

Visor de información de telemetría

CP1412.1

Informe final

COORDINACIÓN DE COMUNICACIÓN, PARTICIPACIÓN E
INFORMACIÓN

SUBCOORDINACIÓN DE DIFUSIÓN Y DIVULGACIÓN

Fernando Flores Prior

México, 2015

F.CO.2.03.00

Índice

| | |
|--|-----|
| 1 Resumen ejecutivo | 3 |
| 2 Objetivos | 3 |
| 3 Antecedentes | 3 |
| TELEMETRIA | 3 |
| 4 Metodología | 3 |
| ANSI SQL | 4 |
| MYSQL 5.6 | 5 |
| PHP | 6 |
| SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN DE DATOS DE TELEMETRÍA | 7 |
| MYSQL WORKBENCH | 7 |
| APLICACIÓN MÓVIL PARA LA VISUALIZACIÓN DE INFORMACIÓN DE TELEMETRÍA | 8 |
| JAVA..... | 8 |
| JAVASCRIPT | 9 |
| ANDROID..... | 9 |
| 5 Resultados | 12 |
| ESTRUCTURACION DEL SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN DE DATOS DE TELEMETRÍA..... | 12 |
| MODULO GOOGLE MAPS..... | 15 |
| 6 Conclusiones | 21 |
| 7 Bibliografía | 21 |
| 8 Anexos | 22 |
| Anexo 1. Código fuente del sistema informático Web para la gestión de datos de telemetría..... | 22 |
| Anexo 2. Código fuente de la Aplicación Móvil para la visualización de información de telemetría. | 112 |

1 Resumen ejecutivo

Contenido

2 Objetivos

- Llevar a cabo el desarrollo de una herramienta para el manejo y visualización de datos de telemetría contenidos en una base de datos origen. La herramienta a desarrollar llevara a cabo la extracción de información que podrá ser visualizada desde un dispositivo móvil.
- Desarrollo de aplicación móvil en la plataforma Android para la visualización de los datos de las distintas estaciones de telemetría recabados por el sistema web contenedor mediante el análisis de los requerimientos.

3 Antecedentes

TELEMETRIA

La telemetría también conocida como telemedición y control es una rama tecnológica que permite la medición a distancia de magnitudes físicas, de manera manual o automática. Los sistemas de telemetría permiten conocer los estados que guardan equipos, procesos y sistemas, así como controlar de manera remota su funcionamiento, identificar fallas y corregir estados de error de los mismos, permitiendo así, el envío de la información recabada hacia un sistema de notificación y/o despliegue para su uso y aprovechamiento.

El proyecto actual se engloba dentro del área de desarrollo de aplicaciones móviles. De las diferentes opciones en la actualidad para desarrollos tecnológicos se ha elegido la plataforma Android debido a las opciones presentadas para la generación de aplicaciones bajo este entorno.

La idea inicial se basa en desarrollar una aplicación para dispositivos Android que permita a los usuarios mantener conocimiento de la información recabada por las estaciones hidrométricas.

La funcionalidad de la aplicación permite la extracción de datos a través de un servicio web el cual establece vínculo a la base de datos en el formato seleccionado.

4 Metodología

La aplicación desarrollada en este proyecto consiste en un sistema visor de telemetría utilizado para dar seguimiento a los datos proporcionados por los sensores de cada una de las estaciones, para lo

que se ha desarrollado un sistema informático Web para la gestión de datos de telemetría y una Aplicación Móvil para la visualización de información.

Para estas aplicaciones se han utilizado en común los siguientes lenguajes de programación y estándares para el desarrollo:

ANSI SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) 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 ellas.

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones.

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros —y no a registros individuales— permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros. SQL también tiene las siguientes características:

- Lenguaje de definición de datos: El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.
- Lenguaje interactivo de manipulación de datos: El LMD de SQL incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas.
- Integridad: El LDD de SQL incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.
- Definición de vistas: El LDD incluye comandos para definir las vistas.
- Control de transacciones: SQL tiene comandos para especificar el comienzo y el final de una transacción.
- SQL incorporado y dinámico: Esto quiere decir que se pueden incorporar instrucciones de SQL en lenguajes de programación como: C++, C, Java, PHP, Cobol, Pascal y Fortran.
- Autorización: El LDD incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas.

Tipos de Datos

Algunos de los tipos de datos básicos de SQL son:

- Date: una fecha de calendario que contiene el año (de cuatro cifras), el mes y el día.
- Time: La hora del día en horas minutos segundos (el valor predeterminado es 0).

- Timestamp: la combinación de Date y Time.

Optimización

Como ya se dijo antes, y suele ser común en los lenguajes de acceso a bases de datos de alto nivel, el SQL es un lenguaje declarativo. O sea, que especifica qué es lo que se quiere y no cómo conseguirlo, por lo que una sentencia no establece explícitamente un orden de ejecución.

El orden de ejecución interno de una sentencia puede afectar seriamente a la eficiencia del SGBD, por lo que se hace necesario que éste lleve a cabo una optimización antes de su ejecución. Muchas veces, el uso de índices acelera una instrucción de consulta, pero ralentiza la actualización de los datos. Dependiendo del uso de la aplicación, se priorizará el acceso indexado o una rápida actualización de la información. La optimización difiere sensiblemente en cada motor de base de datos y depende de muchos factores.

MYSQL 5.6

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL cuenta con elementos esenciales propios de las bases de datos relacionales tales como la integridad referencial y transacciones.

Entre las muchas características propias de MySQL tenemos:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones...
- Transacciones y claves foráneas.
- Conectividad segura.

- Replicación.
- Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

PHP

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas. PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc. El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección tipo `http://www.aprenderaprogramar.com/index.php` en un navegador web como Internet Explorer, Firefox o Chrome, ¿qué ocurre? Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

El esquema es: Petición de página web al servidor --> El servidor recibe la petición, reúne la información necesaria consultando a bases de datos o a otras páginas webs, otros servidores, etc. -> El servidor responde enviando una página web "normal" (estática) pero cuya creación ha sido dinámica (realizando procesos de modo que la página web devuelta no siempre es igual).

SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN DE DATOS DE TELEMETRÍA

Los módulos del sistema se han desarrollado mediante la siguiente tecnología, adicionales a las mencionadas en el apartado anterior.

MYSQL WORKBENCH

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.

MySQL Workbench permite modelar diagramas de Entidad-Relación para bases de datos MySQL, con esta herramienta se puede elaborar una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real. Se puede realizar una ingeniería directa e ingeniería inversa para exportare e importar el esquema de una base de datos ya existente el cual haya sido guardado o hecho copia de seguridad con MySQL Administrador.

MySQL Workbench puede generar también el guión necesario para crear la base de datos que se ha dibujado en el esquema; es compatible con los modelos de base de datos de DBDesigner 4 y soporta las novedades incorporadas en MySQL 5.x

APLICACIÓN MÓVIL PARA LA VISUALIZACIÓN DE INFORMACIÓN DE TELEMETRÍA

El análisis y diseño de la aplicación se ha realizado acorde a los siguientes lenguajes de programación y estándares para el desarrollo.

JAVA

Java es un lenguaje de programación orientado a objetos que se popularizó a partir del lanzamiento de su primera versión comercial de amplia difusión, la JDK 1.0 en 1996. Actualmente es uno de los lenguajes más usados para la programación en todo el mundo.

Los antecedentes de Java habría que buscarlos en los lenguajes de programación C y C++. El lenguaje C fue desarrollado en la década de los 70 y constituye un lenguaje muy robusto usado como núcleo del sistema operativo Unix. C no admite la orientación a objetos y está considerado un lenguaje “poco amigable” desde el punto de vista de que su sintaxis, elementos de programación que emplea (manejo directo de memoria) y otras cuestiones hacen que sea un lenguaje difícil de aprender. C++ fue una evolución de C desarrollada en los años 80. Introdujo el diseño orientado a objetos, pero manteniendo la compatibilidad con C. Esto suponía que C++ permitiera tanto el uso de la programación estructurada “tradicional” como la programación orientada a objetos. Además C++ mantuvo ciertas características de C como el manejo directo de la memoria, el uso de variables globales, sentencia goto, etc. que hicieron que la evolución fuera “parcial”.

Como paso final en esta evolución tenemos Java, un lenguaje que evoluciona a partir de C y C++, pero que elimina diversos aspectos de estos lenguajes y se constituye en un lenguaje definitivamente orientado a objetos. El romper con distintos aspectos de C++ cuyo manejo inadecuado por parte de muchos programadores daba lugar a problemas en las aplicaciones ha sido un factor decisivo para convertir a Java en un lenguaje popular y de amplio uso.

Java es un lenguaje útil para casi todo tipo de problemas. Podemos citar como funcionalidades de Java varias:

1. Aplicaciones “cliente”: son las que se ejecutan en un solo ordenador (por ejemplo el portátil de tu casa) sin necesidad de conectarse a otra máquina. Pueden servirte por ejemplo para realizar cálculos o gestionar datos.
2. Aplicaciones “cliente/servidor”: son programas que necesitan conectarse a otra máquina (por ejemplo un servidor de datos) para pedirle algún servicio de forma más o menos continua, como podría ser el uso de una base de datos. Pueden servir por ejemplo para el teletrabajo: trabajar desde casa pero conectados a un ordenador de una empresa.

3. Podemos hablar también de “aplicaciones web”, que son programas Java que se ejecutan en un servidor de páginas web. Estas aplicaciones reciben “solicitudes” desde un ordenador y envían al navegador (Internet Explorer, Firefox, Safari, etc.) que actúa como su cliente páginas de respuesta en HTML.

JAVASCRIPT

El JavaScript es un lenguaje de programación que surgió por la necesidad de ampliar las posibilidades del HTML. En efecto, al poco tiempo de que las páginas web aparecieran, se hizo patente que se necesitaba algo más que las limitadas prestaciones del lenguaje básico, ya que el HTML solamente provee de elementos que actúan exclusivamente sobre el texto y su estilo, pero no permite, como ejemplo sencillo, ni siquiera abrir una nueva ventana o emitir un mensaje de aviso. La temprana aparición de este lenguaje, es posiblemente la causa de que se haya convertido en un estándar soportado por todos los navegadores actuales, a diferencia de otros, que solo funcionan en los navegadores de sus firmas creadoras.

Como tantas otras aportaciones al mundo www, fue Netscape quien inició la implementación de JavaScript (aunque al principio no se llamaba así), y posteriormente, una alianza entre Netscape y Sun, creadora del lenguaje Java, permitió que JavaScript tomara la debida consistencia, definiéndose como una variante de Java, pero mucho más simple de usar. Esto no significa que JavaScript sea Java simplificado o reducido. Salvo el nombre y la sintaxis, JavaScript no tiene mucho en común con Java, pero cumple su propósito de lenguaje auxiliar del HTML en los navegadores, y sólo en ellos ya que no es posible utilizarlo fuera del entorno de las páginas. No hay que confundirlo con el JScript de Microsoft, que aunque bastante parecido, no tiene la compatibilidad del original JavaScript, ya que, como todo lo que hacen, está pensado exclusivamente para su propio navegador.

ANDROID

Android es un sistema operativo móvil basado en Linux enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es desarrollado por la Open Handset Alliance, liderada por Google.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica, un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic.

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), Existen otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C, C++ u otros lenguajes de programación.

Android se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

En la actualidad existen más de 700.000 aplicaciones para Android y se estima que 1.000.000 teléfonos móviles se activan diariamente.

Android es criticado por la fragmentación que sufren sus terminales al no ser soportados con actualizaciones por parte de los fabricantes, algo que se podría considerar obsolescencia programada. Esto supone problemas para los desarrolladores que deben lidiar con la retro compatibilidad entre diferentes versiones del sistema operativo.

Resulta evidente que aprender a desarrollar proyectos para este sistema operativo mejora las perspectivas tanto laborales como académicas de cualquier estudiante de Informática o similares.

Para desarrollar sobre Android se hará uso del paquete ADT (Android Developer Tools), en su versión Bundle, que incluye el software necesario para comenzar desarrollar para esta plataforma.

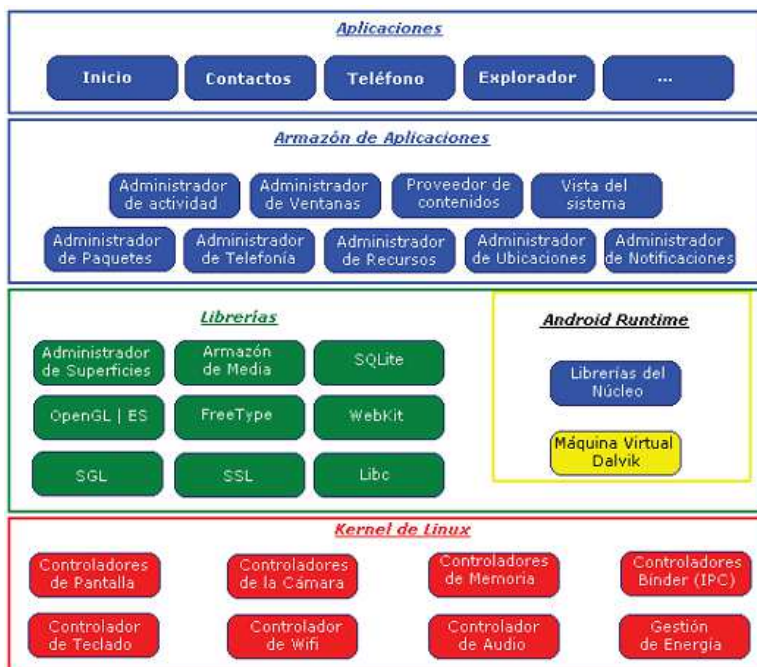


Ilustración 1 Sistema de capas de Android

Componentes de una aplicación

Todas las aplicaciones en Android pueden descomponerse en cuatro tipos de bloques o componentes principales. Cada aplicación será una combinación de uno o más de estos componentes, que deberán ser declarados de forma explícita en un fichero con formato XML denominado “AndroidManifest.xml”, junto a otros datos asociados como valores globales, clases que implementa, datos que puede manejar, permisos, etc. Este fichero es básico en cualquier aplicación en Android y permite al sistema desplegar y ejecutar correctamente la aplicación. A continuación se exponen los cuatro tipos de componentes en los que puede dividirse una aplicación para Android.

Activity

Sin duda es el componente más habitual de las aplicaciones para Android. Un componente Activity refleja una determinada actividad llevada a cabo por una aplicación, y que lleva asociada típicamente una ventana o interfaz de usuario; es importante señalar que no contempla únicamente el aspecto gráfico, sino que éste forma parte del componente Activity a través de vistas representadas por clases como View y sus derivadas. Este componente se implementa mediante la clase de mismo nombre Activity. La mayoría de las aplicaciones permiten la ejecución de varias acciones a través de la existencia de una o más pantallas. Por ejemplo, piénsese en una aplicación de mensajes de texto. En ella, la lista de contactos se muestra en una ventana. Mediante el despliegue de una segunda ventana, el usuario puede escribir el mensaje al contacto elegido, y en otra tercera puede repasar su historial de mensajes enviados o recibidos. Cada una de estas ventanas debería estar representada a través de un componente Activity, de forma que navegar de una ventana a otra implica lanzar una actividad o dormir otra. Android permite controlar por completo el ciclo de vida de los componentes Activity. Muy vinculado a este componente se encuentran los Intents, una interesante novedad introducida por Android. Un Intent consiste básicamente en la voluntad de realizar alguna acción, generalmente asociada a unos datos. Lanzando un Intent, una aplicación puede delegar el trabajo en otra, de forma que el sistema se encarga de buscar qué aplicación entre las instaladas es la que puede llevar a cabo la acción solicitada. Por ejemplo, abrir una URL en algún navegador web, o escribir un correo electrónico desde algún cliente de correo.

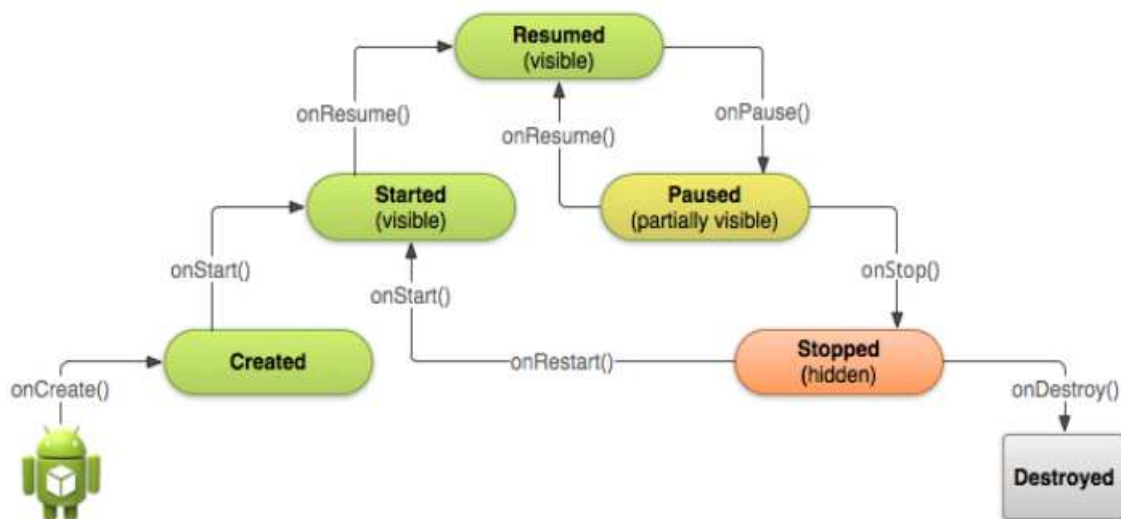


Ilustración 2 Ciclo de vida de una aplicación Android

La aplicación se ha probado mediante dispositivos Android en las versiones 3.0 hasta la versión 5.0 tanto en teléfonos móviles como en tabletas Android de 7 y 10 pulgadas.

5 Resultados

ESTRUCTURACION DEL SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN DE DATOS DE TELEMETRÍA

La información captada y procesada de cada una de las estaciones climáticas por el servicio meteorológico nacional y posteriormente disponible para su visualización es mostrada en el siguiente esquema en donde se aprecia el ciclo que esta conlleva (medición, envío, procesamiento y visualización)

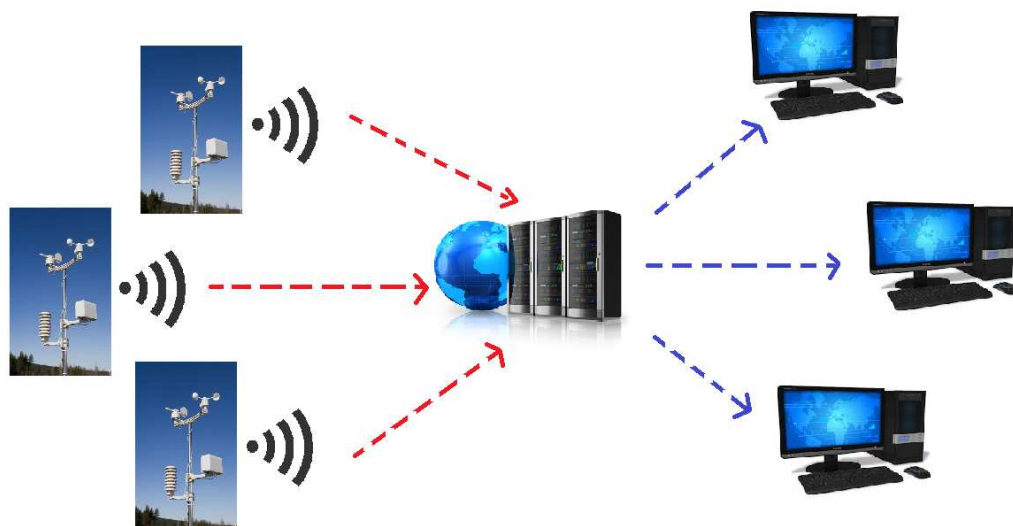


Ilustración 3 Esquema de comunicación

La instalación del servidor en la sede central con plataforma Linux, en la que está instalado el Servidor WEB, mismo que soporta el lenguaje de programación PHP para el desarrollo de la página dinámica y la base de datos en el estándar establecido previamente para el almacenamiento de la información. El Modelo de Desarrollo Web y su misión de explicitar todos los elementos o componentes que forman parte del sistema en el cual se encuentra los usuarios y sus grupos, estaciones y el tipo de estas; se describe en las siguientes secciones.

La base de datos contenedora del sistema de telemetría se encuentra en tercera forma normal la cual establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos.

Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Una dependencia transitiva es aquella en la cual existen columnas que no son llave que dependen de otras columnas que tampoco son llave.

Cuando las tablas están en la Tercera Forma Normal se previenen errores de lógica cuando se insertan o borran registros. Cada columna en una tabla está identificada de manera única por la llave primaria y no debe haber datos repetidos. Esto provee un esquema limpio y elegante, que es fácil de trabajar y expandir.

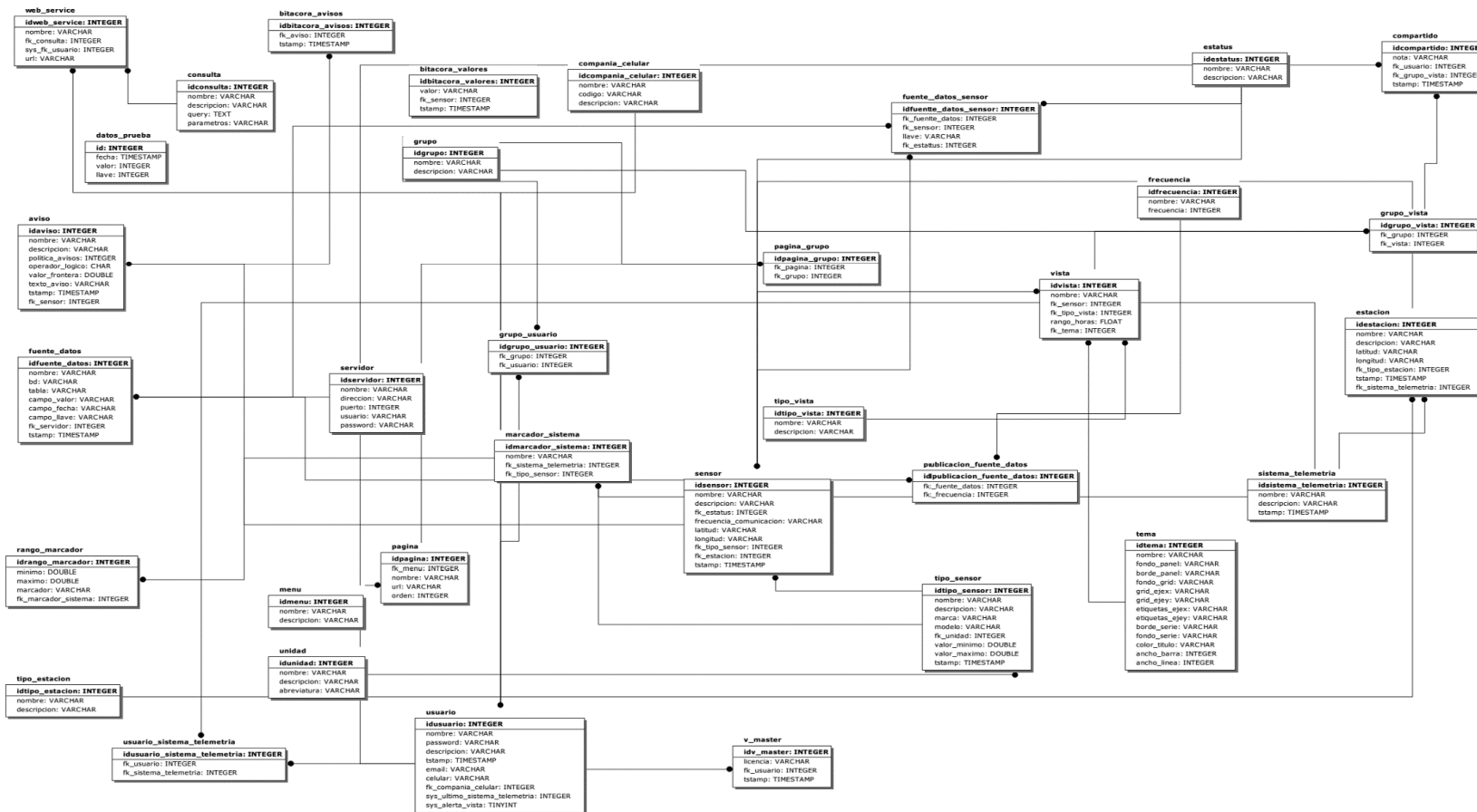


Ilustración 4 Diagrama Entidad-Relación de la base de datos del sistema visor de telemetría

El esquema para la visualización de información se encuentra dividido en niveles de permisos definidos por el usuario administrativo, mismo que integra los grupos a los que cada usuario pertenecerá.

