-'ME'
table.verify[1,4]<-'MSE'
table.verify[1,5]<-'R(Kendall)'
table.verify[1,6]<-'MAE*ME*MSE'
mse.min<-min(as.real(table.verify[2:(nrow(comandos)+1),4]))

# %%%%Se construye el GAM con mejor presision (MSE menor)%%%%
#bgamname<-table.verify[which(as.real(table.verify[2:(nrow(comandos)+1),4])==mse.min)+1,1]
bgamname<-'gm0'
togam<-paste('bgam<-',comandos[which(comandos$GAM==bgamname),2],sep='')
eval(parse(text=togam))
```

```
#Se guarda imagen que contiene los grÃĄficos 'deviance residual vs theoretical quantiles', 'resid vs linearpred', 'histogram of resid' y 'response vs fitted val'
jpeg(file=paste(ad,'check',bgamname,'.jpeg',sep=''),quality=100,width=800,height=800)
gam.check(bgam,mar=c(4,5,4,2)+0.1,pch=19,cex.axis=1.5,cex.lab=1.5);dev.off()

#se guarda como un archivo de texto el resumen del gam (GCV, EDF, etc)
sink(paste(ad,'summar',bgamname,'.txt',sep=''))
print(summary(bgam))
sink()
# se calcula la media del 'mejor' modelo
#primero: contorno cuadrado
togam<-paste("out.sqv0<-mean(prediction[,'",bgamname,"'])",sep='')
eval(parse(text=togam))
#segundo: contorno construido por las estaciones
#a) se determinan los puntos que forman un poligono convexo que contiene todas las estacones (shell)
chull.bgam<-chull(datosdf[,1],datosdf[,2])
#b) a partir de los datos contenidos en el data frame 'prediction', se extraen todos aquellos puntos contenidos en shell (se usa el algoritmo desarrollado por Checo)
#b).1 primero se contruye el archivo necesario (matriz de dos columnas y N filas. N=numero de puntos que forman el poligono de interes+numero de puntos que se desea verificar+ 1 -renglon vacio-)
# nchpoints<-length(chull.bgam);ndpoints<-nrow(datosdf)
# topip<-matrix(NA,ncol=2,nrow=(nchpoints+ndpoints+1))
# topip[1:nchpoints,1]<-datosdf$lon[chull.bgam]
# topip[1:nchpoints,2]<-datosdf$lat[chull.bgam]
# topip[(nchpoints+2):nrow(topip),1]<-datosdf$lon
# topip[(nchpoints+2):nrow(topip),2]<-datosdf$lat
#
write(t(topip),'/home/obed/Documents/codigos/R/GAM/modelos/datosin/tpip.txt',ncol=2,sep='\t')
# setwd('/home/obed/Documents/codigos/R/GAM/modelos/datosin')
# system("python pip0.py <tpip.txt> pinside.txt")

#b).1.1Se usa la libreria SPLANCS y la funcion pip
poly.get<-data.frame(x=datosdf$lon[chull.bgam],y=datosdf$lat[chull.bgam])
points.get<-data.frame(x=prediction$lon,y=prediction$lat)
pip.get<-pip(points.get,poly.get,bound=T,out=F,quiet=T)
pip.row<-as.real(rownames(pip.get))
togam<-paste("out.chullv0<-mean(prediction[pip.row,'",bgamname,"'])",sep='')
eval(parse(text=togam))
file.out<-matrix(NaN,ncol=3,nrow=2)
file.out[1,1]<-''
file.out[1,2]<-'sq'
file.out[1,3]<-'chull'
file.out[2,1]<-'media'
file.out[2,2]<-out.sqv0
file.out[2,3]<-out.chullv0
write(t(file.out),paste(ad,'out.txt',sep=''),ncol=3,sep='\t')
#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
# grafica del modelo en contorno cuadrado
#marcadores y etiquetas de la leyenda
marcadores<-c(0,5,10,15,20,25,30,35,40,45,50)
#limite inferior y superor de la escala considerada
limites.plot<-c(0,max(prediction[,bgamname]))

tgplot<-ggplot()
#se agregan grafico equivalente a 'image' (cuadros con color que representa una variable) con las predicciones
top<-paste('tgplot<-
tgplot+geom_tile(data=prediction,aes(x=lon,y=lat,fill=',bgamname,'))+scale_fill_gradientn(colours=c("red","green","blue"),limits=limites.plot,breaks=marcadores,legend=F)',sep='')
eval(parse(text=top))
#tamÃąo del circulo con que se identifica la estaciÃ‚n
est.size<-3
#se agregan puntos que representan las estaciones climatologicas; solo es un circulo vacio en color negro
tgplot<-tgplot+geom_point(data=datosdf,aes(x=lon,y=lat),size=est.size+2,shape=21)
#punto que representa la cantidad de precipitacion registrada por la estacion correspondiente; el color indica la escala
tgplot<-
tgplot+geom_point(data=datosdf,aes(x=lon,y=lat,colour=preac09),size=est.size)+scale_color_gradientn(colours=c("red","green","blue"),limits=limites.plot,breaks=marcadores,expression(Precipitacion))+xlab('LON')+ylab('LAT')+theme(text=element_text(size=20))
#se agregan lineas que repÅ•esentan a la Republica Nexicana
# tgplot<-tgplot+geom_path(data=mexico,aes(x=lon,y=lat))
#se guarda como imagen las predicciones del GAM
jpeg(file=paste(ad,'predictionsq',bgamname,'.jpeg',sep=''),quality=100,width=800,height=800)
eval(parse(text='tgplot'));dev.off()
#