El entorno administrativo contiene un catálogo de parámetros los cuales dependiendo del nivel o jerarquía del usuario este podrá contener en su entorno del sistema:

- Grupos
- Grupos – usuarios
- Tipo de estaciones
- Tipo de sensores
- Usuarios
- Etc.

El usuario administrativo y su acceso total al entorno le permiten definir las vistas a las que tendrá acceso el resto de los usuarios.

MODULO GOOGLE MAPS

El sistema visor de telemetría contiene un módulo de google-maps que permite visualizar mediante las coordenadas de cada estación de medición su ubicación satelital.

El acceso al sistema visor redirecciona directamente a este módulo y mostrando el geo posicionamiento en pantalla de las estaciones a las que el usuario (dependiendo de su nivel) tenga acceso.

Se tienen alrededor de 50 estaciones de medición a lo largo de la república. El administrador se encargara de crear los grupos a los cuales considere pertenecen estas.

Las siguientes pantallas muestran las estaciones a nivel nacional, estatal y las pertenecientes al Instituto Mexicano de Tecnología del Agua

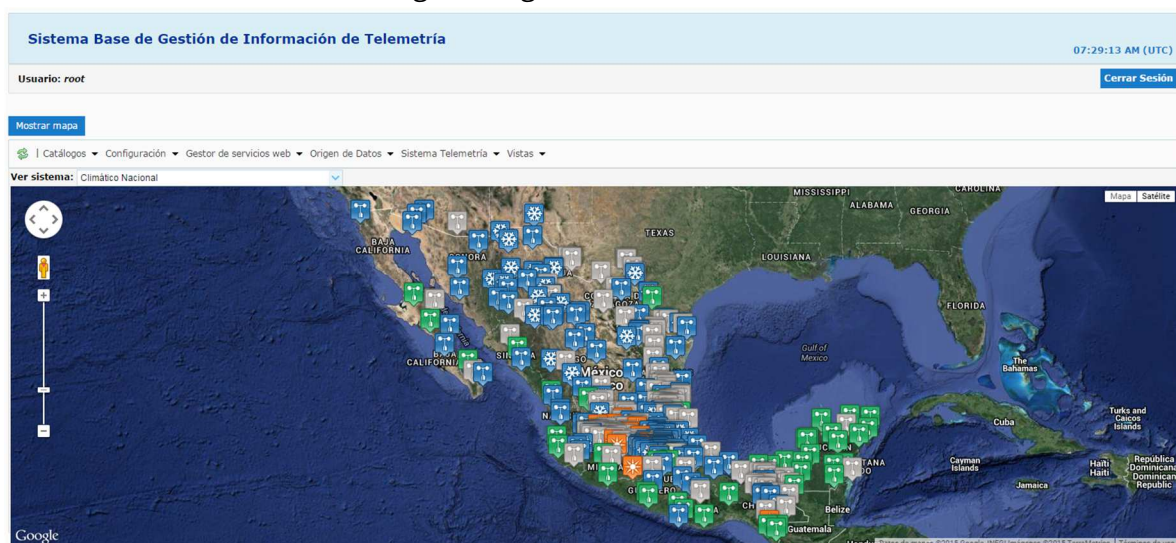


Ilustración 5 Estaciones Climáticas Nacionales

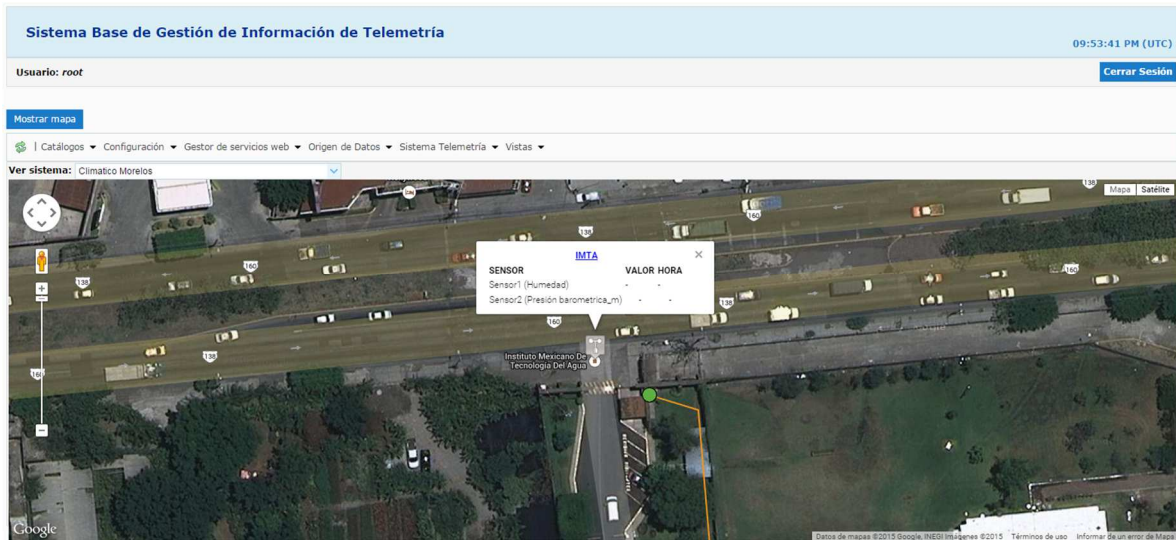


Ilustración 6 Estación climática en Morelos (IMTA)

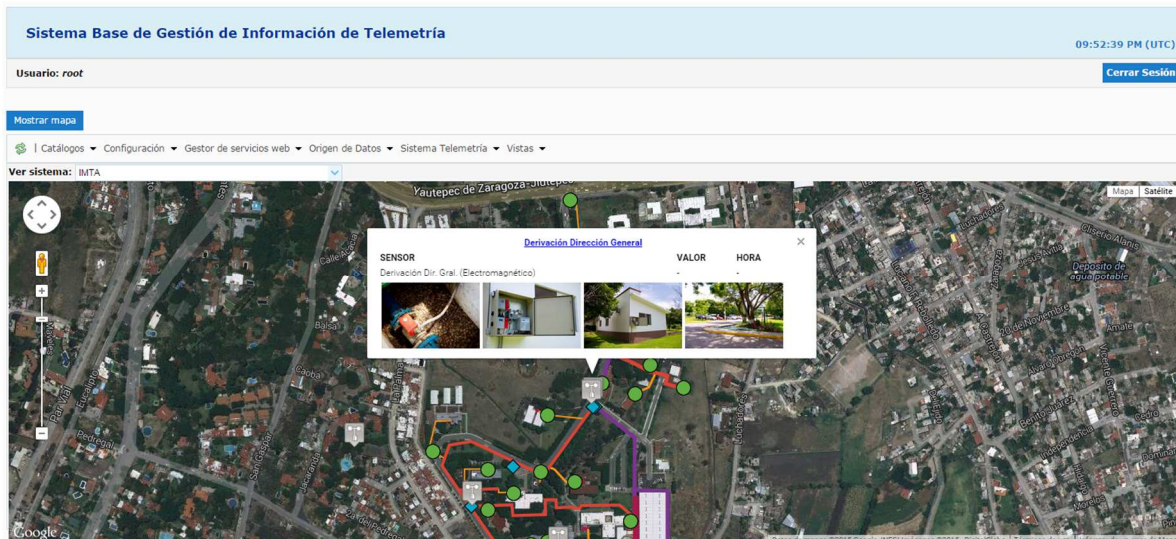


Ilustración 7 Estaciones de medición en IMTA

La funcionalidad del sistema visor de telemetría para el resto de los usuarios queda a consideración del administrador quien se encargara de llenar los parámetros que resultan en funciones de utilidad para el resto de los interesados en los datos recabados en cada una de las estaciones.

Los módulos de configuración se muestran a continuación; también se incluyen los códigos fuente generadores de estos.

CATALOGO

Sistema Base de Gestión de Información de Telemetría 07:22:08 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

[Catálogos](#) |
 [Configuración](#) |
 [Gestor de servicios web](#) |
 [Origen de Datos](#) |
 [Sistema Telemetría](#) |
 [Vistas](#)

TIPO SENSOR

[Nuevo](#) |
 [Eliminar](#) |
 [Guardar](#) |
 [Cancelar](#)

| NOMBRE | DESCRIPCION | MARCA | MODELO | UNIDAD | VALOR MINIMO | VALOR MAXIMO |
|--------------------------|----------------------------------|--------------|----------|-----------------------|--------------|--------------|
| 1 Temperatura_c | Sensor de temperatura | Emmerson | AS10 | Celsius | -40 | 60 |
| 2 Temperatura_f | Sensor de temperatura | Motorola | MZ1E | Celsius | -10 | 80 |
| 3 Velocidad | Sensor de velocidad de viento | Motorola | MZ25 | Metros por segundo | 0 | 25 |
| 4 Humedad | Sensor de humedad | Emmerson | A361 | % de humedad relativa | 0 | 100 |
| 5 Presión barometrica_m | Sensor de presión | Motorola | MZC2 | mm Hg | 780 | 1320 |
| 6 Temperatura Genérico | Sensor genérico de temperatura | Genérico | NA | Celsius | -40 | 80 |
| 7 Presión Genérico | Sensor genérico de presión | Genérico | NA | mm Hg | 780 | 1320 |
| 8 Precipitación Genérico | Sensor genérico de precipitación | Genérico | NA | Milímetros | 0 | 100 |
| 9 Genérico | Sensor genérico | Genérico | Genérico | Unidades | 0 | 100 |
| 10 Electromagnético | Electromagnético | Badger Meter | Genérico | Metros por segundo | 0 | 100 |

20 | [H](#) | [P](#) | [Página 1](#) de 1 | [P](#) | [H](#) | [O](#) Mostrando 1 a 10 de 10 registros

Ilustración 8 Catalogo - Tipo de Sensor

Sistema Base de Gestión de Información de Telemetría 07:21:34 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

[Catálogos](#) |
 [Configuración](#) |
 [Gestor de servicios web](#) |
 [Origen de Datos](#) |
 [Sistema Telemetría](#) |
 [Vistas](#)

TIPO ESTACION

[Nuevo](#) |
 [Eliminar](#) |
 [Guardar](#) |
 [Cancelar](#)

| NOMBRE | DESCRIPCION |
|--------------------|------------------------------|
| 1 Estación General | Tipo de estación por defecto |
| 2 Meteorologica A | Estación climática básica |
| 3 Meteorologica B | Estación climática avanzada |
| 4 Nacional | Estación Climática Nacional |
| 5 Electromagnético | Estación Electromagnética |

20 | [H](#) | [P](#) | [Página 1](#) de 1 | [P](#) | [H](#) | [O](#) Mostrando 1 a 5 de 5 registros

Ilustración 9 Catalogo - Tipo de Estación

Sistema Base de Gestión de Información de Telemetría 07:20:39 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

[Catálogos](#) |
 [Configuración](#) |
 [Gestor de servicios web](#) |
 [Origen de Datos](#) |
 [Sistema Telemetría](#) |
 [Vistas](#)

GRUPO

[Nuevo](#) |
 [Eliminar](#) |
 [Guardar](#) |
 [Cancelar](#)

| NOMBRE | DESCRIPCION |
|-----------|-------------------------------------|
| 1 root | Grupo de super usuarios |
| 2 Centro | Grupo con las estaciones del centro |
| 3 Sur | Grupo con las estaciones del sur |
| 4 Norte | Grupo con las estaciones del norte |
| 5 IMTA | IMTA Pozos |
| 6 Morelos | Grupo del estado de Morelos |

20 | [H](#) | [P](#) | [Página 1](#) de 1 | [P](#) | [H](#) | [O](#) Mostrando 1 a 6 de 6 registros

Ilustración 10 Catalogo - Grupo de Usuarios

Sistema Base de Gestión de Información de Telemetría 07:20:10 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

Catálogos > Configuración > Gestor de servicios web > Origen de Datos > Sistema Telemetría > Vistas >

ESTATUS

Nuevo Eliminar Guardar Cancelar

| NOMBRE | DESCRIPCION |
|------------|-------------------|
| 1 Activo | Elemento activo |
| 2 Inactivo | Elemento inactivo |

Mostrando 1 a 2 de 2 registros

Ilustración 11 Catalogo - Estatus

CONFIGURACIÓN

Sistema Base de Gestión de Información de Telemetría 07:23:47 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

Catálogos > Configuración > Gestor de servicios web > Origen de Datos > Sistema Telemetría > Vistas >

PAGINA

Nuevo Eliminar Guardar Cancelar

| MENU | NOMBRE | URL | ORDEN |
|-----------------------|------------------|---|-------|
| 1 Configuración | Menu | Grid.php?tabla=menu | 1 |
| 2 Configuración | Pagina | Grid.php?tabla=pagina | 1 |
| 3 Catálogos | Usuario | Grid.php?tabla=usuario | 1 |
| 4 Catálogos | Unidad | Grid.php?tabla=unidad | 1 |
| 5 Catálogos | Estatus | Grid.php?tabla=estatus | 1 |
| 6 Catálogos | Grupo | Grid.php?tabla=grupo | 1 |
| 7 Sistema Telemetría | Estación | Grid.php?tabla=estacion | 1 |
| 8 Catálogos | Tipo Estación | Grid.php?tabla=tipo_estacion | 1 |
| 9 Sistema Telemetría | Sistemas | Grid.php?tabla=sistema telemetria | 1 |
| 10 Sistema Telemetría | Sensor | Grid.php?tabla=sensor | 1 |
| 11 Catálogos | Tipo Sensor | Grid.php?tabla=tipo_sensor | 1 |
| 12 Sistema Telemetría | Aviso | Grid.php?tabla=aviso | 1 |
| 13 Sistema Telemetría | Detalle Sistemas | Grid.php?tabla=sistema telemetria&detalle=1 | 1 |
| 14 Botones | Mostrar mapa | Mapa.php?tabla=sistema telemetria | 1 |
| 15 Sistema Telemetría | Ayuda | Grid.php?tabla=sensor | 10 |
| 16 Vistas | Tipo Vistas | Grid.php?tabla=tipo_vista | 1 |
| 17 Vistas | Vista | Grid.php?tabla=vista | 1 |
| 18 Vistas | Grupo-Vistas | Grid.php?tabla=grupo_vista | 1 |
| 19 Catálogos | Grupo-Usuarios | Grid.php?tabla=grupo_usuario | 1 |
| 20 Catálogos | Tema | Grid.php?tabla=tema | 1 |

Mostrando 1 a 20 de 31 registros

Ilustración 12 Configuración – Página

Sistema Base de Gestión de Información de Telemetría 07:23:15 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

Catálogos > Configuración > Gestor de servicios web > Origen de Datos > Sistema Telemetría > Vistas >

MENU

Nuevo Eliminar Guardar Cancelar

| NOMBRE | DESCRIPCION |
|----------------------------|---------------------------------------|
| 1 Botones | Barra de botones |
| 2 Configuración | Apartado para menus |
| 3 Usuarios | Apartado para los usuarios |
| 4 Catálogos | Apartado para catalogos del sistema |
| 5 Sistema Telemetría | Apartado para las estaciones |
| 6 Sensores | Apartado para los sensores |
| 7 Avisos | Apartado para los avisos |
| 8 Vistas | Apartado para vistas |
| 9 Origen de Datos | Apartado para los origenes de datos |
| 10 Gestor de servicios web | Apartado para Gestor de servicios web |

Mostrando 1 a 10 de 10 registros

Ilustración 13 Configuración – Menú

SERVICIOS WEB

Sistema Base de Gestión de Información de Telemetría 07:25:17 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

1 Catálogos | Configuración | Gestor de servicios web | Origen de Datos | Sistema Telemetría | Vistas

CONSULTA

Nuevo Eliminar Guardar Cancelar

| NOMBRE | DESCRIPCIÓN | QUERY | PARAMETROS |
|----------------------------|--|--|------------------|
| 1 ListaUsuarios | Retorna todos los usuarios del sistema | select * from usuario | |
| 2 GetEstacion | Retorna la informacion de una estacion | select * from estacion where idestacion={estacion} | {estacion} |
| 3 GetEstacion_TipoYSistema | Retorna las estaciones filtrado por tipo y sistema | select * from estacion where fk_sistema={sistema} and fk_ti {sistema},{tipo} | {sistema},{tipo} |
| 4 GetLatitudEstacion | Retorna la latitud de la estacion | select latitud from estacion where idestacion={estacion} | {estacion} |

Mostrando 1 a 4 de 4 registros

Ilustración 14 Servicios web - Consulta

ORIGEN DE DATOS

Sistema Base de Gestión de Información de Telemetría 07:49:30 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

1 Catálogos | Configuración | Gestor de servicios web | Origen de Datos | Sistema Telemetría | Vistas

FUENTE DATOS

| NOMBRE | BD | TABLA | CAMPO VALOR | CAMPO FECHA | CAMPO LLAVE | SERVIDOR |
|-----------|-----|-----------|-------------|-------------|-------------|--------------------------------|
| 1 Origen1 | sih | dtempaire | Valuee | Datee | Station | Estaciones Climaticas Nacional |

Mostrando 1 a 1 de 1 registros

Ilustración 15 Origen de datos - Fuente de Datos

Sistema Base de Gestión de Información de Telemetría 07:47:54 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

1 Catálogos | Configuración | Gestor de servicios web | Origen de Datos | Sistema Telemetría | Vistas

Origen de Datos

Origen de datos

Nombre:

Servidor:

Base de datos:

Estaciones

Tabla de estaciones:

Clave de la estación (Se relacionará con el sensor):

Nombre de la estación:

Latitud:

Longitud:

Valores

Tabla de valores:

Clave (Campo que se relaciona con la estación):

Fecha-Hora (Fecha y hora del valor medido):

Valor (Campo con el valor medido):

Sensores

Tipo de sensor:

Extracción de datos

Frecuencia:

[Aceptar](#)

Ilustración 16 Origen de Datos - Nuevo Origen de Datos

SISTEMA DE TELEMETRIA

Sistema Base de Gestión de Información de Telemetría 07:52:06 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

Catálogos | Configuración | Gestor de servicios web | Origen de Datos | Sistema Telemetría | Vistas

SENSOR

Nuevo Eliminar Guardar Cancelar

| ID | NOMBRE | DESCRIPCION | ESTATUS | FRECUENCIA COMUNICACION | LATTITUD | LONGITUD | TIPO SENSOR | ESTACION |
|----|----------------------------|--------------------------------|---------|-------------------------|-----------|-------------|------------------------|--------------------------------|
| 1 | Sensor1 | Sensor1 | Activo | 1 | 18.885147 | -99.158953 | Humedad | IMTA |
| 2 | Sensor2 | Sensor2 | Activo | 1 | 18.885147 | -99.158953 | Presión barometrica_m | IMTA |
| 3 | ACAPONETA Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 22.479444 | -105.403333 | Precipitación Genérico | Acaponeta, Nay. SMN* |
| 4 | ACAPULCOob Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 16.762778 | -99.749167 | Precipitación Genérico | Observatorio Acapulco, Gro. SM |
| 5 | ACEQUIAMADRE Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 31.76 | -106.51 | Precipitación Genérico | Acequia Madre, Chih. GASIR* |
| 6 | ACHAPALAGDL Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 20.316667 | -103.133333 | Precipitación Genérico | Acueducto Chapala-GuadalaJarr |
| 7 | ACOLMAN Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 19.634722 | -98.911667 | Precipitación Genérico | Acolman, Méx. GASIR* |
| 8 | ACTOPAN Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 20.280556 | -98.9725 | Precipitación Genérico | Actopan, Hgo. SMN* |
| 9 | ACUGR Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 16.762778 | -99.749167 | Precipitación Genérico | Acapulco, Gro. SMN* |
| 10 | ADJQT Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 20.592222 | -100.490833 | Precipitación Genérico | Las Adjuntas, Qro. GASIR* |
| 11 | ALTAMIRAob Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 22.3875 | -97.925556 | Precipitación Genérico | Observatorio Altamira, Tamps. |
| 12 | ALTZOMONI Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 19.118611 | -98.655278 | Precipitación Genérico | Altzomoni, Mex. SMN* |
| 13 | ALVAROBREGON Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 21.193056 | -100.977778 | Precipitación Genérico | Álvoro Obregón, Gto. GASIR* |
| 14 | ALZATE Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 19.483333 | -99.65 | Precipitación Genérico | Alzate, Méx. GASIR* |
| 15 | AMZDR Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 25.263333 | -104.066111 | Precipitación Genérico | Agustín Melgar, Dgo. SMN* |
| 16 | AMECAMECA Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 19.140556 | -98.805556 | Precipitación Genérico | Ameameca, Méx. SMN* |
| 17 | AMZQT Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 20.704167 | -100.269444 | Precipitación Genérico | Amazcala, Qro. GASIR* |
| 18 | ANGMC Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 20.125556 | -101.722778 | Precipitación Genérico | Angamacutiro, Mich. SMN* |
| 19 | APAMC Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 19.082778 | -102.371667 | Precipitación Genérico | Apatzingan, Mich. SMN* |
| 20 | ARMMC Precipitacion | Sensor Genérico de Precipitaci | Activo | 1 | 19.925556 | -101.669722 | Precipitación Genérico | Aristeo Mercado, Mich. GASIR* |

Mostrando 1 a 20 de 941 registros

Ilustración 17 Sistema de Telemetría – Sensores

Sistema Base de Gestión de Información de Telemetría 07:51:37 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

Catálogos | Configuración | Gestor de servicios web | Origen de Datos | Sistema Telemetría | Vistas

ESTACION

Nuevo Eliminar Guardar Cancelar

| ID | NOMBRE | DESCRIPCION | LATTITUD | LONGITUD | TIPO ESTACION | SISTEMA TELEMETRIA |
|----|-----------------------------|----------------|-----------|-------------|-----------------|--------------------|
| 1 | IMTA | E.M. Auditorio | 18.885147 | -99.158953 | Meteorologico A | Climático Morelos |
| 2 | Arizpe, Son. SMN* | ARZSO | 30.335833 | -110.1675 | Nacional | Climático Nacional |
| 3 | Agustín Melgar, Dgo. SMN* | AMZDR | 25.263333 | -104.066111 | Nacional | Climático Nacional |
| 4 | Acequia Madre, Chih. GASIR* | ACEQUIAMADRE | 31.76 | -106.51 | Nacional | Climático Nacional |
| 5 | Arroyo Blanco, Coah. GASIR* | ARROYOBLANCO | 28.89 | -100.62 | Nacional | Climático Nacional |
| 6 | Colombia, N.L. GASIR* | COLOMBIA | 27.699167 | -99.746944 | Nacional | Climático Nacional |
| 7 | El Amole, Coah. GASIR* | ELAMOLE | 28.12 | -100.12 | Nacional | Climático Nacional |
| 8 | El Azucar, Tamps. GASIR* | ELAZUCAR | 26.250278 | -98.850556 | Nacional | Climático Nacional |
| 9 | El Chapeño, Tams. SMN* | ELCHAPENO | 26.526944 | -99.138611 | Nacional | Climático Nacional |
| 10 | El Coyote, Tams. GASIR* | ELCOYOTE | 27.435556 | -99.526389 | Nacional | Climático Nacional |
| 11 | El Moral, Coah. GASIR* | ELMORAL | 28.9 | -100.62 | Nacional | Climático Nacional |
| 12 | El Morillo, Tams. GASIR* | ELMORILLO | 26.1525 | -98.384167 | Nacional | Climático Nacional |
| 13 | El Mulato, Chih. GASIR* | ELMULATO | 29.48 | -104.22 | Nacional | Climático Nacional |
| 14 | El Nogal, Chih. SMN* | ELNOGAL | 29.42 | -104.26 | Nacional | Climático Nacional |
| 15 | El Puente, Tamps. GASIR* | ELPUENTE | 26.790833 | -99.421944 | Nacional | Climático Nacional |
| 16 | El Umbral, Tams. GASIR* | ELUMBRAL | 26.04 | -98.04 | Nacional | Climático Nacional |
| 17 | Jimenez, Coah. GASIR* | JIMENEZ | 29.05 | -100.663889 | Nacional | Climático Nacional |
| 18 | La Fragua, Coah. GASIR* | LAFRAGUA | 10.083333 | -24 | Nacional | Climático Nacional |
| 19 | Matamoros, Camp. INIFAP* | MATAMOROS | 18.591944 | -90.641667 | Nacional | Climático Nacional |
| 20 | N L Rio Bravo, N.L. GASIR* | NLRIOBRAVO | 27.498333 | -99.506944 | Nacional | Climático Nacional |

Mostrando 1 a 20 de 1942 registros

Ilustración 18 Sistema de Telemetría - Estaciones

Sistema Base de Gestión de Información de Telemetría 07:50:54 AM (UTC)

Usuario: root [Cerrar Sesión](#)

[Mostrar mapa](#)

Catálogos ▾ Configuración ▾ Gestor de servicios web ▾ Origen de Datos ▾ Sistema Telemetría ▾ Vistas ▾

| SISTEMA TELEMETRIA | |
|--|---|
| NOMBRE | DESCRIPCION |
| 1  Climatico Morelos | Sistema de telemetría para estaciones meteo |
| 2  Climático Nacional | Sistema Climático Nacional |
| 3  Presas Nacional | Sistema de Presas Nacional |
| 4  Origen1 | Sistema creado desde un origen de datos |
| 5  IMTA | Sistema de medidores ultrasónicos en IMTA |

20 |   Página 1 de 1   Mostrando 1 a 5 de 5 registros

Ilustración 19 Sistema de Telemetría por región

6 Conclusiones

Contenido

7 Bibliografía

Contenido

8 Anexos

Anexo 1. Código fuente del sistema informático Web para la gestión de datos de telemetría

Se enlistan las carpetas del proyecto mismas que contienen los scripts generadores de los módulos y componentes propios del sistema.



Ilustración 18 Carpetas del sistema

AJAX

```
# NOMBRE
# crear_sensores.php
# VARIABLES
# $dump
# $conexionLocal
# $tablas
# MODULOS
# /Class/SQL.php
# /Class/Dump.php
# /Class/JSON.php
# /Config/DB.php
```

```
<?php
set_time_limit(0);
require_once '../Class/SQL.php';
require_once '../Class/Dump.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';
//require_once '../Config/DB_estaciones.php';

//Conexion local
//$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);
//$sensores = array("dtprecipitacio"=>8, "dtpresbarometr"=>7, "dttempaire"=>6);
$dump = new Dump($conexionLocal, $conexionLocal);
$tablas = array(
```

```
// "sensor" => array(  
//      "sql"=>"select distinct s.idestacion fk_estacion, concat('Sensor ',v.station,'  
Precipitacion') nombre, concat('Sensor Genérico de Precipitacion estacion:',v.station)  
descripcion, 1 fk_estatus, 1 frecuencia_comunicacion,s.latitud,s.longitud, 8 fk_tipo_sensor  
from dtprecipitacio v inner join estacion s on s.descripcion=v.station",  
//      "crearTabla"=>false),  
      "sensor" => array(  
          "sql"=>"select distinct s.idestacion fk_estacion, concat('Sensor ',v.station,' Presión  
barometrica') nombre, concat('Sensor Genérico de Precipitacion estacion:',v.station)  
descripcion, 1 fk_estatus, 1 frecuencia_comunicacion,s.latitud,s.longitud, 7 fk_tipo_sensor  
from dtpresbarometr v inner join estacion s on s.descripcion=v.station",  
          "crearTabla"=>false)  
// "sensor" => array(  
//      "sql"=>"select distinct s.idestacion fk_estacion, concat('Sensor ',v.station,'  
Temperatura del aire') nombre, concat('Sensor Genérico de Precipitacion estacion:',v.station)  
descripcion, 1 fk_estatus, 1 frecuencia_comunicacion,s.latitud,s.longitud, 6 fk_tipo_sensor  
from dttempaire v inner join estacion s on s.descripcion=v.station",  
//      "crearTabla"=>false)  
    );  
  
$dump->Importar($tablas);  
?>
```

```
# NOMBRE  
# delete_data.php  
# VARIABLES  
# $sql  
# $tabla
```

```
# $_bd
# $res
# $linea
# $archivos
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Class/Fichero.php
# /Config/DB.php
*****

<?php
/*
 * Archivo para eliminar los datos del grid
 */
session_start();
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Class/Fichero.php';
require_once '../Config/DB.php';

//$_bd = "INFORMATION_SCHEMA";
$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd_IS, $_puertoSql);

$tabla = $_REQUEST["tabla"];
$id = $_REQUEST["id"];

$lave = $sql->Consulta("laveTabla", array("<tabla>"=>$tabla));
$lave = $lave[0]["pk"];

$_bd = "sistema_telemetria";
$sql->UsaBD($_bd);

//Acciones antes de eliminar
switch($tabla){
    case "web_service":
        $query = $sql->Consulta("consultaWebServiceIdConsulta", array("<consulta>"=>$id));
        $nombre = $query[0]["nombre"];
        $parametros = explode(",", $query[0]["parametros"]);
        $query = $query[0]["query"];
        //Reemplaza parametros
        foreach($parametros as $parametro){
            $query = str_replace($parametro, str_replace(array("{","}"), array("<",">"),
            $parametro), $query);
        }
    }
}
```



```
    }
    //Texto a escribir Json
    $textoJson = "\t".":".$nombre.":".$query.":".PHP_EOL;
    break;
}

//Eliminar registro
$res = $sql->Eliminar($tabla, $llave, $id);

//Acciones despues de eliminar
switch($tabla){
    case "web_service":
        //Escribir en JSON
        $fichero = new Fichero("../JSON/ConsultasWebService.json");
        $fichero->EliminarLinea($textoJson);
        break;
    case "fuente_datos":
    case "publicacion_fuente_datos":
        $archivos = $sql->Consulta("frecuencias", array());
        foreach($archivos as $archivo){
            $lineas = $sql->Consulta("generaContenidoArchivo",
array("<frecuencia>"=>$archivo["frecuencia"]));
            $texto = "";
            foreach($lineas as $linea){
                $texto .= $linea["linea"].PHP_EOL;
            }
            $fichero = new Fichero("../OrigenDatos/".$archivo["idfrecuencia"].".txt");
            $fichero->EscribirContenido($texto);
        }
        break;
}

//Resultado
if($res != false){
    echo json_encode(array("success"=>$res));
}
else{
    echo json_encode(array(
        'isError' => true,
        'msg' => 'El registro no se ha podido eliminar.<br><br>'.$sql->getError()
    ));
}
?>
```

```
# NOMBRE
# dump.php
# VARIABLES
# $sql
# $tabla
# $_bd
# $res
# $linea
# $archivos
# MODULOS
# /Class/SQL.php
# /Class/Dump.php
# /Class/JSON.php
# /Config/DB.php
# /Config/DB_estaciones.php
```

```
<?php
set_time_limit(0);
require_once '../Class/SQL.php';
require_once '../Class/Dump.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';
require_once '../Config/DB_estaciones.php';

$dump = new Dump($conexionLocal, $conexionEstaciones, true);

$tablas = array(
    "stations" => array("sql"=>"select * from stations", "crearTabla"=>true)
    //"dtprecipitacio" => array("sql"=>"select * from dtprecipitacio where datee between
'2014-06-10' AND '2014-06-12'", "crearTabla"=>true),
    //"dtpresbarometr" => array("sql"=>"select * from dtpresbarometr where datee between
'2014-06-10' AND '2014-06-12'", "crearTabla"=>true),
    //"dttempaire" => array("sql"=>"select * from dttempaire where datee between '2014-
06-10' AND '2014-06-12'", "crearTabla"=>true)
);

$dump->Importar($tablas);
?>
```

```
# NOMBRE
# dump_import.php
# VARIABLES
# $sql
# $_mysql
# $_userSql
# $_passSql
# $_bd
# $_puertoSql
# $tipo_estacion
# $sistema_climatico
# $dump
# $tablas
# MODULOS
# /Class/SQL.php
# /Class/Dump.php
# /Class/JSON.php
# /Config/DB.php
```

```
<?php
set_time_limit(0);
require_once '../Class/SQL.php';
require_once '../Class/Dump.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';
//require_once '../Config/DB_estaciones.php';

//Conexion local
$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);

////Insercion del sistema de telemetria
//$datos_st = array("nombre"=>"Climático Nacional", "descripcion"=>"Sistema Climático
Nacional");
//$sistema_climatico = $sql->Insertar("sistema_telemetria", $datos_st);
//
//echo "Sistema Telemetria: ".$sistema_climatico."<br>";
//
////Insercion del sistema de telemetria
//$datos_te = array("nombre"=>"Estación Climática Nacional", "descripcion"=>"Estacion
Climática Nacional");
```

```
// $tipo_estacion = $sql->Insertar("tipo_estacion", $datos_te);  
//  
// echo "Tipo Estacion: ".$tipo_estacion."<br>";  
$tipo_estacion = 4;  
$sistema_climatico = 3;  
  
$dump = new Dump($conexionLocal, $conexionLocal);  
$tablas = array(  
    "estacion" => array(  
        "sql" => "select stationName nombre, station descripcion, latitud2 latitud, longitud2  
longitud, ".$tipo_estacion." fk_tipo_estacion, ".$sistema_climatico." fk_sistema_telemetria  
from stations where stationName like '%*%'",  
        "crearTabla" => false)  
    );  
  
$dump->Importar($tablas);  
?>
```

```
# NOMBRE  
# feeder.php  
# VARIABLES  
# $sql  
# $fechas  
# $tablasActualizar  
# $fechaMaxima  
# $tabla  
# $dump  
# $conexionLocal  
# $tipo_sensor  
# $conexionEstaciones  
# $query  
# MODULOS  
# /Class/SQL.php  
# /Class/Dump.php  
# /Class/JSON.php  
# /Config/DB_estaciones.php  
# /Config/DB.php
```

```
<?php  
set_time_limit(0);
```

```
require_once '../Class/SQL.php';
require_once '../Class/Dump.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';
require_once '../Config/DB_estaciones.php';

//Conexion local
$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);
$fechas = array();

$tablasActualizar = array("dtprecipitacio", "dtpresbarometr", "dttempaire");
$tablas = array();
foreach($tablasActualizar as $tabla){
    //Fecha maxima por tabla
    $fechaMaxima = $sql->Consulta("maxDatee", array("<tabla>"=>$tabla));
    $fechaMaxima = $fechaMaxima[0]["fecha"];
    //$fechaMaxima = "2014-06-17 09:10:00";
    $fechas[$tabla] = $fechaMaxima;
    //Consulta para sacar los valores apartir de la ultima fecha
    $tablas[$tabla] = array("sql"=>"select * from ".$tabla." where Datee > '". $fechaMaxima.'"
and Datee <= UTC_TIMESTAMP() order by Datee", "crearTabla"=>false);
}

//Importar datos a tabla local print_r($tablas);
$dump = new Dump($conexionLocal, $conexionEstaciones);
$dump->Importar($tablas);

//Importar datos a la tabla de bitacora
$dump = new Dump($conexionLocal);

//Tablas por tipo de sensor y fecha maxima obtenida
$tipo_sensor = array(
    "dttempaire"=>array("tipo"=>6, "fecha"=>$fechas["dttempaire"]),
    "dtpresbarometr"=>array("tipo"=>7, "fecha"=>$fechas["dtpresbarometr"]),
    "dtprecipitacio"=>array("tipo"=>8, "fecha"=>$fechas["dtprecipitacio"])
);
//Recorrido de tablas para importar a bitacora_valores
foreach($tipo_sensor as $tabla => $dato){
    $query = "select Valuee valor,Datee tstamp,s.idsensor fk_sensor "
        . "from ".$tabla." d inner join estacion e on e.descripcion=d.station inner join sensor s
on s.fk_estacion=e.idestacion "
        . "where fk_tipo_sensor='".$dato["tipo"]." and Datee>".$dato["fecha"]." order by
Datee";
```

```
//echo $query."<br>";
$tablas = array("bitacora_valores" => array("sql" => $query, "crearTabla" => false));
$dump->Importar($tablas);
//Eliminar datos anteriores, solo se queda la ultima importacion desde host externo
$total = $sql->Consulta($query);
if(sizeof($total)>0)
    $r = $sql->Query("DELETE FROM ".$tabla." WHERE Datee <= '". $dato["fecha"]."'");
}
?>
```

```
# NOMBRE
# get_combobox .php
# VARIABLES
# $sql
# $tabla
# $item
# $valores
# $default
# $_mysql
# $_userSql
# $_passSql
# $_bd
# $_puertoSql
# $tabla
# $item
# $valores
# $default
# $result
# $servidor
# $sqlRemoto
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Config/DB.php
```

```
<?php
```

```
/*
 * Archivo para obtener los datos para el grid
 */
```

```
session_start();
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';

$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);

$tabla = $_REQUEST["tabla"];
$item = isset($_REQUEST["item"]) ? $_REQUEST["item"] : NULL;

// $result = $sql->Consulta("genericComboBox", array("<tabla>" => $tabla));

function ComboDatos($valores, $default = true){
    if(sizeof($valores)==0)
        $valores[0] = array("id"=>"-1", "text"=>"No hay datos");

    if($default==true)
        $valores[0]["selected"] = true;

    return $valores;
}

switch ($tabla) {
    case "sistema_telemetria":
        $result = $sql->Consulta("comboSistemasMapa", array("<usuario>" =>
$_SESSION["idUsuario"]));
        // $result = $sql->Consulta("genericComboBox", array("<tabla>" => $tabla,
"<filtro>"=>""));
        $selected = $sql->Consulta("ultimoSistemaVisto", array("<usuario>" =>
$_SESSION["idUsuario"]));
        $selected = $item==null ? $selected[0]["id"] : $item;
        if($selected!=""){
            foreach ($result as $index => $fila){
                if($fila["id"]== $selected)
                    $result[$index]["selected"] = true;
            }
        }
        else{
            $result[0]["selected"] = true;
        }
        break;
    case "servidor":
```

```
$result = $sql->Consulta("genericComboBox", array("<tabla>" => $tabla,
"<filtro>"=>""));
$result = ComboDatos($result, false);
break;
case "estacion_fuente":
    $result = $sql->Consulta("estacionesFuente", array());
    $result = ComboDatos($result);
    break;
case "sensor_fuente_":
    $result = $sql->Consulta("sensoresFuente", array("<estacion>" => $_REQUEST["id"]));
    $result = ComboDatos($result, false);
    break;
case "sensor_fuente":
    $result = $sql->Consulta("sensoresFuente", array());
    $result = ComboDatos($result, false);
    break;
case "bd_servidor":
    $servidor = $sql->Consulta("datosServidor", array("<servidor>"=>$_REQUEST["id"]));
    $servidor = $servidor[0];
    $sqlRemoto = new SQL($servidor["direccion"], $servidor["usuario"],
    $servidor["password"], "information_schema", $servidor["puerto"]);
    $result = $sqlRemoto->Consulta("showDataBases", array());
    $result = ComboDatos($result);
    break;
case "tabla_servidor":
    $servidor = $sql->Consulta("datosServidor",
    array("<servidor>"=>$_REQUEST["servidor"]));
    $servidor = $servidor[0];
    $sqlRemoto = new SQL($servidor["direccion"], $servidor["usuario"],
    $servidor["password"], "information_schema", $servidor["puerto"]);
    $result = $sqlRemoto->Consulta("showTables", array("<bd>" => $_REQUEST["id"]));
    $result = ComboDatos($result);
    break;
case "campo_fecha":
    $servidor = $sql->Consulta("datosServidor",
    array("<servidor>"=>$_REQUEST["servidor"]));
    $servidor = $servidor[0];
    $sqlRemoto = new SQL($servidor["direccion"], $servidor["usuario"],
    $servidor["password"], "information_schema", $servidor["puerto"]);
    $result = $sqlRemoto->Consulta("columnasFecha", array("<bd>" => $_REQUEST["bd"],
    "<tabla>" => $_REQUEST["id"]));
    $result = ComboDatos($result);
    break;
```



```

case "campo_llave":
case "campo_valor":
    $servidor = $sql->Consulta("datosServidor",
array("<servidor>"=>$ _REQUEST["servidor"]));
    $servidor = $servidor[0];
    $sqlRemoto = new SQL($servidor["direccion"], $servidor["usuario"],
    $servidor["password"], "information_schema", $servidor["puerto"]);
    $result = $sqlRemoto->Consulta("columnasLlaveValor", array("<bd>" =>
    $ _REQUEST["bd"], "<tabla>" => $ _REQUEST["id"]));
    $result = ComboDatos($result);
    break;
case "llaves_":
    $servidor = $sql->Consulta("datosServidor",
array("<servidor>"=>$ _REQUEST["servidor"]));
    $servidor = $servidor[0];
    $sqlRemoto = new SQL($servidor["direccion"], $servidor["usuario"],
    $servidor["password"], $ _REQUEST["bd"], $servidor["puerto"]);
    $result = $sqlRemoto->Consulta("distinctLlaves", array("<tabla>" =>
    $ _REQUEST["table"], "<llave>" => $ _REQUEST["id"]));
    $result = ComboDatos($result);
    break;
case "llaves":
    $servidor = $sql->Consulta("datosFuente", array("<fuente>"=>$ _REQUEST["fuente"]));
    $servidor = $servidor[0];
    $sqlRemoto = new SQL($servidor["direccion"], $servidor["usuario"],
    $servidor["password"], $servidor["bd"], $servidor["puerto"]);
    $usados = $sql->Consulta("llavesUsadas", array("<comilla>"=>"\",
"<fuente>"=>$ _REQUEST["fuente"]));
    $filtro = $usados[0]["llaves"] != "" || $usados[0]["llaves"] != null ? " and
".$servidor["campo_llave"]." not in (\".$usados[0][\"llaves\"].\") : ";
    $result = $sqlRemoto->Consulta("distinctLlaves", array("<tabla>" => $servidor["tabla"],
"<llave>" => $servidor["campo_llave"], "<filtro>"=>$filtro));
// $result = $sqlRemoto->Consulta("select distinct Station from ddprecipitacio where
Station is not null", array("<tabla>" => $ _REQUEST["table"], "<llave>" =>
$ _REQUEST["id"]));
    $result = ComboDatos($result);
    break;
default:
    $result = $sql->Consulta("genericComboBox", array("<tabla>" => $tabla,
"<filtro>"=>""));
    $result = ComboDatos($result);
    break;
}

```

```
echo json_encode($result);  
?>
```

```
*****
```

```
# NOMBRE  
# get_combobox_image.php  
# VARIABLES  
# $sql  
# $_mysql  
# $_userSql  
# $_passSql  
# $_bd  
# $_puertoSql  
# $tabla  
# $filtro  
# $result  
# $directorio  
# $archivo  
# MODULOS  
# /Class/SQL.php  
# /Class/JSON.php  
# /Config/DB.php
```

```
*****
```

```
<?php  
/*  
 * Archivo para obtener los datos para el grid  
 */  
session_start();  
require_once '../Class/SQL.php';  
require_once '../Class/JSON.php';  
require_once '../Config/DB.php';  
  
$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);  
  
$tabla = $_REQUEST["tabla"];  
$filtro = "";  
  
$result = array();  
$directorio = opendir("../Image/".$tabla); //ruta actual  
while ($archivo = readdir($directorio)){ //obtenemos un archivo y luego otro sucesivamente
```

```
if (is_dir($archivo)){//verificamos si es o no un directorio
}
else{
    $result[] = array("id"=>$archivo, "nombre"=>substr($archivo, 0, strlen($archivo)-4));
}
}

echo json_encode($result);
?>
```

```
# NOMBRE
# get_data.php
# VARIABLES
# $_schema
# $_bd
# $_mysql
# $_userSql
# $_passSql
# $_bd_IS
# $_puertoSql
# $tabla
# $filtro
# $order
# $sort
# $page
# $rows
# $columnas
# $sql
# $campos
# $inners
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Config/DB.php
```

```
<?php
/*
 * Archivo para obtener los datos para el grid
 */
session_start();
```

```
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';

$_schema = $_bd;
//$_bd = "INFORMATION_SCHEMA";
$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd_IS, $_puertoSql);

$tabla = $_REQUEST["tabla"];
$filtro = $_REQUEST["filtro"];
$order = $_REQUEST["order"];
$sort = (strpos($_REQUEST["sort"],"fk_") === 0) ? str_replace("fk_", "",
$_REQUEST["sort"])."nombre" : $tabla.".".$_REQUEST["sort"];
$page = $_REQUEST["page"];
$rows = $_REQUEST["rows"];

//echo json_encode($_REQUEST);
//exit();
$columnas = $sql->Consulta("columnasLlaves", array("<tabla>"=>$tabla,
"<schema>"=>$_schema));

switch($tabla){
    case "grupo":
        $filtro .= ($filtro=="" ? " WHERE" : " AND") . " grupo.nombre NOT LIKE
'Compartido_%'";
        break;
    case "grupo_vista":
    case "grupo_usuario":
        $filtro .= ($filtro=="" ? " WHERE" : " AND") . " ".$tabla.".fk_grupo not in (select idgrupo
from grupo where nombre LIKE 'Compartido_%')";
        break;
    default:
        break;
}

$campos = "";
$inners = $tabla;
foreach($columnas as $columna){
    $campo = "";

// if($columna["nombre"]!="fk_marcador_sistema"){
    if($columna["columna"]!=""){
        $campo = $columna["tabla"]."nombre AS ".$columna["nombre"];
    }
}
```

```
        $inners      .=      "      INNER      JOIN      ".$columna["tabla"]."      ON
".$columna["tabla"]." ".$columna["columna"]."=".$tabla"." ".$columna["nombre"];
    }
    else{
        $campo = $tabla"." ".$columna["nombre"];
    }
// }

    $campos .= ($campos == "" ? "" : ",") . $campo;
}

//$_bd = "sistema_telemetria2";
$sql->UsaBD($_bd);

$datos = $sql->Consulta("datosTabla", array("<campos>"=>$campos, "<tabla>"=>$inners,
"<filtro>"=>$filtro, "<order>"=>$order, "<sort>"=>$sort, "<inicio>"=>(($page-1)*$rows),
"<registros>"=>$rows));
//echo      json_encode(array("<campos>"=>$campos,      "<tabla>"=>$inners,
"<filtro>"=>$filtro, "<order>"=>$order, "<sort>"=>$sort, "<inicio>"=>(($page-1)*$rows),
"<registros>"=>$rows));
$total = $sql->Consulta("totalDatos", array("<tabla>"=>$tabla, "<filtro>"=>$filtro));

$result = array("rows"=>$datos, "total"=>$total[0]["total"]);

echo json_encode($result);
?>

*****
# NOMBRE
# get_detail.php
# VARIABLES
# $sql
# $_mysql
# $_userSql
# $_passSql
# $_bd
# $_puertoSql
# $estaciones
# $html
# $dato
# $llave
```

```
# $sensores
# $titulosSe
# $datoSe
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Config/DB.php
*****
<?php

/*
 * Archivo para obtener los datos para el grid
 */
session_start();
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';

$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);

//print_r($_REQUEST);
switch ($_REQUEST["tabla"]) {
    case "fuente_datos":
        $estaciones = $sql->Consulta("estacionesSistemaOrigenDatos", array("<fuente>" =>
$_REQUEST["id"]));
        $html = "<div id='divDetail' style='height:".($_REQUEST["alto"]-6)."px'>";
        $html .= "<table id='headDetail'>";
        foreach ($estaciones as $index => $estacion) {
            $html .= "<tr>";
            foreach ($estacion as $llave => $dato) {
                if ($llave != "idestacion")
                    $html .= "<td><b>" . str_replace("_", " ", strtoupper($llave)) . "</b>: " . $dato .
"</td>";
            }
            $html .= "</tr>";
            $html .= "<tr><td colspan='". (sizeof($estacion) - 1) . "'>";
            $sensores = $sql->Consulta("sensoresEstacion", array("<estacion>" =>
$estacion["idestacion"]));
            $html .= "<table id='bodyDetail'>";
            if (sizeof($sensores) > 0) {
                foreach ($sensores as $indexSe => $sensor) {
                    if ($indexSe == 0) {
                        $titulosSe = array_keys($sensor);
```

```

    $html .= "<tr>";
    foreach ($titulosSe as $datoSe) {
        if ($datoSe != "idsensor")
            $html .= "<td><b><i>" . str_replace("_", " ", strtoupper($datoSe)) .
"</i></b></td>";
        }
    }
    $html .= "</tr>";
}
$html .= "<tr>";
foreach ($sensor as $llaveDatoSe => $datoSe) {
    if ($llaveDatoSe != "idsensor")
        $html .= "<td>" . $datoSe . "</td>";
    }
    $html .= "</tr>";
}
} else {
    $html .= "<tr><td><b> - Sin Sensores</b></td></tr>";
}
}
$html .= "</table></td></tr>";
}
$html .= "</table></div>";
echo $html;
break;
case "sistema_telemetria":
    $estaciones = $sql->Consulta("estacionesSistema", array("<sistema>" =>
$_REQUEST["id"]));
    $html = "<div id='divDetail' style='height:'.($_REQUEST["alto"]-6).'px'>";
    $html .= "<table id='headDetail'>";
    foreach ($estaciones as $index => $estacion) {
//        if ($index == 0) {
//            $titulos = array_keys($estacion);
//            $html .= "<tr>";
//            foreach ($titulos as $dato) {
//                if ($dato != "idestacion")
//                    $html .= "<td><b>" . str_replace("_", " ", strtoupper($dato)) . "</b></td>";
//            }
//            $html .= "</tr>";
//        }
        $html .= "<tr>";
        foreach ($estacion as $llave => $dato) {
            if ($llave != "idestacion")
                $html .= "<td><b>" . str_replace("_", " ", strtoupper($llave)) . "</b>: " . $dato .
"</td>";