tgplot<-ggplot()
#se agregan grafico equivalente a 'image' (cuadros con color que representa una variable) con las predicciones
top<-paste('tgplot<-
tgplot+geom_tile(data=prediction[pip.row,],aes(x=lon,y=lat,fill=',bgamname,'))+scale_fill_gradientn(colours=c("red","green","blue"),limits=limites.plot,breaks=marcadores,legend=F)',sep='')
eval(parse(text=top))
# tgplot<-tgplot+geom_text(size=10)
#se agregan puntos que representan las estaciones climatologicas; la intensidad y tamaÃąo de los puntos tienen siginficado
tgplot<-tgplot+geom_point(data=datosdf,aes(x=lon,y=lat),size=est.size+2,shape=21)
tgplot<-
tgplot+geom_point(data=datosdf,aes(x=lon,y=lat,colour=preac09),size=est.size)+scale_color_gradientn(colours=c("red","green","blue"),limits=limites.plot,breaks=marcadores,expression(Precipitacion))+xlab('LON')+ylab('LAT')+theme(text=element_text(size=20))
#se agregan lineas que repÅ•esentan a la Republica Nexicana
# tgplot<-tgplot+geom_path(data=mexico,aes(x=lon,y=lat))
```

```
#se guarda como imagen las predicciones del GAM
jpeg(file=paste(ad,'predictionch',bgamname,'.jpeg',sep=''),quality=100,width=800,height=800)
eval(parse(text='tgplot'));dev.off()
```

12 ANEXO 4: Muestra de salidas del sistema

A continuación se muestran algunas de las salidas gráficas y tabulares del sistema.

******* ***** RESUMEN ESTADISTICO DE LA REGION SELECCIONADA ***** *******

FECHA CONSULTA:

meses= (Jul, Ago, Sep, Oct, Nov)

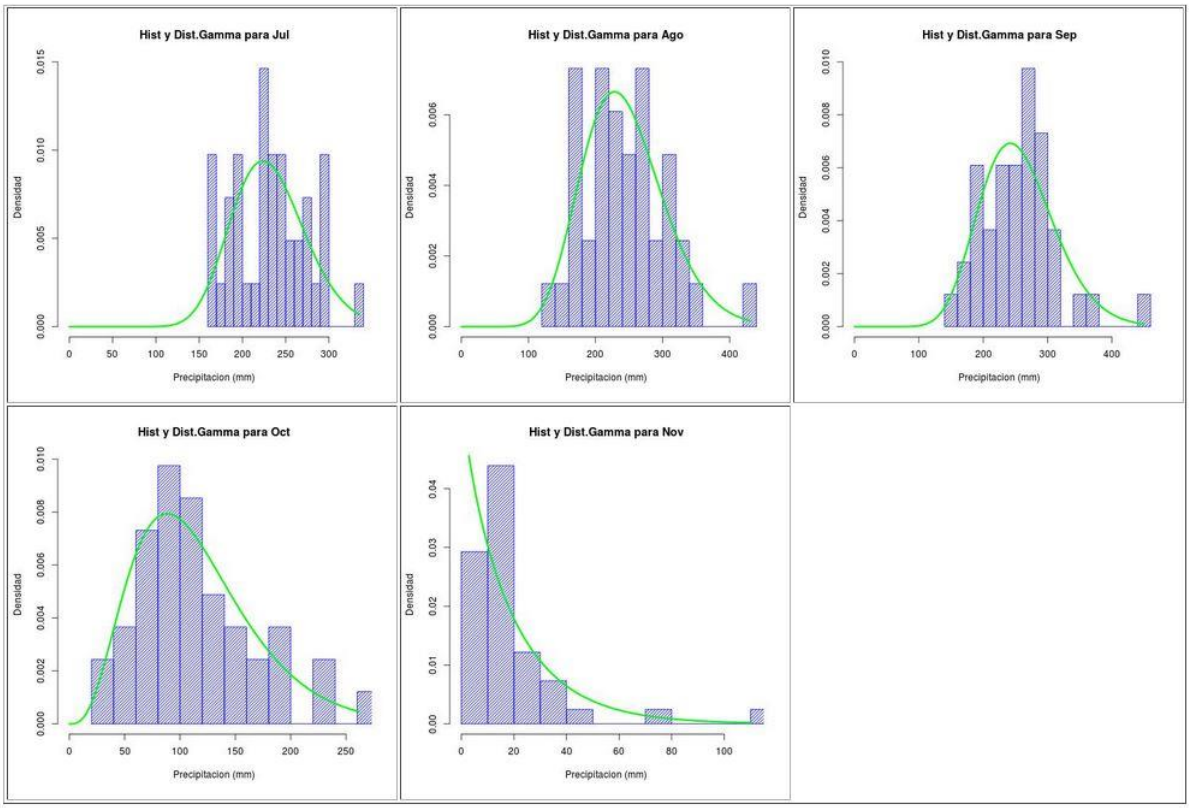
período años= 1970 ... 2010

Tabla de Precipitaciones Mensuales Acumuladas

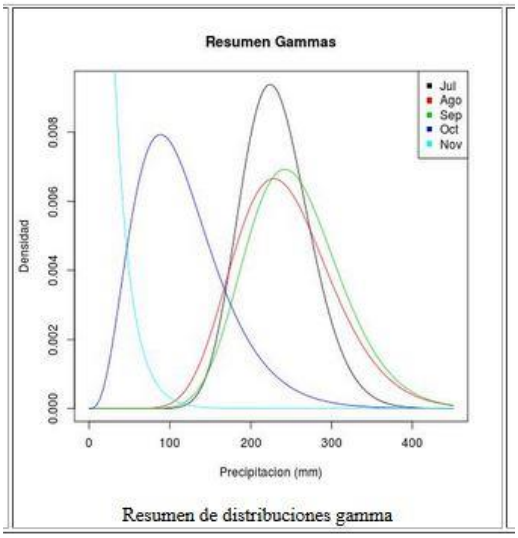
	Jul	Ago	Sep	Oct	Nov
1970	227	240	281	81.2	19.10
1971	196	202	272	104.0	13.15
1972	260	158	181	63.5	30.99
1973	211	262	292	188.9	16.75
1974	201	174	289	48.1	23.86
1975	227	245	243	95.1	10.68
1976	166	196	205	261.9	42.08
1977	168	226	189	78.3	23.64
1978	299	223	275	145.4	16.67
1979	246	292	273	54.8	3.47
1980	248	309	244	76.7	23.33
1981	269	349	243	190.0	15.98
1982	181	134	184	89.0	7.49
1983	292	209	239	108.4	21.62
1984	292	233	451	43.8	9.25

En el ejemplo que se muestra se ha seleccionado como área de operación el estado de Guerrero, y se han elegido para el análisis un período entre los años 1970 y 2010, y los meses lluviosos, esto es, entre julio y noviembre. Se muestra sólo un fragmento de la tabla real que arroja el sistema. Cada entrada de la tabla es la precipitación promedio en el año y mes, para el área seleccionada. Esta precipitación promedio se encuentra por los métodos teóricos del tipo descrito anteriormente. La configuración de esta tabla constituye el principal mecanismo y punto de partida del sistema.