```

```

}
$html .= "</tr>";
$html .= "<tr><td colspan='". (sizeof($estacion) - 1) . "'>";
$sql = "SELECT * FROM sensores WHERE estacion = '$estacion'";
$sensores = $sql->Consulta("sensoresEstacion", array("<estacion>" =>
$estacion["idestacion"]));
$html .= "<table id='bodyDetail'>";
if (sizeof($sensores) > 0) {
    foreach ($sensores as $indexSe => $sensor) {
        if ($indexSe == 0) {
            $titulosSe = array_keys($sensor);
            $html .= "<tr>";
            foreach ($titulosSe as $datoSe) {
                if ($datoSe != "idsensor")
                    $html .= "<td><b><i>" . str_replace("_", " ", strtoupper($datoSe)) .
" </i></b></td>";
            }
            $html .= "</tr>";
        }
        $html .= "<tr>";
        foreach ($sensor as $llaveDatoSe => $datoSe) {
            if ($llaveDatoSe != "idsensor")
                $html .= "<td>" . $datoSe . "</td>";
        }
        $html .= "</tr>";
    }
} else {
    $html .= "<tr><td><b> - Sin Sensores</b></td></tr>";
}
$html .= "</table></td></tr>";
}
$html .= "</table></div>";
echo $html;
break;
case "sensor":
    $valores = $sql->Consulta("valoresSensor", array("<sensor>" => $_REQUEST["id"]));
    $html = "<div id='divDetail' style='height:".($_REQUEST["alto"]-6)."px'>";
    $html .= "<table id='headDetail'>";

    if (sizeof($valores) > 0) {
        $titulos = array_keys($valores[0]);
        $html .= "<tr>";
        foreach ($titulos as $titulo) {
            $html .= "<td align='center'><b>" . strtoupper($titulo) . "<b></td>";
        }
    }
}

```



```
}  
$html .= "</tr>";  
$html .= "<tr><td colspan=" . sizeof($titulos) . ">";  
$html .= "<table id='bodyDetail'>";  
foreach ($valores as $valor) {  
    $html .= "<tr>";  
    foreach ($valor as $val) {  
        $html .= "<td>" . $val . "</td>";  
    }  
    $html .= "</tr>";  
}  
$html .= "</table>";  
$html .= "</td></tr>";  
}  
$html .= "</table></div>";  
echo $html;  
break;  
default:  
    echo "Caso no definido";  
    break;  
}  
?>
```

```
# NOMBRE  
# get_markerEstaciones.php  
# VARIABLES  
# $sql  
# $_mysql  
# $_userSql  
# $_passSql  
# $_bd  
# $_puertoSql  
# $sistema  
# $res  
# $tipo_sensor  
# $idmarcador  
# $case  
# $result  
# $titulos  
# $dato
```

```
# $datos
# $extensiones
# $files
# $file
# $ext
# $kml
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Config/DB.php
*****
    <?php

/*
 * Archivo para obtener los datos para el grid
 */
session_start();
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';

$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);

$sistema = $_REQUEST["sistema"];

$res     =     $sql->Actualizar("usuario",     "idusuario",     $_SESSION["idUsuario"],
array("sys_ultimo_sistema_telemetria" => $sistema));

$tipo_sensor     =     $sql->Consulta("sensorMarcadorSistema",     array("<sistema>"     =>
$sistema));

if (sizeof($tipo_sensor) > 0) {
    $idmarcador = $tipo_sensor[0]["idmarcador_sistema"];
    $tipo_sensor = $tipo_sensor[0]["fk_tipo_sensor"];
    $case = $sql->Consulta("rangoMarcador", array("<sistema>" => $idmarcador, "<comilla>"
=> "\""));
    $case = $case[0]["rango"];
    $result = $sql->Consulta("markerEstacionesRango", array("<sistema>" => $sistema,
"<sensor>" => $tipo_sensor, "<case>" => $case));
} else {
    $result = $sql->Consulta("makerEstaciones", array("<sistema>" => $sistema));
}
// $resultInfo = $sql->Consulta("infoMakerEstacion", array("<sistema>" => $sistema));
```

```

$titulos = null;

if ($result) {
    foreach ($result as $index => $fila) {
        $info = $sql->Consulta("infoMakerEstaciones", array("<estacion>" =>
$fila["idestacion"]));
        $html = "<table class='infoBoxMap'>";
        if (sizeof($info) > 0) {
            $html .= "<tr><td colspan='". sizeof($fila) . "'><a
href='Grid.php?tabla=sensor&detalle=1&filtro=fk_estacion=" . $fila["idestacion"] .
"&tituloGrid=" . $fila["estacion"] . "' title='Ultimos valores'> . $fila["estacion"] .
"</a></td></tr>";
            $titulos = array_keys($info[0]);
            $html .= "<tr>";
            foreach ($titulos as $dato) {
                $html .= "<th>" . str_replace("_", " ", $dato) . "</th>";
            }
            $html .= "</tr>";
            foreach ($info as $datos) {
                $html .= "<tr>";
                foreach ($datos as $dato) {
                    $html .= "<td>" . $dato . "</td>";
                }
                $html .= "</tr>";
            }
        } else {
            $html .= "<tr><td><h3>" . $fila["estacion"] . "</h3></td></tr>";
            $html .= "<tr><td><b>Sin sensores</b></td></tr>";
        }
    }

    //Agregar imagenes si existen
    $dir = "../Image/ESTACIONES/" . $fila["idestacion"] . "/";
    if (file_exists($dir)) {
        $extenciones = array("JPG", "PNG", "JPEG", "GIF", "BMP");
        $files = scandir($dir);
        natcasesort($files);
        if (count($files) >= 1) {
            $html .= "<tr><td colspan='3'>";
            // All files
            foreach ($files as $file) {
                if (file_exists($dir . $file) && $file != '.' && $file != '..' && !is_dir($dir . $file)) {
                    $ext = strtolower($file, ".");
                    $ext = strtoupper(substr($file, $ext+1));
                }
            }
        }
    }
}

```

```
        if(in_array($ext, $extenciones))
            $html .= "<img src=\"" . $dir . $file . "\"/>";
        }
    }
    $html .= "</td></tr>";
}
}

$html .= "</table>";
$result[$index]["html"] = $html;
}
//Nombre sistema
$nombre = $sql->Consulta("traduccion", array("<campo>" => "nombre", "<tabla>" =>
"sistema_telemetria", "<llave>" => "idsistema_telemetria", "<valor>" => $sistema));
$nombre = $nombre[0]["valor"];

$kml = file_exists("../Data/" . $nombre . ".kml") ? $nombre : false;

echo json_encode(array("res" => $result, "kml" => $kml));
} else {
    echo json_encode(array());
}
?>
```

```
# NOMBRE
# login.php
# VARIABLES
# $sql
# $_mysql
# $_userSql
# $_passSql
# $_bd
# $_puertoSql
# $res
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Config/DB.php
```

```
<?php
```

```
/*
 * Archivo para verificar el usuario y contraseña del login
 */
session_start();
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';

$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);

$res = $sql->Consulta("login",
array("<user>"=>$_REQUEST["usuario"],"<password>"=>$_REQUEST["password"]));

if(sizeof($res)>0){
    $_SESSION["idUsuario"] = $res[0]["idusuario"];
    $_SESSION["user"] = $res[0]["nombre"];
    $_SESSION["password"] = $res[0]["password"];
}
else{
    echo "error";
}
?>

*****
# NOMBRE
# logout.php
*****

<?php
/*
 * Archivo para verificar el usuario y contraseña del login
 */
session_start();

unset($_SESSION["idUsuario"]);
unset($_SESSION["password"]);
unset($_SESSION["user"]);
session_destroy();

echo "1";
?>
```

```
# NOMBRE
# sabe_data.php
# VARIABLES
# $sql
# $_mysql
# $_userSql
# $_passSql
# $_bd
# $_puertoSql
# $tabla
# $aux
# $posicion
# $query
# $parametros
# $complementoUrl
# $textoJson
# $fichero
# $campo
# $traduccion
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Class/Fichero.php
# /Config/DB.php
```

```
<?php
/*
 * Archivo para guardar los datos para el grid
 */
session_start();
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Class/Fichero.php';
require_once '../Config/DB.php';

$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);

$tabla = $_REQUEST["tabla"];
$aux = $_REQUEST;
unset($aux["tabla"]);
```

```
unset($aux["isNewRecord"]);

//Encriptacion si es campo password
if(isset($aux["password"]) && $tabla!="servidor"){
    $aux["password"] = md5($aux["password"]);
}

//Caso especial en tablas
switch($tabla){
    case "fuente_datos":
        $aux["tabla"] = $_REQUEST["tablacampo"];
        unset($aux["tablacampo"]);
        break;
    case "sensor":
        if($aux["latitud"]=="" && $aux["longitud"]==""){
            $posicion = $sql->Consulta("posicionEstacion",
array("<estacion>"=>$aux["fk_estacion"]));
            $aux["latitud"] = $posicion[0]["latitud"];
            $aux["longitud"] = $posicion[0]["longitud"];
        }
        break;
    case "web_service":
        $aux["sys_fk_usuario"] = $_SESSION["idUsuario"];
        $query = $sql->Consulta("consultaWebService",
array("<consulta>"=>$aux["fk_consulta"]));
        $parametros = explode(",", $query[0]["parametros"]);
        $query = $query[0]["query"];
        $complementoUrl = "";
        //Reemplaza parametros
        foreach($parametros as $parametro){
            $complementoUrl .= "&" . str_replace(array("{","}"), array("",""), $parametro) . "=" .
$parametro;
            $query = str_replace($parametro, str_replace(array("{","}"), array("<",">"),
$parametro), $query);
        }
        //Texto a escribir Json
        $textoJson = "\t'.'. '$aux["nombre"].'": '$query.''.PHP_EOL;

        $aux["url"] = "http://pnh.gob.mx/sistema_telemetria/Services/WebService.php?key="
. $aux["nombre"] . $complementoUrl;
        break;
}
}
```

```
//Insercion
$id = $sql->Insertar($tabla, $aux);

//Registro exitoso
if($id != false){
    //Caso especial en tablas
    switch($tabla){
        case "usuario":
            $idAux = $sql->Insertar("grupo", array("nombre"=>"Compartido_". $id,
"descripcion"=>"Grupo para vistas compartidas, automatico no visible"));
            $res = $sql->Insertar("grupo_usuario", array("fk_usuario"=>$id,
"fk_grupo"=>$idAux));
            break;
        case "web_service":
            //Escribir en JSON
            $fichero = new Fichero("../JSON/ConsultasWebService.json");
            $fichero->EscribirLineas($textoJson, "p");
            break;
        case "publicacion_fuente_datos":
            $archivos = $sql->Consulta("frecuencias", array());
            foreach($archivos as $archivo){
                $lineas = $sql->Consulta("generaContenidoArchivo",
array("<frecuencia>"=>$archivo["frecuencia"]));
                $texto = "";
                foreach($lineas as $linea){
                    $texto .= $linea["linea"].PHP_EOL;
                }
                $fichero = new Fichero("../OrigenDatos/".$archivo["idfrecuencia"].".txt");
                $fichero->EscribirContenido($texto);
            }
            break;
    }

    foreach($aux as $campo => $valor){
        if(strpos($campo,"fk_") === 0){
            $traduccion = $sql->Consulta("traduccion", array("<tabla>"=>str_replace("fk_", "",
$campo), "<llave>"=>str_replace("fk_", "id", $campo), "<campo>"=>"nombre",
"<valor>"=>$valor));
            $aux[$campo] = $traduccion[0]["valor"];
        }
    }

    $aux["id".$tabla] = $id;
}
```



```
    echo json_encode($aux);
}
//Registro erroneo
else{
    echo json_encode(array(
        'isError' => true,
        'msg' => 'El registro no se ha podido guardar.<br><br>'.$sql->getError()
    ));
}
?>
```

```
# NOMBRE
# sabe_origenDatos.php
# VARIABLES
# $sql
# $aux
# $servidor
# $sqlremoto
# $estaciones
# $msgAux
# $time
# $idSistema
# $idSensor
# $idFuente_datos_sensor
# $idPoblacion_fuente_datos
# $fichero
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Class/Fichero.php
# /Config/DB.php
```

```
<?php
/*
 * Archivo para guardar los datos para el grid
 */
session_start();
set_time_limit(0);
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
```

```
require_once '../Class/Fichero.php';
require_once '../Config/DB.php';

$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);

$aux = $_REQUEST;
$servidor = $sql->Consulta("datosServidor", array("<servidor>"=>$aux["fk_servidor"]));
$servidor = $servidor[0];
$sqlRemoto = new SQL($servidor["direccion"], $servidor["usuario"], $servidor["password"],
$aux["bd"], $servidor["puerto"]);

//Consulta de estaciones
$estaciones = $sqlRemoto->Consulta("getEstacionesRemotas", array(
    "<idestacion>"=>$aux["campo_estacion"],    "<nombre>"=>$aux["campo_nombre"],
"<estaciones>"=>$aux["bd"].".".$aux["estaciones"],
    "<fkestacion>"=>$aux["campo_llave"],    "<valores>"=>$aux["bd"].".".$aux["tabla"],
"<longitud>"=>$aux["campo_lon"], "<latitud>"=>$aux["campo_lat"]
));

$msgAux = "";
if(sizeof($estaciones)>0){
    //Creacion de la fuente de datos
    $time = $sql->Consulta("NowUTC", array());
    $time = $time[0]["time"];
    $idFuente_datos = $sql->Insertar("fuente_datos", array("nombre"=>$aux["nombre"],
"bd"=>$aux["bd"], "tabla"=>$aux["tabla"], "campo_valor"=>$aux["campo_valor"],
    "campo_fecha"=>$aux["campo_fecha"],    "campo_llave"=>$aux["campo_llave"],
"fk_servidor"=>$aux["fk_servidor"], "tstamp"=>$time), true);
    $msgAux .= "<br>".$sql->getError();

    if($idFuente_datos != false){
        //Creacion el sistema de telemetria
        $idSistema = $sql->Insertar("sistema telemetria", array("nombre"=>$aux["nombre"],
"descripcion"=>"Sistema creado desde un origen de datos"), true);
        $msgAux .= "<br>".$sql->getError();

        if($idSistema != false){
            //Creacion de las estaciones
            foreach($estaciones as $estacion){
                $idEstacion = $sql->Insertar("estacion", array("nombre"=>$estacion["nombre"],
"latitud"=>$estacion["latitud"], "longitud"=>$estacion["longitud"],
                "fk_sistema telemetria"=>$idSistema,    "fk_tipo_estacion"=>0,
"descripcion"=>"Estacion creada desde un origen de datos"), true);
```

```

$msgAux .= "<br>".$sql->getError();

//Creacion de los sensores (1 por estacion)
$idSensor = $sql->Insertar("sensor", array("nombre"=>$estacion["nombre"],
"descripcion"=>"Sensor creado desde un origen de datos", "fk_estatus"=>1,
"frecuencia_comunicacion"=>"0", "latitud"=>$estacion["latitud"],
"longitud"=>$estacion["longitud"], "fk_tipo_sensor"=>$aux["sensor"],
"fk_estacion"=>$idEstacion), true);
$msgAux .= "<br>".$sql->getError();

//Creacion de la relacion Sensor-Llave para la extraccion de datos mas adelante
$idFuente_datos_sensor = $sql->Insertar("fuente_datos_sensor",
array("fk_sensor"=>$idSensor, "fk_fuente_datos"=>$idFuente_datos,
"llave"=>$estacion["llave"]), true);
$msgAux .= "<br>".$sql->getError();

if($idEstacion == false || $idSensor == false || $idFuente_datos_sensor == false){
    $sql->Eliminar("sistema_telemetria", "idsistema_telemetria", $idSistema);
    $sql->Eliminar("fuente_datos", "idfuentes_datos", $idFuente_datos);
    echo json_encode(array(
        'isError' => true,
        'msg' => 'Error al crear la estación y el sensor '.$msgAux
    ));
    exit();
}
}

//Publicacion de la fuente
$idPublicacion_fuente_datos = $sql->Insertar("publicacion_fuente_datos",
array("fk_fuente_datos"=>$idFuente_datos, "fk_frecuencia"=>$aux["frecuencia"]));
$msgAux .= "<br>".$sql->getError();
if($idPublicacion_fuente_datos != false){
    $archivos = $sql->Consulta("frecuencias", array());
    //Añadir publicacion al cron
    foreach($archivos as $archivo){
        $lineas = $sql->Consulta("generaContenidoArchivo",
array("<frecuencia>"=>$archivo["frecuencia"]));
        $texto = "";
        foreach($lineas as $linea){
            $texto .= $linea["linea"].PHP_EOL;
        }
        $fichero = new Fichero("../OrigenDatos/".$archivo["idfrecuencia"].".txt");
        $fichero->EscribirContenido($texto);
    }
}

```

```
}
//Respuesta exitosa
echo json_encode(array(
    'isError' => false,
    'msg' => 'El origen de datos '.$aux["nombre"]." se ha creado ",
    'sistema' => $idSistema
));
}
else{
    $sql->Eliminar("sistema_telemetria", "idsistema_telemetria", $idSistema);
    $sql->Eliminar("fuente_datos", "idfuelle_datos", $idFuente_datos);
    echo json_encode(array(
        'isError' => true,
        'msg' => 'Error al publicar el origen de datos '.$msgAux
    ));
}
}
else{
    $sql->Eliminar("fuente_datos", "idfuelle_datos", $idFuente_datos);
    echo json_encode(array(
        'isError' => true,
        'msg' => 'Error al crear el sistema de telemetria '.$msgAux
    ));
}
}
else{
    echo json_encode(array(
        'isError' => true,
        'msg' => 'Error al crear el origen de datos '.$msgAux
    ));
}
}
else{
    echo json_encode(array(
        'isError' => true,
        'msg' => 'No hay coincidencia con las estaciones'
    ));
}
?>
```

```
# NOMBRE
# update_data.php
# VARIABLES
# $sql
# $_mysql
# $_userSql
# $_passSql
# $_bd
# $_puertoSql
# $tabla
# $aux
# $llave
# $query
# $nombre
# $parametros
# $complementoUrl
# $textoJsonNew
# $res
# $fichero
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Class/Fichero.php
# /Config/DB.php
*****
<?php
/*
 * Archivo para actualizar los datos del grid
 */
session_start();
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Class/Fichero.php';
require_once '../Config/DB.php';

$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);

$tabla = $_REQUEST["tabla"];
$aux = $_REQUEST;
unset($aux["tabla"]);
unset($aux["isNewRecord"]);
$llave = "id".$tabla;
```

```
//Encriptacion para campo password
if(isset($aux["password"]) && $tabla!="servidor"){
    $pass = $sql->Consulta("password", array("<usuario>"=>$aux[$llave]));
    $pass = $pass[0]["password"];
    $aux["password"] = $pass==$aux["password"] ? $aux["password"] :
md5($aux["password"]);
}

foreach($aux as $campo => $valor){
    if(strpos($campo,"fk_") === 0 && is_numeric($valor)==false){
        $traduccion = $sql->Consulta("traduccion", array("<tabla>"=>str_replace("fk_", "",
$campo), "<llave>"=>"nombre", "<campo>"=>str_replace("fk_", "id", $campo),
"<valor>"=>$valor));
        $aux[$campo] = $traduccion[0]["valor"];
    }
}
//Acciones antes de eliminar
switch($tabla){
    case "web_service":
        $aux["sys_fk_usuario"] = $_SESSION["idUsuario"];
        $query = $sql->Consulta("consultaWebServiceIdConsulta",
array("<consulta>"=>$aux[$llave]));
        $nombre = $query[0]["nombre"];
        $parametros = explode(",", $query[0]["parametros"]);
        $query = $query[0]["query"];
        //Reemplaza parametros
        foreach($parametros as $parametro){
            $query = str_replace($parametro, str_replace(array("{","}"), array("<",">"),
$parametro), $query);
        }
        //Texto a eliminar Json
        $textoJsonOld = "\t'.'. '$nombre.'': '$query.'".PHP_EOL;
        //-----
        $query = $sql->Consulta("consultaWebService",
array("<consulta>"=>$aux["fk_consulta"]));
        $parametros = explode(",", $query[0]["parametros"]);
        $query = $query[0]["query"];
        $complementoUrl = "";
        //Reemplaza parametros
        foreach($parametros as $parametro){
            $complementoUrl .= "&" . str_replace(array("{","}"), array("", ""), $parametro) . "=" .
$parametro;
        }
    }
}
```

```

        $query = str_replace($parametro, str_replace(array("{","}"), array("<",">"),
$parametro), $query);
    }
    //Texto a escribir Json
    $textoJsonNew = "\t'.'".$aux["nombre"].'": "'.$query.'".PHP_EOL;
    $aux["url"] = "http://pnh.gob.mx/sistema_telemetria/Services/WebService.php?key="
. $aux["nombre"] . $complementoUrl;
    break;
}
//Actualizacion del registro
$res = $sql->Actualizar($tabla, $llave, $aux[$llave], $aux);
//Respuesta
if($res != false){
    //Caso especial en tablas
    switch($tabla){
        case "web_service":
            //Escribir en JSON
            $fichero = new Fichero("../JSON/ConsultasWebService.json");
            $fichero->EliminarLinea($textoJsonOld);
            $fichero->EscribirLineas($textoJsonNew, "p");
            break;
        case "publicacion_fuente_datos":
            $archivos = $sql->Consulta("frecuencias", array());
            foreach($archivos as $archivo){
                $lineas = $sql->Consulta("generaContenidoArchivo",
array("<frecuencia>"=>$archivo["frecuencia"]));
                $texto = "";
                foreach($lineas as $linea){
                    $texto .= $linea["linea"].PHP_EOL;
                }
                $fichero = new Fichero("../OrigenDatos/".$archivo["idfrecuencia"].".txt");
                $fichero->EscribirContenido($texto);
            }
            break;
    }

    foreach($aux as $campo => $valor){
        if(strpos($campo,"fk_") === 0){
            $traduccion = $sql->Consulta("traduccion", array("<tabla>"=>str_replace("fk_", "",
$campo),"<llave>"=>str_replace("fk_", "id", $campo), "<campo>"=>"nombre",
"<valor>"=>$valor));
            $aux[$campo] = $traduccion[0]["valor"];
        }
    }