Con los datos de la tabla anterior se pueden calcular las frecuencias y con ello ajustar a una distribución de probabilidades, en este caso, la distribución gamma.

***** ***** frecuencias y ajuste a distribuciones gamma ***** *****



En la figura se muestran las frecuencias de precipitación para cada uno de los meses analizados y su correspondiente ajuste a la distribución de probabilidades gamma. En seguida se muestra un resumen de todas las distribuciones.



Finalmente, se muestra la tabla de curvamasas de precipitación, esto es, en el renglón correspondiente a cada año, tomando como base la tabla de precipitaciones, se acumula mes a mes la precipitación.

Tabla de Curva-masas

	Jul	Ago	Sep	Oct	Nov
1970	227	467	748	829	848
1971	196	397	670	774	787
1972	260	418	599	662	693
1973	211	473	765	954	971
1974	201	375	664	712	736
1975	227	472	715	810	821
1976	166	362	567	829	871
1977	168	394	583	661	685
1978	299	521	796	942	958
1979	246	538	811	866	869
1980	248	556	800	877	900
1981	269	617	860	1050	1066
1982	181	315	499	588	596
1983	292	502	740	848	870
1984	292	525	975	1019	1029
1985	242	518	691	778	789
1986	177	356	514	579	598
1987	272	449	640	666	676
1988	242	581	823	890	894
1989	194	409	767	868	884
1990	192	376	552	684	703
1991	168	337	565	684	703
1992	232	492	777	896	922
1993	270	490	766	865	879
1994	160	379	560	691	707
1995	223	562	829	918	934
1996	238	527	739	916	928
1997	183	362	610	845	883
1998	297	603	979	1207	1242

13 Bibliografía y fuentes de información adicional

- Downie N.M. Métodos estadísticos aplicados. Harla, México, primera edition, 1973.
- Wilks Daniels S. Statistical Methods in the Atmosferic Sciences. Elsevier, Amsterdam, segunda edición, 2006.
- Grace Wahba. Spline Models for Observational Data. Society for industrial and applied mathematics, Pennsylvania, 1990.
- Simon N. Wood. Generalized Additive Models: an introduction with R. Champman & Hall, Londres, 2006.
- Elliotte Rusty Harold. XML Bible. Hungry Minds, Inc. Second Edition 2001.
- http://www.php.net/manual/es/
- http://www.w3schools.com/js/
- https://developers.google.com/maps/documentation/javascript/?hl=es
- http://www.json.org/