```

```
    }  
    echo json_encode($aux);  
  }  
  else{  
    echo json_encode(array(  
      'isError' => true,  
      'msg' => 'El registro no se ha podido guardar.<br><br>'.$sql->getError()  
    ));  
  }?>
```

CLASS

```
*****  
# NOMBRE  
# Dump.php  
# VARIABLES  
# $_hostImportar  
# $_hostExportar  
# $_sql_hi  
# $_sql_he  
# $tablas  
# $datos  
# $queryImporta  
# $creaTablas  
# $tablas  
*****  
<?php  
/*
```


* Clase para crear e importar tablas

*/

```
class Dump {

    var $_hostImpotar;
    var $_hostExportar;
    var $_sql_hi;
    var $_sql_he;
    var $_utf8Encode;

    function Dump($_hostImportar, $_hostExportar = false, $_utf8Encode = false) {
        $this->_utf8Encode = $_utf8Encode;
        $this->_hostImpotar = $_hostImportar;
        $this->_hostExportar = $_hostExportar == false ? $_hostImportar : $_hostExportar;
        $this->_sql_hi = new SQL($this->_hostImpotar["host"], $this->_hostImpotar["user"],
$this->_hostImpotar["pass"], $this->_hostImpotar["db"], $this->_hostImpotar["puerto"]);
        $this->_sql_he = new SQL($this->_hostExportar["host"], $this->_hostExportar["user"],
$this->_hostExportar["pass"], $this->_hostExportar["db"], $this->_hostExportar["puerto"]);
    }

    public function Importar($tablas) {
        echo "Dump iniciado<br><br>";
        $this->_sql_hi->Begin();
        foreach ($tablas as $tabla => $datos) {
            $queryImporta = $datos["sql"];
            $creaTabla = isset($datos["crearTabla"]) ? $datos["crearTabla"] : true;

            $creado = $creaTabla == true ? $this->CrearTabla($tabla) : true;

            if ($creado==true) {
                if($creaTabla == true)
                    echo "Tabla ".$tabla." creada.<br>";

                $insertado = $this->ImportarDatos($tabla, $queryImporta);
                if ($insertado === false) {
                    echo "Insert Error";
                    $this->_sql_hi->RollBack();
                    exit();
                }
            }
            else{
                echo $insertado." datos insertados en tabla ".$tabla."<br>";
            }
        }
    }
}
```

```

    }
  } else {
    echo "Create Error";
    $this->_sql_hi->RollBack();
    exit();
  }
}
$this->_sql_hi->Commit();
echo "<br>Dump finalizado<br>";
}

public function CrearTabla($tabla) {
  $info = $this->_sql_he->Consulta("describeTable", array("<tabla>" => $tabla));
  $queryCreate = "CREATE TABLE IF NOT EXISTS " . $tabla . " (";
  $queryColumns = "";
  $queryPrimaryKey = "";
  $queryUniqueKey = "";

  foreach ($info as $columna) {
    $queryColumns .= $queryColumns != "" ? "," : "";
    $queryColumns .= $columna["Field"] . " " . //Nombre
      $columna["Type"] . //Tipo de dato
      ($columna["Null"] == "NO" ? " NOT NULL" : " NULL") . //Nuleable
      ($columna["Default"] != "" ? " DEFAULT " . $columna["Default"] . "" : ""); .
    //Default
    $columna["Extra"]; //Extra

    switch ($columna["Key"]) {
      case "PRI":
        $queryPrimaryKey .= ($queryPrimaryKey != "" ? "," : "") . $columna["Field"];
        break;
      case "UNI":
        $queryUniqueKey .= ($queryUniqueKey != "" ? "," : "") . $columna["Field"];
        break;
      default:
        break;
    }
  }
  $queryCreate .= $queryColumns;
  $queryCreate .= $queryPrimaryKey != "" ? ", PRIMARY KEY (" . $queryPrimaryKey . ") : "" ;
  $queryCreate .= $queryUniqueKey != "" ? ", UNIQUE (" . $queryUniqueKey . ") : "" ;
  $queryCreate .= ")";
  //echo "<br>".$queryCreate."<br>";
}

```

```
        return $this->_sql_hi->Query($queryCreate);
    }

    public function ImportarDatos($tabla, $queryImporta) {
        $return = true;
        $data = $this->_sql_he->Query($queryImporta);
        foreach ($data as $registro) {
            $res = $this->_sql_hi->Insertar($tabla, $registro, $this->_utf8Encode);
            if ($res == false) {
                $return = false;
                break;
            }
        }
        return $return == true ? sizeof($data) : $return;
    }
}

?>
```

```
# NOMBRE
# Fichero.php
# VARIABLES
# $_archivo
# $archivo
# $file
# $lineas
# $claveLinea
# $inicio
# $final
# $posicion
# MODULOS
# /Class/SQL.php
# /Class/Dump.php
# /Class/JSON.php
# /Config/DB.php
```

```
<?php
```

```
/*
```

* Clase para escribir en ficheros

*/

```
class Fichero{
    var $_archivo;

    function Fichero($archivo){
        $this->_archivo = $archivo;
    }

    function getCotenido(){
        $file = fopen($this->_archivo, 'r');
        while(!feof($file)) {
            $linea = fgets($file);
            $lineas[] = $linea;
        }
        fclose($file);
        return $lineas;
    }

    function EscribirContenido($lineas){
        file_put_contents($this->_archivo, $lineas);
    }

    function EliminarLinea($claveLinea){
        $lineas = $this->getCotenido();
        $posicion = array_search($claveLinea, $lineas);
        unset($lineas[$posicion]);
        $this->EscribirContenido($lineas);
    }

    function EliminarLineas($lineaClave){
        $lineas = $this->getCotenido();

        $inicio = array_search("//Inicio:". $lineaClave.PHP_EOL, $lineas);
        $final = array_search("//Final:". $lineaClave.PHP_EOL, $lineas);

        for($i=$inicio; $i<=$final; $i++){
            unset($lineas[$i]);
        }

        $this->EscribirContenido($lineas);
    }
}
```

```
//u = ultimo, p = penultimo
function EscribirLineas($texto, $posicion){
    $lineas = $this->getCotenido();

    if($posicion == "u"){
        $lineas[sizeof($lineas)] = $texto;
    }
    else if($posicion == "p"){
        $tam = sizeof($lineas);
        $lineas[$tam] = $lineas[$tam-1];
        $lineas[$tam-1] = $texto;
    }

    $this->EscribirContenido($lineas);
}
}
/*
function escribeWebService() {
    $archivo = "../Services/Services.php";
    $archivo = "../JSON/ConsultasWebService.json";

    if(file_exists($archivo)) {
        echo "leer";
        $file = fopen($archivo,'r');
        while(!feof($file)) {
            $name = fgets($file);
            $lineas[] = $name;
        }
        fclose($file);
        print_r($lineas);
        file_put_contents("Hola.txt", $lineas);
        // Todas las lineas quedan almacenadas en $lineas
        // Ahora eliminamos la fila 15 por ejemplo, en el array sería la posición 14 (empezamos por
        la 0)
        $lineas[sizeof($lineas)-2] = $lineas[sizeof($lineas)-2].",";
        // unset($lineas[sizeof($lineas)-1]);
        $lineas[sizeof($lineas)-1] = ""xxx":"xxxxxxx";
        $lineas[sizeof($lineas)] = '}';
        $lineas = array_values($lineas);
        print_r($lineas);
        // GUARDAMOS
        $file = fopen($archivo, "w");
```

```
foreach( $lineas as $linea ) {
    echo "escribir ". $linea."<br>";
    fwrite( $file, $linea );
}
fclose( $file );

}
else{
    echo "archivo no encontrado";
}
}

//escribeWebService();

$lineaClave = "1".PHP_EOL;
$lineas =
array("//InicioLineaClave:1".PHP_EOL,"//InicioLineaClave:11","//FinalLineaClave:1","//Final
LineaClave:");

$inicio = array_search("//InicioLineaClave:".$lineaClave, $lineas);
$final = array_search("//FinalLineaClave:".$lineaClave, $lineas);

echo $inicio."<br>";
echo $final."<br>";
echo "\n111";
*/
?>

*****
# NOMBRE
# GRID.php
# VARIABLES
# $_tabla
# $_titulo
# $_sql
# $_columnas
# $_reemplazar
# $html
# $fila
# $opciones
# $password
```

```
# $tabla
# MODULOS
# /Class/SQL.php
# /Class/Dump.php
# /Class/JSON.php
# /Config/DB.php
*****

<?php

class GRID{
    var $_tabla;
    var $_titulo;
    var $_sql;
    var $_columnas = array();
    var $_reemplazar = array(
        "buscar" => array("fk_", "_"),
        "reemplazo" => array(" ", " ")
    );

    function GRID($tabla, $titulo){
        include '../Config/DB.php';
        $_schema = $_bd;
//    $_bd = "INFORMATION_SCHEMA";
        $this->_sql = new SQL($_mysql, $_userSql, $_passSql, $_bd_IS, $_puertoSql);
        $this->_tabla = $tabla;
        $this->_titulo = $titulo;
        $this->_columnas = $this->_sql->Consulta("columnas", array("<tabla>"=>$tabla,
"<schema>"=>$_schema));
    }

    public function comboBox($tabla, $label, $itemSeleccion = NULL){
        $html = "<label><b>".$label."&nbsp;</b></label><input id='cc' class='easyui-combobox genericCobmo"
        data-
options='valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla='.$tabla.('$itemSeleccion!=NULL
?
'&item='.$itemSeleccion:')."'\,onSelect:
function(rec){selectCombo(rec.id);},
onLoadSuccess:
function(){setTimeout(selectCombo('$\#cc').comboBox('\getValue\'), 7000);}'>";
        return $html;
    }

    public function getTablaHTML(){
```

```

$html = '<table id="dg" title="'.str_replace("_", " ", strtoupper($this->_titulo)).'"
style="width:auto;height:auto"
    toolbar="#toolbar"    pagination="true"    pageSize="20"    idField="'. $this-
>_columnas[0]["column_name"].'"
    rownumbers="true" fitColumns="true" singleSelect="true"
    sortName="'. $this->_columnas[0]["column_name"].'" sortOrder="asc">
<thead>
<tr>';

foreach($this->_columnas as $fila){
    if($fila["column_key"] != "PRI"){
        $opciones = $this->getOpciones($fila);
        $password = $fila["column_name"] == "password" ? 'formatter="formatPassword"'
: "";

        $html .= '<th field="'. $fila["column_name"].'" width="50" editor="'. $opciones.'"
'. $password.' sortable="true">'.
            strtoupper(str_replace($this->_reemplazar["buscar"], $this-
>_reemplazar["reemplazo"], $fila["column_name"])).
            '</th>';
    }
}

$html .= '</tr>
</thead>
</table>';
return $html;
}

private function getOpciones($parametros){
    $opciones = "";

//    if(($parametros["column_key"] == "MUL" || $parametros["column_key"] == "UNI")
&& strpos($parametros["column_name"], "fk_")!=false && $parametros["is_nullable"] ==
"NO"){
//        $llaveForanea = $this->_sql->Consulta("llaveForanea",
array("<columna>"=>$parametros["column_name"], "<tabla>"=>$this->_tabla));
//        $opciones .= "{type:'combobox',options:{required:true,
valueField:'. $llaveForanea[0]["columna"].',textField:'nombre',url:'../AJAX/get_combobox.p
hp?tabla=.' $llaveForanea[0]["tabla"].'&pk=.' $llaveForanea[0]["columna"].'"}';
//    }
    if(strpos($parametros["column_name"], "fk_")==0){
        $temp = explode("fk_", $parametros["column_name"]);

```



```
$opciones .= "{type:'combobox',options:{required:". ($parametros["is_nullable"] ==
"NO" ? "true" : "false") .", valueField:". "id". $temp[1].", textField:'nombre',
url:'../AJAX/get_combobox.php?tabla=". $temp[1]. "&pk=". "id". $temp[1]. "}}";
}
else if($parametros["column_name"]=="marcador"){
    $opciones .= "{type:'combobox',options:{formatter:function(row){return
combolmageFormat(row);},required:". ($parametros["is_nullable"] == "NO" ? "true" : "false")
.", valueField:". "id", textField:'nombre',
url:'../AJAX/get_combobox_image.php?tabla=Markers'}}";
}
else{
    $tipo = "";
    $requerido = "";

    if($parametros["is_nullable"] == "NO"){
        $requerido = "true";
    }
    else{
        $requerido = "false";
    }
}

switch($parametros["data_type"]){
    case "double":
    case "int":
        $tipo = "numberbox";
        break;
    case "varchar":
    case "char":
    case "text":
        $tipo = $parametros["is_nullable"] == "NO" ? "validatebox" : "text";
        break;
    case "datetime":
        $tipo = "datebox";
        break;
    default:
        $tipo = "text";
        break;
}

if($parametros["column_name"]=="password"){
    //$tipo = "password";
}
```

```
        $opciones .= "{type: ".$tipo.",options:{required: ".$requerido."}"}";
    }

    return $this->_tabla != "fuente_datos" ? $opciones : "";
}
}
?>

*****
# NOMBRE
# JSON.php
# VARIABLES
# $direccion
# $contenido
# $archivo
*****
<?php
/*
 * Clase para interpretar los archivos JSON
 * @author JOSE DE JESUS ROSALES BANDERAS
 */

class JSON{
    var $_direccion;

    function JSON($direccion){
        $this->_direccion = $direccion;
    }

    public function getArchivo(){
        return file_exists($this->_direccion) ? fopen($this->_direccion, "r") : NULL;
    }

    public function getContenido(){
        $contenido = "";
        $archivo = $this->getArchivo();
        if (!is_null($archivo)) {
            while (!feof($archivo))
                $contenido .= fgets($archivo);
            fclose($archivo);
        }
    }
}
```

```
        return $contenido;
    }

    public function JSON_array(){
        $contenido = $this->getContenido();
        return json_decode($contenido, true);
    }
}
?>

*****
# NOMBRE
# Menu.php
# VARIABLES
# $_sql
# $_mysql
# $_userSql
# $_passSql
# $_bd
# $_puertoSql
# $htmlMenu
# $htmlItem
# $menus
# $paginas
# $botones
*****
<?php

class Menu{
    var $_sql;

    function Menu(){
        include '../Config/DB.php';
        $this->_sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);
    }

    public function getMenuHTML(){
        $htmlMenu = '<div id="divMenu"><div id="menuActualizable" class="easyui-panel"
style="padding:5px;">';

```

```

// $htmlMenu .= '<a href="#" id="btn_logout" class="easyui-linkbutton" style="float:
right;">Cerrar Sesión</a>';
$htmlItem = "";
$htmlMenu .= '<a href="#" class="easyui-linkbutton"
onClick="window.location.reload();" data-options="iconCls:\'icon-reload\', plain:true"></a>
|';
$menus = $this->_sql->Consulta("menus",
array("<usuario>"=>$_SESSION["idUsuario"]));
foreach($menus as $menu){
$htmlMenu .= '<a href="#" class="easyui-menubutton" data-
options="menu:\'#"#menu".$menu["idmenu"]."\'"\'>\'$menu["menu"].\'</a>';

$paginas = $this->_sql->Consulta("paginas", array("<menu>"=>$menu["idmenu"],
"<usuario>"=>$_SESSION["idUsuario"]));

$htmlItem .= '<div id="menu\'$menu["idmenu"]."\' style="width:auto;">';
foreach($paginas as $pagina){
$htmlItem .= '<div><a href="\'$pagina["url"]."\'
class="MenuUI">\'$pagina["pagina"].\'</a></div>';
}
$htmlItem .= "</div>";
}

$htmlMenu .= '</div></div>';
$html = $htmlMenu . $htmlItem;

return $html;
}

public function getBotones(){
$html = '<div class="panel" style="padding:0px 0px 5px 0px;">';
$botones = $this->_sql->Consulta("botones", array());
foreach($botones as $boton){
$html .= '<a href="\'$boton["url"]."\' class="easyui-linkbutton"
style="width:100px;">\'$boton["boton"].\'</a>';
}
$html .= '</div>';
return $html;
}
}
?>

```

```
# NOMBRE
# SMS.php
# VARIABLES
# $_opciones
# $_URL
# $filename
```

```
<?php
```

```
class SMS {
```

```
    private $_opciones;
```

```
    private $_URL;
```

```
    public function __construct() {
```

```
        $json = new JSON("../JSON/SMS.json");
```

```
        $this->_opciones = $json->JSON_array();
```

```
        $this->construyeURL();
```

```
    }
```

```
    private function construyeURL() {
```

```
        $this->_URL = $this->_opciones["URL"];
```

```
        $this->_URL = str_replace("<respuesta>", $this->_opciones["respuesta"], $this->_URL);
```

```
        $this->_URL = str_replace("<key>", $this->_opciones["key"], $this->_URL);
```

```
        $this->_URL = str_replace("<secret>", $this->_opciones["secret"], $this->_URL);
```

```
        $this->_URL = str_replace("<from>", $this->_opciones["from"], $this->_URL);
```

```
        $this->_URL = str_replace("<codigoLocal>", $this->_opciones["codigoLocal"], $this->_URL);
```

```
    }
```

```
    public function enviaSMS($celular, $mensaje, $codigoCompania) {
```

```
        $filename = $this->_URL;
```

```
        $filename = str_replace("<destinatario>", $celular, $filename);
```

```
        $filename = str_replace("<texto>", urlencode($mensaje), $filename);
```

```
        $filename = str_replace("<CodigoRedCelular>", $codigoCompania, $filename);
```

```
        if ($this->_opciones["enabled"] == 1) {
```

```
            return file_get_contents($filename);
```

```
        } else
```

```
            return false;
```

```
    }
```

```
}
```

?>

```
# NOMBRE
# SQL.php
# VARIABLES
# $_server
# $_puerto
# $_user
# $_pass
# $_DB
# $_link
# $_consultas
# $server
# $puerto
# $user
# $pass
# $caso
```

```
<?php
/*
 * Clase para la conexion a la BD
 */
```

```
class SQL{
    //Datos de conexion
    private $_server = "";
    private $_puerto = "";
    private $_user = "";
    private $_pass = "";
    private $_DB = "";
    //Conexion
    private $_link = null;
    //Array con las consultas
    private $_consultas = array();

    //Constructor
    function SQL($server, $user, $pass, $bd, $puerto = null, $json = null){
        $json = new JSON($json==null ? "../JSON/Consultas.json" : $json);
        $this->_consultas = $json->JSON_array();
    }
}
```

```
$this->_server = $server;
$this->_puerto = $puerto;
$this->_user = $user;
$this->_pass = $pass;
$this->_DB = $bd;
$this->_link = mysql_connect($this->_server . ($this->_puerto!=null ? ":".$this->_puerto : ""), $this->_user, $this->_pass);
echo mysql_error($this->_link);
mysql_select_db($this->_DB, $this->_link);
}

public function getError(){
    return mysql_error($this->_link);
}

//Establecer o cambiar base de datos a usar
public function UsaBD($bd){
    $this->_DB = $bd;
    mysql_select_db($this->_DB, $this->_link);
}

//Consulta un caso json
public function Consulta($caso, $variables = array(), $soloJSON = false){
    $query = "";
    if(in_array($caso, array_keys($this->_consultas))){
        $query = $this->_consultas[$caso];
    }
    else{
        if($soloJSON==true)
            return -1;
        else
            $query = $caso;
    }
}

$query = $this->PreparaConsulta($query, $variables);
//return query;
//echo $query."<hr>";
return $this->Query($query);
}

//Preparar consulta con las variables
private function PreparaConsulta($query, $variables){
    foreach($variables as $llave => $valor)
```

```
$query = str_replace($llave, $valor, $query);

return $query;
}

//Consulta SQL
public function Query($sql){
    $salida = array();
    $res = mysql_query($sql, $this->_link);

    if(is_bool($res)){
        $salida = $res;
    }
    else{
        while($row = mysql_fetch_array($res, MYSQL_ASSOC))
            $salida[] = $row;
    }

    return $salida;
}

//Insertar
public function Insertar($tabla, $datos, $utf8Encode = false){
    $replaces = array(
        "busca" => array("`", ""),
        "reemplaza" => array("\`", "\'")
    );
    $return = false;
    $campos = "";
    $valores = "";
    foreach($datos as $llave => $dato){
        $campos .= ($campos!="?" ":"").$llave;
        $dato = str_replace($replaces["busca"], $replaces["reemplaza"], $dato);
        $valores .= ($valores!="?" ":"")."". ($utf8Encode ? utf8_encode($dato) : $dato) ."";
    }
    $query = "INSERT INTO ".$tabla." (".$campos.") SELECT ".$valores."";
    $res = mysql_query($query, $this->_link);
    //echo $query."<br>";//.($res==false? " -> ERROR" : " -> OK")."<br>";
    $id = mysql_insert_id($this->_link);

    if($res == false){
        //echo $query."<br>";
        $return = false;
    }
}
```



```
}
else{
    if($id != false)
        $return = $id;
    else
        $return = true;
}

return $return;
}

//Actualizar
public function Actualizar($tabla, $pk, $valor, $datos){
    $campos = "";
    foreach($datos as $llave => $dato){
        $campos .= ($campos == "" ? "" : ",") . $llave . "=" . $dato . "" ;
    }
    $query = "UPDATE ".$tabla." SET ".$campos." WHERE ".$pk."=".$valor;
    //echo $query;
    $res = mysql_query($query, $this->_link);
    return $res;
}

//Eliminar
public function Eliminar($tabla, $pk, $valor){
    $query = "DELETE FROM ".$tabla." WHERE ".$pk."=".$valor;
    $res = mysql_query($query, $this->_link);
    return $res;
}

//Transaccion inicio
public function Begin(){
    mysql_query("BEGIN", $this->_link);
}

//Transaccion aceptada
public function Commit(){
    mysql_query("COMMIT", $this->_link);
}

//Transaccion cancelada
public function RollBack(){
    mysql_query("ROLLBACK", $this->_link);
}
```

```
    }
}
?>
CONFIG
*****
# NOMBRE
# DB_estaciones.php
*****
<?php
/*
 * Archivo de configuración para la conexión a la Base de Datos
 */

$conexionEstaciones = array(
    "host" => "sig.imta.mx",
    "user" => "ccc",
    "pass" => "",
    "puerto" => "3306",
    "db" => "sih"
);
?>

*****
# NOMBRE
# GUI.php
# MODULOS
# Mapa.php
*****
<?php
/*
 * Archivo para la configuracion de paginas
 */

$_paginaDefault = "Mapa.php?tabla=sistema_telemetria";
?>

*****
# NOMBRE
# DB.php
```

```
# VARIABLES
# $_mysql
# $_bd
# $_bd_IS
# $_puertoSql
# $_userSql
# $_passSql
*****

<?php
/*
 * Archivo de configuración para la conexión a la Base de Datos
 */

$_mysql = "localhost";
$_bd = "sistema_telemetria";
$_bd_IS = "INFORMATION_SCHEMA";
$_puertoSql = "3306";
$_userSql = "root";
$_passSql = "";
//$_passSql = "*****";

$conexionLocal = array(
    "host" => $_mysql,
    "user" => $_userSql,
    "pass" => $_passSql,
    "puerto" => $_puertoSql,
    "db" => $_bd
);
?>

*****

# NOMBRE
# Estilos.css
*****

/*
Archivo CSS con los estilos de las paginas
*/
.item-text {
    vertical-align: super;
    padding-bottom: 5px;
}
}
```

```
.MenuUI {  
  text-decoration: none;  
  color: #666666;  
}  
.MenuUI:hover {  
  color: #fdfdfd;  
}  
body{  
  background: #fbfbfb;  
  padding: 10px;  
  margin: 10px;  
}  
#cc{  
  width:350px;  
  height:250px;  
  margin-top: 70px;  
}  
.cc{  
  width:350px;  
  height:250px;  
}  
#content{  
  padding:5px;  
  background:#f5f5f5;  
}  
#title-header{  
  height: auto;  
  padding: 20px;  
}  
#title-header h2{  
  margin: 0px;  
  color: #003399;  
}  
h3{  
  margin: 10px;  
  margin-right: 3px;  
}  
.panelContenido{  
  margin-top: 0px;  
  margin-bottom: 30px;  
  height: 35px;  
  border-width: thin;  
  border-color: #dedede;
```

```
}  
#btn_logout{  
  float: right;  
  margin-top: -6px;  
}  
#btn_finalizaSensores{  
  float: right;  
}  
#horaActual{  
  float: right;  
  margin-right: -15px;  
  color: #1a7bc9;  
}  
#divDetail{  
  height:244px;  
  overflow-y:scroll;  
}  
#headDetail{  
  background: #daeef5;  
  width: 100%;  
}  
#bodyDetail{  
  border-bottom-color: #9cc8f7;  
  border-bottom-width: thin;  
  border-bottom-style: solid;  
  background: #fafafa;  
  width: 100%;  
}  
#headDetail td{  
  border: 0px;  
}  
#map_canvas {  
  height: 100%;  
  margin: 0px;  
  padding: 0px  
}  
.panelMapa{  
  width: auto;  
  /*height: 550px;*/  
}  
.infoBoxMap{  
  width: 100%;  
  text-align: justify;
```

```
}  
.infoBoxMap tr:first-child{  
  text-align: center;  
  font-weight: bold;  
}  
.infoBoxMap tr:last-child{  
  text-align: center;  
}  
.infoBoxMap th{  
  text-align: justify;  
}  
.infoBoxMap img{  
  margin: 2px;  
  width: 130px;  
}
```

GUI

```
# NOMBRE
# FuenteDatos.php
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Class/Menu.php
# /Class/GRID.php
# /Config/GUI.php
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<?php
```

```
session_start();
```

```
require_once '../Class/SQL.php';
```

```
require_once '../Class/JSON.php';
```

```
require_once '../Class/Menu.php';
```

```
require_once '../Class/GRID.php';
```

```
require_once '../Config/GUI.php';
```

```
if (!isset($_SESSION["user"]))
```

```
    header("Location: Login.php");
```

```
$tabla = "fuente_datos_sensor"; //isset($_REQUEST["tabla"]) ? $_REQUEST["tabla"] : "";
```

```
$filtro = isset($_REQUEST["filtro"]) ? " where ".$_REQUEST["filtro"] : "";
```

```
$detalle = isset($_REQUEST["detalle"]) ? $_REQUEST["detalle"] : 0;
```

```
$tituloGrid = "Relacion Sensor-Llave" //isset($_REQUEST["tituloGrid"]) ?
```

```
$_REQUEST["tituloGrid"] : $tabla;
```

```
?>
```

```
<html>
```

```
    <head>
```

```
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
        <title>Sistema Telemetría</title>
```

```
        <!--<link rel="stylesheet"
```

```
type="text/css"
```

```
href="http://www.jeasyui.com/easyui/themes/default/easyui.css">-->
```

```
<link rel="stylesheet" type="text/css" href="../../CSS/easyui.css">
<link rel="stylesheet" type="text/css" href="../../CSS/icon.css">
<link rel="stylesheet" type="text/css" href="../../CSS/demo.css">
<link rel="stylesheet" type="text/css" href="../../CSS/Estilos.css">
<script type="text/javascript" src="../../JS/jquery-1.6.min.js"></script>
<script type="text/javascript" src="../../JS/jquery.easyui.min.js"></script>
<script type="text/javascript" src="../../JS/jquery.edatagrid.js"></script>
<script type="text/javascript" src="../../JS/datagrid-detailview.js"></script>
<script type="text/javascript" src="../../JS/Scripts.js"></script>
<script>
    var tabla = "<?php echo $tabla; ?>";
    var filtro = "<?php echo $filtro; ?>";
    var detalle = "<?php echo $detalle; ?>";
    var paginaDefault = "<?php echo $_paginaDefault; ?>";
</script>
</head>
<body>
    <div class='panel'>
        <div class="datagrid">
            <div class="panel-header">
                <div class="panel-title" id='title-header'>
                    <h2>
                        Sistema Base de Gestión de Información de Telemetría
                    </h2>
                    <div id="horaActual"></div>
                </div>
            </div>
        </div>
    </div>
    <div class="datagrid-toolbar panelContenido" id='userInfo'>
        <h3>
            Usuario: <i><?php echo $_SESSION["user"]; ?></i>
            <a href="#" id="btn_logout" class="easyui-linkbutton">Cerrar Sesión</a>
        </h3>
    </div>
    <?php
    $menu = new Menu();
    echo $menu->getBotones();
    echo $menu->getMenuHTML();
    ?>
    <div class="datagrid-toolbar panelContenido panelMapa" style="height: auto;">
        <br>
        <form id="ff" method="post">
```



```

<table style="padding: 6px; border: solid #8d8d8d thin" align="center" border="0">
  <tr>
    <td colspan="3">
      <label id="mensaje" style="display: none; float: right;">Cargando
información...&nbsp;&nbsp;&nbsp;</label>
    </td>
  </tr>
  <tr>
    <td colspan="2">
      Origen de Datos
      <hr>
    </td>
  </tr>
  <tr>
    <td>
      <label><b>Nombre:&nbsp;&nbsp;</b></label>
    </td><td>
      <input id="nombre" data-options="required:true" class="easyui-validatebox"
style="width: 250px;">
    </td>
  </tr>
  <tr>
    <td>
      <label><b>Servidor:&nbsp;&nbsp;</b></label>
    </td><td>
      <input id="fk_servidor" class="easyui-combobox cc"
      data-
options="required:true,valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla=serv
idor',onSelect: function(rec){selectComboBox(this, rec.id);}, onLoadSuccess: function(){}">
    </td>
  </tr>
  <tr>
    <td>
      <label><b>Base de datos:&nbsp;&nbsp;</b></label>
    </td><td>
      <input id="bd" class="easyui-combobox cc"
      data-options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this, rec.id);}, onLoadSuccess:
function(){setTimeout(selectComboBox(this,$('#bd').combobox('getValue')), 5000);}">
    </td>
  </tr>
  <tr>
    <td>
  
```

```

        <label><b>Tabla:&nbsp;</b></label>
    </td><td>
        <input id="tabla" class="easyui-combobox cc"
            data-options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){setTimeout(selectComboBox(this,$('#tabla').combobox('getValue')), 5000);}">
    </td>
</tr>
<tr>
<td>
    <label><b>Llave (Se relacionará con el sensor):&nbsp;</b></label>
</td><td>
    <input id="campo_llave" class="easyui-combobox cc"
        data-options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){setTimeout(selectComboBox(this,$('#campo_llave').combobox('getValue')),
5000);}">
    </td>
</tr>
<tr>
<td>
    <label><b>Fecha-Hora (Fecha y hora del valor medido):&nbsp;</b></label>
</td><td>
    <input id="campo_fecha" class="easyui-combobox cc"
        data-options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this, rec.id);}, onLoadSuccess: function(){}">
    </td>
</tr>
<tr>
<td>
    <label><b>Valor (Valor medido por el sensor):&nbsp;</b></label>
</td><td>
    <input id="campo_valor" class="easyui-combobox cc"
        data-options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){$('#mensaje').hide();}">
    </td>
</tr>
<!--
    <tr>
    <td colspan="2">
    <br><br>Relación Llave-Sensor
    <br>
    </td>

```

```

</tr-->
<!--
  <tr>
    <td>
      <label><b id="llaves_label">Llaves:&nbsp;</b></label>
    </td><td>
      <input id="llaves" class="easyui-combobox cc"
        data-options="valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){$('#mensaje').hide()}">
    </td>
  </tr>
  <tr>
    <td>
      <label><b>Estaciones:&nbsp;</b></label>
    </td><td>
      <input id="estacion" class="easyui-combobox cc"
        data-
options="valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla=estacion_fuente',
onSelect:      function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){setTimeout(selectComboBox(this,$('#estacion').combobox('getValue')),
5000);}">
    </td>
  </tr>
  <tr>
    <td>
      <label><b>Sensores:&nbsp;</b></label>
    </td><td>
      <input id="sensor" class="easyui-combobox cc"
        data-options="valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){$('#mensaje').hide()}">
    </td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <a href="#" id="btn_SensorLlave" class="easyui-linkbutton">Agregar
relación</a>
    </td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <a href="#" id="btn_SensorLlave" class="easyui-linkbutton">Finalizar</a>
    </td>

```

```

</tr-->
<tr>
  <td colspan="2" align="center">
    <br>
    <a href="#" id="btn_finalizaFuente" class="easyui-linkbutton">Guardar y
continuar</a>
    <br>
  </td>
</tr>
<tr>
  <td colspan="2" align="center">
    <script>
//  function submitForm(){
//    $('#ff').form('submit');
//  }
//  function clearForm(){
//    $('#ff').form('clear');
//  }
    </script>
  </td>
</tr>
<!-- <tr>
  <td colspan="3">
    <select id="dept" class="easyui-combobox" name="dept" style="width:200px;"
onchange="cambio()" data-options="
      onSelect: function(rec){
        alert(rec.value);
        var url = 'get_data2.php?id='+rec.id;
        $('#cc2').combobox('reload', url);
      }">
    <option value="aa">aitem1</option>
    <option>bitem2</option>
    <option>bitem3</option>
    <option>ditem4</option>
    <option>eitem5</option>
    </select>
  </td>
</tr-->
</table>
</form>
<br>
<?php
//  $grid = new GRID($tabla, $tituloGrid);

```

```
// echo $grid->getTablaHTML();
?>
</div>
</body>
</html>
```

```
*****
```

```
# NOMBRE
# FuenteDatos_Sensor.php
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Class/Menu.php
# /Class/GRID.php
# /Config/GUI.php
```

```
*****
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<?php
```

```
session_start();
```

```
require_once '../Class/SQL.php';
```

```
require_once '../Class/JSON.php';
```

```
require_once '../Class/Menu.php';
```

```
require_once '../Class/GRID.php';
```

```
require_once '../Config/GUI.php';
```

```
if (!isset($_SESSION["user"]))
```

```
    header("Location: Login.php");
```

```
$tabla = "fuente_datos_sensor"; //isset($_REQUEST["tabla"]) ? $_REQUEST["tabla"] : "";
```

```
$filtro = " where fk_fuente_datos=".$_REQUEST["fk_fuente_datos"];
```

```
$fuente = $_REQUEST["fk_fuente_datos"];
```

```
$detalle = isset($_REQUEST["detalle"]) ? $_REQUEST["detalle"] : 0;
```

```
$tituloGrid = (isset($_REQUEST["tituloGrid"]) ? $_REQUEST["tituloGrid"] : $tabla) . ":
```

```
Relacion Sensor-Llave";
```

```
?>
```

```
<html>
```

```
    <head>
```

```
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
        <title>Sistema Telemetría</title>
```

```
<!--<link rel="stylesheet" type="text/css" href="http://www.jeasyui.$_REQUEST["fuente"]com/easyui/themes/default/easyui.css">-->
<link rel="stylesheet" type="text/css" href="../CSS/easyui.css">
<link rel="stylesheet" type="text/css" href="../CSS/icon.css">
<link rel="stylesheet" type="text/css" href="../CSS/demo.css">
<link rel="stylesheet" type="text/css" href="../CSS/Estilos.css">
<script type="text/javascript" src="../JS/jquery-1.6.min.js"></script>
<script type="text/javascript" src="../JS/jquery.easyui.min.js"></script>
<script type="text/javascript" src="../JS/jquery.edatagrid.js"></script>
<script type="text/javascript" src="../JS/datagrid-detailview.js"></script>
<script type="text/javascript" src="../JS/Scripts.js"></script>
<script>
    var tabla = "<?php echo $tabla."&fk_fuente_datos=".$fuente; ?>";
    var filtro = "<?php echo $filtro; ?>";
    var detalle = "<?php echo $detalle; ?>";
    var paginaDefault = "<?php echo $_paginaDefault; ?>";
    var fuente = "<?php echo $fuente; ?>";
</script>
</head>
<body>
    <div class='panel'>
        <div class="datagrid">
            <div class="panel-header">
                <div class="panel-title" id='title-header'>
                    <h2>
                        Sistema Base de Gestión de Información de Telemetría
                    </h2>
                    <div id="horaActual"></div>
                </div>
            </div>
        </div>
    </div>
    <div class="datagrid-toolbar panelContenido" id='userInfo'>
        <h3>
            Usuario: <i><?php echo $_SESSION["user"]; ?></i>
            <a href="#" id="btn_logout" class="easyui-linkbutton">Cerrar Sesión</a>
        </h3>
    </div>
    <div style="padding-right:15px;">
        <?php
        $menu = new Menu();
        echo $menu->getBotones();
        </?php
    </div>
</body>
</html>
```

```

echo $menu->getMenuHTML();
?>
<!-- <div class="datagrid-toolbar panelContenido panelMapa" style="height: auto;">-->
    <br><a href="#" id="btn_finalizaSensores" class="easyui-linkbutton">Publicar
fuente</a>
    <br><br>
    <table id="dg" title="<?php echo $tituloGrid;?>" style="width:auto;height:auto"
        toolbar="#toolbar" pagination="true" pageSize="20"
idField="idfuelle_datos_sensor"
        rownumbers="true" fitColumns="true" singleSelect="true"
        sortName="idfuelle_datos_sensor" sortOrder="asc">
    <thead>
        <tr>
            <th field="fk_sensor" width="50" sortable='true'
                editor="{type:'combobox',options:{required:true,
valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla=sensor_fuente'}}">
                SENSOR</th>
            <th field="llave" width="50" sortable='true'
                editor="{type:'combobox',options:{required:true,
valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla=llaves&fuente=<?php echo
$fuelle;?>'}}">
                LLAVE</th>
            <th
                field="fk_estatus" width="50"
                editor="{type:'combobox',options:{required:true, valueField:'idestatus', textField:'nombre',
url:'../AJAX/get_combobox.php?tabla=estatus&pk=idestatus'}}" sortable="true">
                ESTATUS</th>
        </tr>
    </thead>
    </table>
    <div id="toolbar">
        <a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true"
onclick="javascript:$('#dg').datagrid('addRow')">Nuevo</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-remove" plain="true"
onclick="javascript:$('#dg').datagrid('destroyRow')">Eliminar</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-save" plain="true"
onclick="javascript:$('#dg').datagrid('saveRow')">Guardar</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-undo" plain="true"
onclick="javascript:$('#dg').datagrid('cancelRow')">Cancelar</a>
    </div>
<!-- </div>-->
</body>
</html>

```

```
# NOMBRE
# GraficaMin.php
# VARIABLES
# $sql
# $_mysql
# $userSql
# $_passSql
# $_bd
# $_puertoSql
# $grupo
# $usuario
# $fila
# $serie
# $res
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Config/DB.php
```

```
<?php
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Config/DB.php';

//Conexion a la BD
$sql = new SQL($_mysql, $_userSql, $_passSql, $_bd, $_puertoSql);

$grupo = $_REQUEST["grupo"];
$usuario = $_REQUEST["usuario"];

$res = $sql->Consulta("primerGrafica", array("<grupo>" => $grupo, "<usuario>" =>
$usuario));

$serie = "";
if (sizeof($res) > 0) {
    $fila = $res[0];
    $serie = $sql->Consulta("serieVistaGrafica", array("<sensor>" => $fila["idsensor"],
"<intervalo>" => $fila["rango_horas"]));

    if (sizeof($serie) > 0)
```



```
$serie = $serie[0]["valor"];  
}  
else{  
  $fila["fondo_grid"] = "#000000";  
}  
?>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
<head>
```

```
<script type="text/javascript" src="../JS/jquery-1.6.min.js"></script>  
<script type="text/javascript" src="../JS/jquery.sparkline.js"></script>
```

```
<script type="text/javascript">  
  $(function() {  
    /* This code runs when everything has been loaded on the page */  
    /* Inline sparklines take their values from the contents of the tag */  
    $('inlinesparkline').sparkline();  
  
    /* Sparklines can also take their values from the first argument  
    passed to the sparkline() function */  
    var myvalues = [10, 8, 5, 7, 4, 4, 1];  
    $('dynamicsparkline').sparkline(myvalues);  
  
    /* The second argument gives options such as chart type */  
    $('dynamicbar').sparkline(myvalues, {type: 'bar', barColor: 'green'});  
  
    /* Use 'html' instead of an array of values to pass options  
    to a sparkline with data in the tag */  
    $('inlinebar').sparkline('html', {type: 'bar', barColor: 'red'});  
  
    $('#Lineas').sparkline([<?php echo $serie; ?>], {  
      type: 'line',  
      spotColor: null,  
      minSpotColor: null, width: '100%', height: '100%',  
      maxSpotColor: null,  
      highlightSpotColor: null,  
      highlightLineColor: null,  
      fillColor: null, lineWidth: 10,  
      lineColor: '#2C908E', disableTooltips: true,  
      drawNormalOnTop: false});
```

```

    $("#Areas").sparkline([<?php echo $serie; ?>], {
        type: 'line',
        spotColor: null,
        minSpotColor: null, width: '100%', height: '100%',
        maxSpotColor: null,
        highlightSpotColor: null,
        highlightLineColor: null, lineWidth: 10,
        fillColor: '#109898',
        lineColor: '#066060', disableTooltips: true,
        drawNormalOnTop: false});

    $("#Barras").sparkline([<?php echo $serie; ?>], {disableTooltips: true,
        zeroAxis: false, height: $(window).height(), barColor: '#9FF781',
        barWidth: Math.floor($(window).width() / <?php echo sizeof(explode(", ", $serie))
?>) - 12,
        barSpacing: 12,
        type: 'bar'});
    });
</script>
</head>
<body style="background: #1C1C1C; padding: 0px; margin: 0px;">
    <span id="Lineas"></span>
</body>
</html>
*****
# NOMBRE
# Grid.php
# VARIABLES
# $tabla
# $filtro
# $detalle
# $tituloGrid
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Class/Menu.php
# /Class/GRID.php
# /Config/GUI.php
*****
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<?php
session_start();

```

```
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Class/Menu.php';
require_once '../Class/GRID.php';
require_once '../Config/GUI.php';

if (!isset($_SESSION["user"]))
    header("Location: Login.php");

$tabla = isset($_REQUEST["tabla"]) ? $_REQUEST["tabla"] : "";
$filtero = isset($_REQUEST["filtro"]) ? " where ".$_REQUEST["filtro"] : "";
$detalle = isset($_REQUEST["detalle"]) ? $_REQUEST["detalle"] : 0;
$tituloGrid = isset($_REQUEST["tituloGrid"]) ? $_REQUEST["tituloGrid"] : $tabla;
?>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Sistema Telemetría</title>
    <!--<link rel="stylesheet" type="text/css"
href="http://www.jeasyui.com/easyui/themes/default/easyui.css">-->
    <link rel="stylesheet" type="text/css" href="../CSS/easyui.css">
    <link rel="stylesheet" type="text/css" href="../CSS/icon.css">
    <link rel="stylesheet" type="text/css" href="../CSS/demo.css">
    <link rel="stylesheet" type="text/css" href="../CSS/Estilos.css">
    <script type="text/javascript" src="../JS/jquery-1.6.min.js"></script>
    <script type="text/javascript" src="../JS/jquery.easyui.min.js"></script>
    <script type="text/javascript" src="../JS/jquery.edatagrid.js"></script>
    <script type="text/javascript" src="../JS/datagrid-detailview.js"></script>
    <script type="text/javascript" src="../JS/Scripts.js"></script>
    <script>
      var tabla = "<?php echo $tabla; ?>";
      var filtro = "<?php echo $filtro; ?>";
      var detalle = "<?php echo $detalle; ?>";
      var paginaDefault = "<?php echo $_paginaDefault; ?>";
    </script>
  </head>
  <body>
    <div class='panel'>
      <div class="datagrid">
        <div class="panel-header">
          <div class="panel-title" id='title-header'>
            <h2>
              Sistema Base de Gestión de Información de Telemetría
```

```

        </h2>
        <div id="horaActual"></div>
    </div>
</div>
</div>
</div>
<div class="datagrid-toolbar panelContenido" id='userInfo'>
    <h3>
        Usuario: <i><?php echo $_SESSION["user"]; ?></i>
        <a href="#" id="btn_logout" class="easyui-linkbutton">Cerrar Sesión</a>
    </h3>
</div>
<div style="padding-right:15px;">
<?php
$menu = new Menu();
echo $menu->getBotones();
echo $menu->getMenuHTML();
?>
<?php
$grid = new GRID($tabla, $tituloGrid);
echo $grid->getTablaHTML();

if($detalle!=1){
?>
<div id="toolbar">
    <?php if($tabla!="fuente_datos"){ ?>
        <a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true"
onclick="javascript:$('#dg').edatagrid('addRow')">Nuevo</a>
    <?php } ?>
        <a href="#" class="easyui-linkbutton" iconCls="icon-remove" plain="true"
onclick="javascript:$('#dg').edatagrid('destroyRow')">Eliminar</a>
    <?php if($tabla!="fuente_datos"){ ?>
        <a href="#" class="easyui-linkbutton" iconCls="icon-save" plain="true"
onclick="javascript:$('#dg').edatagrid('saveRow')">Guardar</a>
    <?php } ?>
        <a href="#" class="easyui-linkbutton" iconCls="icon-undo" plain="true"
onclick="javascript:$('#dg').edatagrid('cancelRow')">Cancelar</a>
    </div>
<?php
}
?>
</div>
</body>

```

</html>

```
# NOMBRE
# GridFiltro.php
# VARIABLES
# $tabla
# $filtro
# $detalle
# $tituloGrid
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Class/Menu.php
# /Class/GRID.php
# /Config/GUI.php
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<?php
session_start();
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Class/Menu.php';
require_once '../Class/GRID.php';
require_once '../Config/GUI.php';

if (!isset($_SESSION["user"]))
    header("Location: Login.php");

$tabla = isset($_REQUEST["tabla"]) ? $_REQUEST["tabla"] : "rango_marcador";
$filtro = isset($_REQUEST["filtro"]) ? " where " . $_REQUEST["filtro"] : "";
$detalle = isset($_REQUEST["detalle"]) ? $_REQUEST["detalle"] : 0;
$tituloGrid = isset($_REQUEST["tituloGrid"]) ? $_REQUEST["tituloGrid"] : $tabla;
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Sistema Telemetría</title>
```

```
<!--<link rel="stylesheet" type="text/css"
href="http://www.jeasyui.$_REQUEST["fuente"]com/easyui/themes/default/easyui.css">
-->
<link rel="stylesheet" type="text/css" href="../../CSS/easyui.css">
<link rel="stylesheet" type="text/css" href="../../CSS/icon.css">
<link rel="stylesheet" type="text/css" href="../../CSS/demo.css">
<link rel="stylesheet" type="text/css" href="../../CSS/Estilos.css">
<script type="text/javascript" src="../../JS/jquery-1.6.min.js"></script>
<script type="text/javascript" src="../../JS/jquery.easyui.min.js"></script>
<script type="text/javascript" src="../../JS/jquery.edatagrid.js"></script>
<script type="text/javascript" src="../../JS/datagrid-detailview.js"></script>
<script type="text/javascript" src="../../JS/Scripts.js"></script>
<script>
var tabla = "<?php echo $tabla; ?>";
var filtro = "<?php echo $filtro; ?>";
var detalle = "<?php echo $detalle; ?>";
var paginaDefault = "<?php echo $_paginaDefault; ?>";
var fuente = "<?php echo $fuente; ?>";
</script>
</head>
<body>
<div class='panel'>
<div class="datagrid">
<div class="panel-header">
<div class="panel-title" id='title-header'>
<h2>
Sistema Base de Gestión de Información de Telemetría
</h2>
<div id="horaActual"></div>
</div>
</div>
</div>
<div class="datagrid-toolbar panelContenido" id='userInfo'>
<h3>
Usuario: <i><?php echo $_SESSION["user"]; ?></i>
<a href="#" id="btn_logout" class="easyui-linkbutton">Cerrar Sesión</a>
</h3>
</div>
<div style="padding-right:15px;">
<?php
$menu = new Menu();
echo $menu->getBotones();
```

```

echo $menu->getMenuHTML();
?>
<br><label>Sistema de telemetría:&nbsp;</label>
<input id="marcador_sistema" name="marcador_sistema" class="easyui-combobox
cc"
    data-
options="required:false,valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla=mar
cador_sistema',onSelect: function(rec){selectComboBox(this, rec.id);}, onLoadSuccess:
function(){selectComboBox(this,$('#marcador_sistema').combobox('getValue'))}">
<br><br>
<table id="dg" title="RANGO MARCADOR" style="width:auto; height:auto"
    toolbar="#toolbar" pagination="true" pageSize="20" idField="idrango_marcador"
    rownumbers="true" fitColumns="true" singleSelect="true"
    sortName="idrango_marcador" sortOrder="asc" autoRowHeight="true">
<thead>
<tr>
<th
        field="minimo"
        width="50"
editor="{type:'numberbox',options:{required:true}}" sortable="true">MINIMO</th>
<th
        field="maximo"
        width="50"
editor="{type:'numberbox',options:{required:true}}" sortable="true">MAXIMO</th>
<th
        field="marcador"
        width="50"
        formatter="formatImage"
editor="{type:'combobox',options:{formatter:function(row){return
combolmageFormat(row);},required:true,
        valueField:'id',
        textField:'nombre',
url:'../AJAX/get_combobox_image.php?tabla=Markers'}}"
sortable="true">MARCADOR</th>
<!--<th
        field="fk_marcador_sistema"
        width="50"
editor="{type:'combobox',options:{required:true,
        valueField:'idmarcador_sistema',
textField:'nombre',
url:'../AJAX/get_combobox.php?tabla=marcador_sistema&pk=idmarcador_sistema'}}"
sortable="true"> MARCADOR SISTEMA</th-->
</tr>
</thead>
</table>

<div id="toolbar">
<a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true"
onclick="javascript:$('#dg').edatagrid('addRow')">Nuevo</a>
<a href="#" class="easyui-linkbutton" iconCls="icon-remove" plain="true"
onclick="javascript:$('#dg').edatagrid('destroyRow')">Eliminar</a>
<a href="#" class="easyui-linkbutton" iconCls="icon-save" plain="true"
onclick="javascript:$('#dg').edatagrid('saveRow')">Guardar</a>

```

```
        <a href="#" class="easyui-linkbutton" iconCls="icon-undo" plain="true"
onclick="javascript:$('#dg').edatagrid('cancelRow')">Cancelar</a>
    </div>
    <!-- </div>-->
</body>
</html>
```

```
# NOMBRE
# Login.php
# MODULOS
# /Config/GUI.php
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<?php
session_start();
```

```
require_once '../Config/GUI.php';
```

```
if (isset($_SESSION["user"]))
    header("Location: " . $_paginaDefault);
?>
```

```
<html>
```

```
    <head>
```

```
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
        <title>Sistema Telemetría</title>
```

```
        <!--<link rel="stylesheet" type="text/css"
```

```
href="http://www.jeasyui.com/easyui/themes/default/easyui.css">-->
```

```
        <link rel="stylesheet" type="text/css" href="../CSS/easyui.css">
```

```
        <link rel="stylesheet" type="text/css" href="../CSS/icon.css">
```

```
        <link rel="stylesheet" type="text/css" href="../CSS/demo.css">
```

```
        <link rel="stylesheet" type="text/css" href="../CSS/Estilos.css">
```

```
        <script type="text/javascript" src="../JS/jquery-1.6.min.js"></script>
```

```
        <script type="text/javascript" src="../JS/jquery.easyui.min.js"></script>
```

```
        <script type="text/javascript" src="../JS/jquery.edatagrid.js"></script>
```

```
        <script type="text/javascript" src="../JS/Scripts.js"></script>
```

```
        <script>
```

```
            var tabla = "";
```

```
            var filtro = "";
```

```
            var detalle = "";
```



```
var paginaDefault = "<?php echo $_paginaDefault; ?>";
</script>
</head>
<body>
  <div class='panel'>
    <div class="datagrid">
      <div class="panel-header">
        <div class="panel-title" id='title-header'>
          <h2>
            Sistema Base de Gestión de Información de Telemetría
          </h2>
        </div>
      </div>
    </div>
  </div>
  <div class="datagrid-toolbar" id='userInfo'>
    <h3>
    </h3>
  </div>
  <center>
    <div id="cc" class="easyui-layout">
      <div id="content" data-options="region:'center',title:'Iniciar Sesión'">
        <center>
          <form id="ff" method="post">
            <div>
              <br><br>
              <label for="usuario">Usuario:</label>
              <br>
              <input class="easyui-validatebox" type="text" name="usuario"
id="usuario" autofocus="autofocus">
              <br><br>
              <label for="password">Contraseña:</label>
              <br>
              <input class="easyui-validatebox" type="password" name="password"
id="password"/>
              <br><br><br>
              <a href="#" id="btn_submit" class="easyui-linkbutton">Enviar</a>
            </div>
          </form>
        </center>
      </div>
    </div>
  </center>
</body>
```

```
</body>  
</html>
```

```
*****
```

```
# NOMBRE  
# Mapa.php  
# VARIABLES  
# $tabla  
# $detalle  
# $filtro  
# $sistema  
# MODULOS  
# /Class/SQL.php  
# /Class/JSON.php  
# /Class/Menu.php  
# /Class/GRID.php  
# /Config/DB.php
```

```
*****
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<?php
```

```
session_start();
```

```
require_once '../Class/SQL.php';
```

```
require_once '../Class/JSON.php';
```

```
require_once '../Class/Menu.php';
```

```
require_once '../Class/GRID.php';
```

```
require_once '../Config/GUI.php';
```

```
if (!isset($_SESSION["user"]))
```

```
    header("Location: Login.php");
```

```
$tabla = isset($_REQUEST["tabla"]) ? $_REQUEST["tabla"] : "";
```

```
$detalle = isset($_REQUEST["detalle"]) ? $_REQUEST["detalle"] : 0;
```

```
$filtro = isset($_REQUEST["filtro"]) ? " where ".$_REQUEST["filtro"] : "";
```

```
$sistema = isset($_REQUEST["sistema"]) ? $_REQUEST["sistema"] : "";
```

```
?>
```

```
<html>
```

```
    <head>
```

```
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
        <title>Sistema Telemetría</title>
```

```
<!--<link rel="stylesheet" type="text/css"
href="http://www.jeasyui.com/easyui/themes/default/easyui.css">-->
<link rel="stylesheet" type="text/css" href="../CSS/easyui.css">
<link rel="stylesheet" type="text/css" href="../CSS/icon.css">
<link rel="stylesheet" type="text/css" href="../CSS/demo.css">
<link rel="stylesheet" type="text/css" href="../CSS/Estilos.css">
<script
src="http://maps.googleapis.com/maps/api/js?key=AlzaSyCZFBd3S4RR24Qwd7dL2IS81s
uMt_GAuSU&sensor=false"></script>
<script type="text/javascript" src="../JS/jquery-1.6.min.js"></script>
<script type="text/javascript" src="../JS/jquery.easyui.min.js"></script>
<script type="text/javascript" src="../JS/jquery.edatagrid.js"></script>
<script type="text/javascript" src="../JS/datagrid-detailview.js"></script>
<script type="text/javascript" src="../JS/Scripts.js"></script>
<script type="text/javascript" src="../JS/Maps.js"></script>
<script>
var tabla = "<?php echo $tabla; ?>";
var filtro = "<?php echo $filtro; ?>";
var detalle = "<?php echo $detalle; ?>";
var paginaDefault = "<?php echo $_paginaDefault; ?>";
</script>
</head>
<body>
<div class='panel'>
<div class="datagrid">
<div class="panel-header">
<div class="panel-title" id='title-header'>
<h2>
Sistema Base de Gestión de Información de Telemetría
</h2>
<div id="horaActual"></div>
</div>
</div>
</div>
</div>
<div class="datagrid-toolbar panelContenido" id='userInfo'>
<h3>
Usuario: <i><?php echo $_SESSION["user"]; ?></i>
<a href="#" id="btn_logout" class="easyui-linkbutton">Cerrar Sesión</a>
</h3>
</div>
<?php
$menu = new Menu();
```



```

$detalle = isset($_REQUEST["detalle"]) ? $_REQUEST["detalle"] : 0;
$tituloGrid = "Relacion Sensor-Llave"//isset($_REQUEST["tituloGrid"]) ?
$_REQUEST["tituloGrid"] : $tabla;
?>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Sistema Telemetría</title>
    <!--<link rel="stylesheet" type="text/css"
href="http://www.jeasui.com/easyui/themes/default/easyui.css">-->
    <link rel="stylesheet" type="text/css" href="../CSS/easyui.css">
    <link rel="stylesheet" type="text/css" href="../CSS/icon.css">
    <link rel="stylesheet" type="text/css" href="../CSS/demo.css">
    <link rel="stylesheet" type="text/css" href="../CSS/Estilos.css">
    <script type="text/javascript" src="../JS/jquery-1.6.min.js"></script>
    <script type="text/javascript" src="../JS/jquery.easyui.min.js"></script>
    <script type="text/javascript" src="../JS/jquery.edatagrid.js"></script>
    <script type="text/javascript" src="../JS/datagrid-detailview.js"></script>
    <script type="text/javascript" src="../JS/Scripts.js"></script>
    <script>
      var tabla = "<?php echo $tabla; ?>";
      var filtro = "<?php echo $filtro; ?>";
      var detalle = "<?php echo $detalle; ?>";
      var paginaDefault = "<?php echo $_paginaDefault; ?>";
    </script>
  </head>
  <body>
    <div class='panel'>
      <div class="datagrid">
        <div class="panel-header">
          <div class="panel-title" id='title-header'>
            <h2>
              Sistema Base de Gestión de Información de Telemetría
            </h2>
            <div id="horaActual"></div>
          </div>
        </div>
      </div>
    </div>
    <div class="datagrid-toolbar panelContenido" id='userInfo'>
      <h3>
        Usuario: <i><?php echo $_SESSION["user"]; ?></i>
        <a href="#" id="btn_logout" class="easyui-linkbutton">Cerrar Sesión</a>
      </h3>
    </div>
  </body>
</html>

```

```

    </h3>
</div>
<?php
$menu = new Menu();
echo $menu->getBotones();
echo $menu->getMenuHTML();
?>
<div class="datagrid-toolbar panelContenido panelMapa" style="height: auto;">
  <br>
  <table align="center">
    <tr>
      <td>
        <div class="easyui-panel" title="Origen de Datos" style="width:800px;">
          <form id="ff" method="post" action="../AJAX/save_origenDatos.php">
            <table align="center" border="0">
              <tr>
                <td colspan="3">
                  <label id="mensaje" style="display: none; float: right;">Cargando
información...&nbsp;&nbsp;&nbsp;</label>
                </td>
              </tr>
              <tr>
                <td colspan="2">
                  <b>Origen de datos</b>
                  <hr>
                </td>
              </tr>
              <tr>
                <td>
                  <label>Nombre:&nbsp;</label>
                </td><td>
                  <input id="nombre" name="nombre" data-options="required:true"
class="easyui-validatebox" style="width: 250px;">
                </td>
              </tr>
              <tr>
                <td>
                  <label>Servidor:&nbsp;</label>
                </td><td>
                  <input id="fk_servidor" name="fk_servidor" class="easyui-
combobox cc"

```

```

        data-
options="required:true,valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla=serv
idor',onSelect: function(rec){selectComboBox(this, rec.id);}, onLoadSuccess: function(){}">
        </td>
</tr>
<tr>
        <td>
                <label>Base de datos:&nbsp;</label>
        </td><td>
                <input id="bd" name="bd" class="easyui-combobox cc"
        data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,                rec.id);},                onLoadSuccess:
function(){setTimeout(selectComboBox(this,$('#bd').combobox('getValue')), 5000);}">
        </td>
</tr>
<tr>
        <td colspan="2">
                <br><b>Estaciones</b>
                <hr>
        </td>
</tr>
<tr>
        <td>
                <label>Tabla de estaciones:&nbsp;</label>
        </td><td>
                <input id="estaciones" name="estaciones" class="easyui-combobox
cc"
        data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,                rec.id);},                onLoadSuccess:
function(){setTimeout(selectComboBox(this,$('#estaciones').combobox('getValue')),
5000);}">
        </td>
</tr>
<tr>
        <td>
                <label>Clave de la estación (Se relacionará con el
sensor):&nbsp;</label>
        </td><td>
                <input                id="campo_estacion"                name="campo_estacion"
class="easyui-combobox cc"

```

```

        data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){$('#mensaje').hide();}">
        </td>
    </tr>
    <tr>
    <td>
        <label>Nombre de la estación:&nbsp;</label>
    </td><td>
        <input id="campo_nombre" name="campo_nombre" class="easyui-
combobox cc"

```

```

        data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){$('#mensaje').hide();}">
        </td>
    </tr>
    <tr>
    <td>
        <label>Latitud:&nbsp;</label>
    </td><td>
        <input id="campo_lat" name="campo_lat" class="easyui-
combobox cc"

```

```

        data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){$('#mensaje').hide();}">
        </td>
    </tr>
    <tr>
    <td>
        <label>Longitud:&nbsp;</label>
    </td><td>
        <input id="campo_lon" name="campo_lon" class="easyui-
combobox cc"

```

```

        data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,          rec.id);},          onLoadSuccess:
function(){$('#mensaje').hide();}">
        </td>
    </tr>
    <tr>

```



```

        <td colspan="2">
            <br><b>Valores</b>
            <hr>
        </td>
    </tr>
    <tr>
        <td>
            <label>Tabla de valores:&nbsp;</label>
        </td><td>
            <input id="tabla" name="tabla" class="easyui-combobox cc"
                data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,                rec.id);},                onLoadSuccess:
function(){setTimeout(selectComboBox(this,$('#tabla').combobox('getValue')), 5000);}">
        </td>
    </tr>
    <tr>
        <td>
            <label>Clave    (Campo    que    se    relaciona    con    la
estación):&nbsp;</label>
        </td><td>
            <input id="campo_llave" name="campo_llave" class="easyui-
combobox cc"
                data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this,                rec.id);},                onLoadSuccess:
function(){setTimeout(selectComboBox(this,$('#campo_llave').combobox('getValue')),
5000);}">
        </td>
    </tr>
    <tr>
        <td>
            <label>Fecha-Hora (Fecha y hora del valor medido):&nbsp;</label>
        </td><td>
            <input id="campo_fecha" name="campo_fecha" class="easyui-
combobox cc"
                data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this, rec.id);}, onLoadSuccess: function(){}">
        </td>
    </tr>
    <tr>
        <td>
    
```

```

        <label>Valor (Campo con el valor medido):&nbsp;&nbsp;&nbsp;</label>
    </td><td>
        <input id="campo_valor" name="campo_valor" class="easyui-
combobox cc"
        data-
options="required:true,valueField:'id',textField:'text',onSelect:
function(rec){selectComboBox(this, rec.id);}, onLoadSuccess: function(){}">
    </td>
</tr>
<tr>
    <td colspan="2">
        <br><b>Sensores</b>
        <hr>
    </td>
</tr>
<tr>
    <td>
        <label>Tipo de sensor:&nbsp;&nbsp;&nbsp;</label>
    </td><td>
        <input id="sensor" name="sensor" class="easyui-combobox cc"
        data-
options="required:true,valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla=tipo
_sensor',onSelect: function(rec){selectComboBox(this, rec.id);}, onLoadSuccess:
function(){$('#mensaje').hide();}">
    </td>
</tr>
<tr>
    <td colspan="2">
        <br><b>Extracción de datos</b>
        <hr>
    </td>
</tr>
<tr>
    <td>
        <label>Frecuencia:&nbsp;&nbsp;&nbsp;</label>
    </td><td>
        <input id="frecuencia" name="frecuencia" class="easyui-combobox
cc"
        data-
options="required:true,valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla=frec
uencia',onSelect: function(rec){selectComboBox(this, rec.id);}, onLoadSuccess:
function(){$('#mensaje').hide();}">
    </td>

```

```

        </tr>
        <tr>
            <td colspan="2" align="center">
                <br>
                <a href="#" id="btn_finalizaFuente" class="easyui-
linkbutton">Aceptar</a>
                <a href="#" id="btn_finalizaFuenteDisable" style="display: none;"
class="easyui-linkbutton">Aceptar</a>
                <br>
            </td>
        </tr>
    </table>
</form>
</div>
</td>
</tr>
</table>
<br>
</div>
</body>
</html>

```

```

# NOMBRE
# Rango_marcador.php
# VARIABLES
# $tabla
# $filtro
# $detalle
# $tituloGrid
# MODULOS
# /Class/SQL.php
# /Class/JSON.php
# /Class/Menu.php
# /Class/GRID.php
# /Config/DB.php

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<?php
session_start();

```

```
require_once '../Class/SQL.php';
require_once '../Class/JSON.php';
require_once '../Class/Menu.php';
require_once '../Class/GRID.php';
require_once '../Config/GUI.php';

if (!isset($_SESSION["user"]))
    header("Location: Login.php");

$tabla = isset($_REQUEST["tabla"]) ? $_REQUEST["tabla"] : "rango_marcador";
$filtero = isset($_REQUEST["filtro"]) ? " where " . $_REQUEST["filtro"] : "";
$detalle = isset($_REQUEST["detalle"]) ? $_REQUEST["detalle"] : 0;
$tituloGrid = isset($_REQUEST["tituloGrid"]) ? $_REQUEST["tituloGrid"] : $tabla;
?>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Sistema Telemetría</title>
    <!--<link
      rel="stylesheet"
      type="text/css"
      href="http://www.jeasyui.$_REQUEST["fuente"]com/easyui/themes/default/easyui.css">
    -->
    <link rel="stylesheet" type="text/css" href="../CSS/easyui.css">
    <link rel="stylesheet" type="text/css" href="../CSS/icon.css">
    <link rel="stylesheet" type="text/css" href="../CSS/demo.css">
    <link rel="stylesheet" type="text/css" href="../CSS/Estilos.css">
    <script type="text/javascript" src="../JS/jquery-1.6.min.js"></script>
    <script type="text/javascript" src="../JS/jquery.easyui.min.js"></script>
    <script type="text/javascript" src="../JS/jquery.edatagrid.js"></script>
    <script type="text/javascript" src="../JS/datagrid-detailview.js"></script>
    <script type="text/javascript" src="../JS/Scripts.js"></script>
    <script>
      var tabla = "<?php echo $tabla; ?>";
      var filtro = "<?php echo $filtro; ?>";
      var detalle = "<?php echo $detalle; ?>";
      var paginaDefault = "<?php echo $_paginaDefault; ?>";
      var fuente = ""; "<?php echo @$fuente; ?>";
    </script>
  </head>
  <body>
    <div class='panel'>
      <div class="datagrid">
        <div class="panel-header">
          <div class="panel-title" id='title-header'>
```

```

        <h2>
            Sistema Base de Gestión de Información de Telemetría
        </h2>
        <div id="horaActual"></div>
    </div>
</div>
</div>
</div>
</div>
<div class="datagrid-toolbar panelContenido" id='userInfo'>
    <h3>
        Usuario: <i><?php echo $_SESSION["user"]; ?></i>
        <a href="#" id="btn_logout" class="easyui-linkbutton">Cerrar Sesión</a>
    </h3>
</div>
<div style="padding-right:15px;">
    <?php
    $menu = new Menu();
    echo $menu->getBotones();
    echo $menu->getMenuHTML();
    ?>
    <br><label>Sistema de telemetría:&nbsp;</label>
    <input id="marcador_sistema" name="marcador_sistema" class="easyui-combobox
cc"
        data-
options="required:false,valueField:'id',textField:'text',url:'../AJAX/get_combo.php?tabla=mar
cador_sistema',onSelect: function(rec){selectComboBox(this, rec.id);}, onLoadSuccess:
function(){selectComboBox(this,$('#marcador_sistema').combobox('getValue'))}">
    <br><br>
    <table id="dg" title="RANGO MARCADOR" style="width:auto; height:auto"
        toolbar="#toolbar" pagination="true" pageSize="20" idField="idrango_marcador"
        rownumbers="true" fitColumns="true" singleSelect="true"
        sortName="idrango_marcador" sortOrder="asc" autoRowHeight="true">
    <thead>
    <tr>
    <th
                field="minimo"
                width="50"
editor="{type:'numberbox',options:{required:true}}" sortable="true">MINIMO</th>
    <th
                field="maximo"
                width="50"
editor="{type:'numberbox',options:{required:true}}" sortable="true">MAXIMO</th>
    <th
                field="marcador"
                width="50"
                formatter="formatImage"
editor="{type:'combobox',options:{formatter:function(row){return
combolmageFormat(row);},required:true,
                valueField:'id',
                textField:'nombre',
url:'../AJAX/get_combobox_image.php?tabla=Markers'}}"
                sortable="true">MARCADOR</th>

```

```

        <!--<th                field="fk_marcador_sistema"                width="50"
editor="{type:'combobox',options:{required:true,                valueField:'idmarcador_sistema',
textField:'nombre',
url:'../AJAX/get_combobox.php?tabla=marcador_sistema&pk=idmarcador_sistema'}}"
sortable="true"> MARCADOR SISTEMA</th-->
    </tr>
</thead>
</table>

```

```

    <div id="toolbar">
        <a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true"
onclick="javascript:$('#dg').edatagrid('addRow')">Nuevo</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-remove" plain="true"
onclick="javascript:$('#dg').edatagrid('destroyRow')">Eliminar</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-save" plain="true"
onclick="javascript:$('#dg').edatagrid('saveRow')">Guardar</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-undo" plain="true"
onclick="javascript:$('#dg').edatagrid('cancelRow')">Cancelar</a>
    </div>
    <!-- </div-->
</body>
</html>

```

```

# NOMBRE
# .php
# VARIABLES
# $
# $
# $
# $
# $
# $
# $
# $
# $
# $
# $
# MODULOS
# /Class/SQL.php
# /Class/Dump.php

```

```
# /Class/JSON.php  
# /Config/DB.php  
*****
```

Anexo 2. Código fuente de la Aplicación Móvil para la visualización de información de telemetría.

Alerta.java

```
/*
 * Clase para los mensajes de alerta de la App
 * @author José De Jesús Rosales Banderas
 */
package app.telemetria;

import android.content.Context;
import android.view.Gravity;
import android.widget.Toast;

public class Alerta {
    Context context;
    String mensaje;
    Toast t;

    //Constructores
    public Alerta(Context context, String mensaje){
        this.context = context;
        this.mensaje = mensaje;
        this.showAlerta();
    }

    public Alerta(Context context){
        this.context = context;
    }

    //Texto a mostrar en el mensaje
    public Alerta setText(String mensaje){
        this.mensaje = mensaje;
        return this;
    }

    //Metodo para mostrar alerta
    public void showAlerta(){
        t = Toast.makeText(this.context, "\n"+this.mensaje+"\n",
        Toast.LENGTH_LONG);
        t.setGravity(Gravity.CENTER, 0, Gravity.CENTER);
    }
}
```



```
        t.show();
    }

    //Metodo rapido para un mensaje
    public void Mensaje(String mensaje){
        this.setText(mensaje);
        this.showAlerta();
    }
}
```

DataManager.java

```
package app.telemetry;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

import org.json.JSONArray;
import org.json.JSONObject;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.text.Html;
import android.text.Spannable;
import android.text.Spanned;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.SimpleAdapter;
import android.widget.TextView;

@SuppressLint("NewApi")
public class DataManager {
```

```
private static final String LOGTAG = "DataManager";
private DataManagerHelper mHelper = null;
private SQLiteDatabase db = null;
private Activity activity;
private Alerta a;
private String msj = "";
private String msjAux = "";
private boolean result = false;
public String aux1 = "";
public String aux2 = "";
public String idAux = "";

public DataManager(final Context context, Activity activity) {
    this.mHelper = new DataManagerHelper(context, activity);
    this.activity = activity;
    this.a = new Alerta(context);
}

public DataManager open(){
    Log.d(LOGTAG, "Abriendo la BD...");
    db = this.mHelper.getWritableDatabase();
    return this;
}

public void close(){
    Log.d(LOGTAG, "Cerrando la BD...");
    if(db != null){
        db.close();
    }
}

public String getMsj(){
    return this.msj;
}

public String getIdAux(){
    return this.idAux;
}

public boolean getResultProcess(){
    String str = "";
    while(str==""){
        Log.d("Mensaje", str);
    }
}
```

```
        str = getMsj();
    }
return result;
}

//Metodo insertar
public int insertar(String tabla, ContentValues campos){
    Log.d(LOGTAG,"Insertando en tabla "+tabla);
    long id = db.insert(tabla, null, campos);
    return (int) id;
}

//Metodo update
public boolean update(String tabla, String llave, String valor, String campo, String
dato){
    Log.d(LOGTAG,"Actualizando dato "+campo+"="+dato+" en "+tabla+" donde
"+llave+"="+valor);
    ContentValues values = new ContentValues();
    values.put(campo, dato);
    return db.update(tabla, values, llave+"="+valor, null) > 0 ? true : false;
}

//Metodo delete
public boolean delete(String tabla, String llave, String valor, boolean borrarTodo){
    Log.d("Delete","Eliminando registro en "+tabla+" donde "+llave+"="+valor);
    return db.delete(tabla, borrarTodo ? null : llave+"="+valor, null) > 0 ? true :
false;
}

//Metodo delete
public boolean deleteMul(String tabla, String where){
    Log.d("Delete","Eliminando registro en "+tabla+" donde "+where);
    return db.delete(tabla, where, null) > 0 ? true : false;
}

//Metodo consulta
public String[][] consulta(String query, HashMap<String, String> campos){
    Log.d(LOGTAG,"Consultado query="+query);
    String consulta = Diccionario.getQuery(query);

    if(campos!=null){
        Iterator<?> it = campos.entrySet().iterator();
        while (it.hasNext()) {
```

```

        Map.Entry e = (Map.Entry)it.next();
        consulta      =      consulta.replace(e.getKey().toString(),
e.getValue().toString());
    }
}
//Log.d(LOGTAG, consulta);
return consultaToMatriz(db.rawQuery(consulta, null));
}

//Metodo consulta
public String consultaDato(String query, HashMap<String, String> campos){
    Log.d(LOGTAG,"Consultado dato de query="+query);
    String dato[][] = this.consulta(query, campos);
    if(dato!=null)
        return dato[0][0];
    else
        return null;
}

//Metodo generador de matriz
public String[][] consultaToMatriz (Cursor cursor){
    Log.d(LOGTAG,"Convirtiendo resultado a matriz");
    if(cursor.getCount()>0 && cursor!=null){
        String[][] salida = new
String[cursor.getCount()][cursor.getColumnNames().length];

        int i = 0;
        String names[] = cursor.getColumnNames();
        while (cursor.moveToNext()){
            for(int j=0; j<names.length; j++){
                salida[i][j] =
cursor.getString(cursor.getColumnIndex(names[j]));
                Log.i(names[j],salida[i][j]);
            }
            i++;
        }

        return salida;
    }
    else{
        return null;
    }
}
}

```

```
//Metodo generador de array
public String[] consultaToArray (String matriz[][]){
    Log.d(LOGTAG,"Convirtiendo resultado a arreglo");

    if(matriz!=null){
        String salida[] = new String[matriz.length];
        for(int i=0; i<matriz.length; i++){
            salida[i] = matriz[i][0];
        }
        return salida;
    }
    else{
        return null;
    }
}

//Metodo generador de adapter para combo
public SimpleAdapter consultaToSimpleAdapter(Context context, String
matriz[][]/*, boolean combo*/) {
    Log.d(LOGTAG,"Convirtiendo resultado a simple adapter");
/*
    String[] lista = consultaToArray(matriz);
    if(lista==null)
        lista = new String[]{"No hay datos"};*/

    ArrayList<HashMap<String, Spanned>> fila = new
ArrayList<HashMap<String, Spanned>>();
    for(int i=0; i<matriz.length; i++){
        HashMap<String, Spanned> mapList = new HashMap<String, Spanned>();
        mapList.put("titulo", Html.fromHtml(matriz[i][0]));
        mapList.put("descripcion", Html.fromHtml(matriz[i][1]));
        fila.add(mapList);
    }

    SimpleAdapter adapter = new SimpleAdapter(context, fila,
R.layout.listview_item, new String[]{"titulo","descripcion"}, new int[]{R.id.titulo,
R.id.descripcion});
    adapter.setViewBinder(new SimpleAdapter.ViewBinder() {
        @Override
        public boolean setViewValue(View view, Object data, String
textRepresentation) {
            if (data instanceof Spanned && view instanceof TextView) {
```

```
        ((TextView) view).setText((Spanned) data);
    } else {
        ((TextView) view).setText(String.valueOf(data));
    }
    return true;
}
}
);
return adapter;
}

//Metodo generador de adapter para combo
public ArrayAdapter<String> consultaToAdapter(Context context, String
matriz[][]/*, boolean combo*/) {
    Log.d(LOGTAG, "Convirtiendo resultado a adapter");

    String[] lista = consultaToArray(matriz);
    if(lista==null)
        lista = new String[]{"No hay datos"};

    ArrayAdapter<String> adapter;
    /*if(combo){
        adapter = new ArrayAdapter<String>(activity, R.layout.listview_item,
lista);
        adapter = new ArrayAdapter<String>(context, R.layout.list_item,
lista);
        adapter.setDropDownViewResource(R.layout.dropdown_item);
    }
    else{
        adapter = new ArrayAdapter<String>(context, R.layout.listview_item,
lista);
    }*/
    adapter = new ArrayAdapter<String>(context, R.layout.listview_item,
R.id.titulo, lista);
    return adapter;
}

public void updateCatalogos(){
    final ProgressDialog dialog = ProgressDialog.show(activity, "", "Cargando
catalogos. Un momento por favor...", true);
    final String tablas[] = Diccionario.tablasActualizables;
    Thread t = new Thread(){
        public void run(){
```

```

Log.d(LOGTAG, "Actualizando tablas...");
try {
    int x = 0;
    //Tablas BD
    Post post = new Post();
    JSONArray datos;
    for(int i=0; i<tablas.length; i++){
        datos =
post.getServerData(null,"AJAX/NOS_datosMovil.php?key="+tablas[i], false);
        db.execSQL("DELETE FROM "+tablas[i]);
        Log.d(LOGTAG, "Vaciado correcto "+tablas[i]);

        if(datos!=null && datos.length()>0){
            for(x=0;x<datos.length();x++){
                ContentValues values = new
ContentValues();
                JSONObject json_data =
datos.getJSONObject(x);
                Iterator<?> fila = json_data.keys();
                while(fila.hasNext()){
                    String key = (String)fila.next();
                    values.put(key,
json_data.getString(key));
                }
                db.insert(tablas[i], null, values);
            }
            Log.d(LOGTAG, "Carga correcta de "+tablas[i]);
        } else {
            Log.d(LOGTAG, "Error al cargar
"+tablas[i]+"/no hay datos");
        }
    }
} catch (Exception e) {
    Log.d(LOGTAG, "Error al conectar con el servidor.
Error:"+e.toString());
}

dialog.dismiss();
activity.runOnUiThread(
    new Runnable(){
        public void run(){
            a.setText("Se han cargado los
catalogos").showAlerta());

```

```
        }  
        });  
    }  
    };  
    t.start();  
}  
}
```

DataManagerHelper.java

```
package app.telemetria;
```

```
import android.app.Activity;  
import android.app.AlertDialog;  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.util.Log;
```

```
public class DataManagerHelper extends SQLiteOpenHelper {  
    private static final String LOGTAG = "DataManagerHelper";  
    private static final String DATABASE_NAME = "sistema_telemetria.db";  
    private static final int DATABASE_VERSION = 1;  
    private static final String tablas[] = new String[]{  
        "CREATE TABLE sesion (idusuario INTEGER, nombre TEXT, password TEXT)",  
        "CREATE TABLE grupos (idgrupo INTEGER, nombre TEXT, descripcion TEXT)",  
        "CREATE TABLE grupo_vista (idgrupo INTEGER, idvista INTEGER)",  
        "CREATE TABLE valores (idvista INTEGER, valor REAL, hora INTEGER)",  
        "CREATE TABLE tema (idtema INTEGER, fondo_panel TEXT, borde_panel  
TEXT, fondo_grid TEXT, grid_ejex TEXT, grid_ejey TEXT, etiquetas_ejex TEXT,  
etiquetas_ejey TEXT, borde_serie TEXT, fondo_serie TEXT, color_titulo TEXT, ancho_barra  
TEXT, ancho_linea TEXT)",  
        "CREATE TABLE vista (idvista INTEGER, idtema INTEGER, tipo_vista TEXT,  
sensor TEXT, vista TEXT, max REAL, min REAL, mensaje TEXT, enviado TEXT, sistema TEXT,  
estacion TEXT, latitud TEXT, longitud TEXT, unidad TEXT)"  
    };  
    private Activity activity;
```

```
    public DataManagerHelper(Context context, Activity activity) {  
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
```



```
        this.activity = activity;
    }

    @Override
    public void onCreate(final SQLiteDatabase db) {
        // TODO Auto-generated method stub
        Log.d(LOGTAG, "Creando tablas iniciales...");
        try{
            for(int i=0; i<tablas.length; i++)
                db.execSQL(tablas[i]);
        }
        catch(Exception e){
            Log.i(LOGTAG, e.toString());
        }
        final ProgressDialog dialog = ProgressDialog.show(activity, "", "Creando Base
de Datos", true);

        Thread t = new Thread(){
            public void run(){
                dialog.dismiss();

                activity.runOnUiThread(
                    new Runnable(){
                        public void run(){
                            activity.recreate();
                        }
                    }
                );
            }
        };
        t.start();
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.d(LOGTAG, "Actualizando BD de version "+oldVersion+" a versión
"+newVersion);
    }

    @SuppressWarnings("unused")
    private void updateVersion(){

    }
}
```

DetalleActivity.java

```
package app.telemetria;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Timer;
import java.util.TimerTask;

import org.json.JSONArray;
import org.json.JSONObject;

import android.support.v7.app.ActionBarActivity;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.WindowManager;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ScrollView;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

public class DetalleActivity extends ActionBarActivity {
    private Dispositivo dispositivo;
```

```
private DataManager DBManager = null;
private Activity activity;
private Network red;
private Alerta alerta;
private String idUsuario, idVista, nombre;
private int x, width, resultTimer = 0;
private LinearLayout ln_grafica, ln_informacion, ln_tabla;
private boolean esCompartido = false, dialogAbierto = false;
private TextView informacion, vActual, titulo, vporcentaje;
private ImageView flecha;
private ScrollView sc_tabla;
private Button compartir;
private Grafica graph;
private Timer timer;
private AlertDialog.Builder builderNotificacion;

@Override
protected void onCreate(Bundle savedInstanceState) {
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_detalle_);
//Recepcion de variables
Intent i = getIntent();
idVista = i.getStringExtra("idVista");
esCompartido = i.getBooleanExtra("compartido", false);
//Variables generales
DBManager = new DataManager(getApplicationContext(), this);
DBManager.open();
activity = this;
red = new Network(getApplicationContext(), this);
alerta = new Alerta(this);
//Obtener usuario logeado
idUsuario = DBManager.consultaDato("usuarioConectado", null);
//Obtener nombre del usuario logeado
nombre = DBManager.consultaDato("getNombreUsuario", null);
setTitle(getTitle() + " - (" + nombre + ")");
//Dimensionar layout y cargar graficas
Layout();
CargaGrafica();
}
/*
@Override
```

```

public void onBackPressed() {
    //super.onBackPressed();
    Intent i = getIntent();
    i.putExtra("men", "hola");
    setResult(RESULT_OK, i);
    this.finish();
}
*/
public void CargaGrafica(){
    double ultimo = 0, penultimo = 0;
    Long tiempo = null;
    final String vistas[][] = DBManager.consulta("getVistald", new HashMap<String,
String>(){{put("<vista>", idVista)}});
    if(vistas!=null){
        //Recorrer vistas, posiciones:
idvista,idtema,tipo_vista,sensor,vista,max,min,mensaje,enviado
        for(x=0; x<vistas.length; x++){
            //Obtencion del tema, posiciones: idtema, fondo_panel, borde_panel,
fondo_grid, grid_ejex, grid_ejey, etiquetas_ejex, etiquetas_ejey, borde_serie, fondo_serie,
color_titulo, ancho_barra, ancho_linea
            String temas[][] = DBManager.consulta("getTema", new
HashMap<String, String>(){{put("<tema>", vistas[x][1])}});
            String tema[] = temas[0];
            //Obtencion de valores, posiciones: valor, hora
            String valores[][] = DBManager.consulta("getValores", new
HashMap<String, String>(){{put("<vista>", vistas[x][0])}});
            //Grafica
            graph = new Grafica(activity, vistas[x][4],
Integer.parseInt(vistas[x][0]));
            graph.setRangoY(Double.parseDouble(vistas[x][6]),
Double.parseDouble(vistas[x][5]), Math.round(((Double.parseDouble(vistas[x][5])-
Double.parseDouble(vistas[x][6]))/4) * 100) /100);
            graph.setColorPanel(tema[1], tema[2]);
            graph.setColorGrid(tema[4], tema[5], tema[3]);
            graph.setColorEtiquetasXY(tema[6], tema[7]);
            if(vistas[x][2].equals("Barras"))
                graph.setBordeyRellenoBarra(tema[8], tema[9],
Integer.parseInt(tema[11]));
            else
                graph.setBordeyRellenoSerie(tema[8], tema[9],
Integer.parseInt(tema[12]), vistas[x][2].equals("Areas")?true:false);
            graph.setColorTitulo(tema[10]);
            //Agregar series a los array

```

```
ArrayList<Long> fechas = new ArrayList<Long>();
ArrayList<Double> valor = new ArrayList<Double>();
//Tabla de datos
TableLayout tabla = new TableLayout(this);
//Filas de la tabla
TableRow fila = new TableRow(this);
fila.setBackgroundResource(R.color.GrisOpaco);
fila.setPadding(1, 1, 1, 0);
TextView tFecha = new TextView(this);
tFecha.setText(" FECHA - HORA");
tFecha.setTextColor(Color.WHITE);
tFecha.setWidth((width/2));
tFecha.setBackgroundResource(R.color.GrisSombra);
TextView tValor = new TextView(this);
tValor.setText(" VALOR (" + vistas[x][13] + ")");
tValor.setTextColor(Color.WHITE);
tValor.setWidth((width/2));
tValor.setBackgroundResource(R.color.GrisSombra);
fila.addView(tFecha);
fila.addView(tValor);
tabla.addView(fila);
//Fecha
    Calendar calendar = Calendar.getInstance();
//Recorrido de valores
    for(int i=0; i<valores.length; i++){
        valor.add(Double.parseDouble(valores[i][0]));

        fechas.add(Long.parseLong(valores[i][1]));
//Ultimos valores
        penultimo = ultimo;
        ultimo = Double.parseDouble(valores[i][0]);
        tiempo = Long.parseLong(valores[i][1]);
        calendar.setTimeInMillis(tiempo);
//Filas de la tabla
        fila = new TableRow(this);
        fila.setBackgroundResource(R.color.GrisOpaco);
        fila.setPadding(1, 1, 1, 0);
        tFecha = new TextView(this);
        tFecha.setText("      " + new SimpleDateFormat("dd/MM/yyyy
HH:mm:ss").format(calendar.getTime()));
        tFecha.setTextColor(Color.LTGRAY);
        tFecha.setWidth((width/2));
        tFecha.setBackgroundResource(R.color.GrisTabla);
```

```

tValor = new TextView(this);
tValor.setText(valores[i][0]);
tValor.setTextColor(Color.LTGRAY);
tValor.setWidth((width/2));
tValor.setBackgroundResource(R.color.GrisTabla);
fila.addView(tFecha);
fila.addView(tValor);
tabla.addView(fila, 1);
    }
In_tabla.addView(tabla);
//Porcentaje
double porcentaje = Math rint( (Math.abs(ultimo - penultimo) * 100 /
penultimo) * 100 ) / 100;
//Agregar quien lo envio solo en grupo compartido
if(esCompartido)
    graph.setPie("Enviado por: "+vistas[x][8]);
//Dibujar grafica
graph.addSerie(fechas, valor, vistas[x][3], vistas[x][2]);
//Agregar listener comentarios
//Habilitar opcion solo con compartidos
if(esCompartido)
graph.getGrafica().setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        final String id = graph.getIdGrafica() + "";
        String mensaje =
DBManager.consultaDato("getMensajeVista", new HashMap<String,
String>(){put("<vista>", id)});
        Dialog dialog = new Dialog(activity);
        dialog.MensajeDialog(mensaje);
    }
});
//Agregar grafica a la pantalla
In_grafica.addView(graph.getGrafica());
//Agregar textos de informacion
titulo.setText("Usuario: "+nombre+"\nHora:" + new
SimpleDateFormat("HH:mm:ss a").format(Calendar.getInstance().getTime()) );
String salto = getResources().getBoolean(R.bool.isTablet) ? "\n\n" : "\n";
informacion.setText("Sistema de telemetría:
"+vistas[x][9]+salto+"Estación: "+vistas[x][10]+salto+"Sensor:
"+vistas[x][3]+salto+"Ultima actualización: "+ (tiempo!=null ? new

```

```

SimpleDateFormat("dd/MM/yyyy HH:mm:ss a").format(calendar.getTime()):"-")
+salto+"Latitud: "+vistas[x][11]+"\\nLongitud: "+vistas[x][12]);
        vActual.setText("Valor actual: "+ultimo+vistas[x][13]);
        vporcentaje.setText((porcentaje>=0 ? porcentaje : "0.0")+ "% ");
        flecha.setImageResource(ultimo==penultimo ? R.drawable.equal :
(ultimo>penultimo ? R.drawable.up : R.drawable.down));
    }
}
}

public void Layout(){
    //Contenedores
    ln_grafica = (LinearLayout) this.findViewById(R.id.grafica);
    ln_informacion = (LinearLayout) this.findViewById(R.id.informacion);
    ln_tabla = (LinearLayout) this.findViewById(R.id.tabla);
    informacion = (TextView) this.findViewById(R.id.txtInformacion);
    vActual = (TextView) this.findViewById(R.id.txtActual);
    titulo = (TextView) this.findViewById(R.id.txtTitulo);
    vporcentaje = (TextView) this.findViewById(R.id.txtPorcentaje);
    flecha = (ImageView) this.findViewById(R.id.arrow);
    sc_tabla = (ScrollView) this.findViewById(R.id.scrollTabla);
    compartir = (Button) this.findViewById(R.id.compartir);
    //Listener del boton compartir
    compartir.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            Dialog dialog = new Dialog(activity);
            dialog.showCompartir("Compartir vista
\\\""+graph.getVistaNombre()+"\\\"", graph.getIdGrafica(), idUsuario);
        }
    });
    //Verificamos el dispositivo
    dispositivo = new Dispositivo(this);
    int divisor = 4;
    double porcentaje = 0.08;
    //Tablet
    if(getResources().getBoolean(R.bool.isTablet)){
        //Tamaño de los contenedores
        LayoutParams params_g = ln_grafica.getLayoutParams();
        LayoutParams params_i = ln_informacion.getLayoutParams();
        LayoutParams params_s = sc_tabla.getLayoutParams();
        double ln_x = (dispositivo.getWidth() - (5*divisor)) / divisor;

```

```

double ln_y = (dispositivo.getHeight() - (dispositivo.getHeight()*porcentaje))
/ divisor;
width = (int) ln_x * 3;
params_g.height = (int) ln_y;
params_g.width = (int) ln_x * 3;
params_i.width = (int) ln_x;
params_s.height = (int) ln_y / 2;
}
//Celular
else{
    divisor = 1;
    LayoutParams params_s = sc_tabla.getLayoutParams();
    LayoutParams params_g = ln_grafica.getLayoutParams();
    LayoutParams params_i = ln_informacion.getLayoutParams();
    double ln_x = (dispositivo.getWidth() - (5*divisor)) / divisor;
    double ln_y = (dispositivo.getHeight() - (dispositivo.getHeight()*porcentaje))
/ divisor;
width = (int) ln_x;
params_g.height = (int) ln_y;
params_g.width = (int) ln_x;
params_i.width = (int) ln_x;
params_s.width = (int) ln_x;
params_s.height = (int) ln_y;
}
}

private void verificarCompartidos(){
    try{
        //Verificamos el acceso a la red
        if (red.networkAvailable()==true){
            Thread t = new Thread() {
                public void run() {
                    //Consulta del Webservice
                    Post post = new Post();
                    JSONArray datos;
                    datos = post.getServerData(null,
Diccionario.URL + "?clave=alertaVistaCompartida&user="+idUsuario, false);
                    try {
                        if (datos != null && datos.length() > 0)
{
                            //Obtencion de la respuesta
                            for (int x = 0; x < datos.length();
x++) {

```



```

                                JSONObject    json_data    =
datos.getJSONObject(x);
                                Iterator<?>    fila    =
json_data.keys();
                                while (fila.hasNext()) {
                                String    key    =
                                (String) fila.next();

                                if(key.equals("alerta")){

                                resultTimer = json_data.getString(key).equals("1") ? 1 : 0;
                                }
                                }
                                } else { }
                                }
                                catch (Exception e) {
                                Log.i("Error ",e.toString());
                                }
                                activity.runOnUiThread(new Runnable() {
                                public void run() {
                                //Si hay notificacion abrimos el
                                dialog
                                if(resultTimer == 1){
                                notificacionDialog();
                                }

                                resultTimer = 0;
                                }
                                });
                                }
                                };
                                t.start();
                                }
                                }
                                catch(Exception e){ }
                                }

private void notificacionDialog(){
    try{
        //Si el dialog no esta abierto
        if(dialogAbierto==false){

```

```

dialogAbierto = true;
//Creacion del dialog
builderNotificacion = new AlertDialog.Builder(activity);
builderNotificacion.setTitle("Notificación");
builderNotificacion.setMessage("¡Tienes una nueva vista
compartida!");

//Agregar botones de accion dependiendo del grupo actual
builderNotificacion.setPositiveButton(esCompartido ?
"Actualizar" : "Ver Compartidos", new DialogInterface.OnClickListener() {

@Override
public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub
//Actualizar vistas
Intent i = getIntent();
i.putExtra("compartido", true);
setResult(RESULT_OK, i);
activity.finish();
// grupo = esCompartido ? grupo : "Compartido";
// actualizarVistas();
dialogAbierto = false;
}
});
builderNotificacion.setNegativeButton("Cerrar", new
DialogInterface.OnClickListener() {

@Override
public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub
dialogAbierto = false;
}
});
builderNotificacion.show();
}
}
catch(Exception e){ }
}

@Override
protected void onStart() {
super.onStart();
//En cuanto inicie la ventana crear timer para revisar las notificaciones
timer = new Timer();
timer.scheduleAtFixedRate(new TimerTask(){

```

```
        @Override
        public void run() {
            // TODO Auto-generated method stub
            verificarCompartidos();
        }
    }, 0, 20000);
}

@Override
protected void onStop() {
    super.onStop();
    //Cuando pare la ventan cancelar el timer de notificaciones
    timer.cancel();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    //getMenuInflater().inflate(R.menu.detalle, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    //int id = item.getItemId();
    //Cerrar sesion
    return super.onOptionsItemSelected(item);
}
}
```

Dialog.java

```
package app.telemetry;
```

```
import java.util.ArrayList;
import java.util.Iterator;
```

```
import org.json.JSONArray;
import org.json.JSONObject;

import android.annotation.SuppressLint;
import android.annotation.TargetApi;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.os.Build;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ListView;

@SuppressLint("NewApi")
public class Dialog {
    private Activity activity;
    private ArrayAdapter<String> adapter;
    private Alerta alerta;
    private AlertDialog.Builder builder;
    private Network red;
    private String[] usuarios;
    private int result;
    private ListView lv;

    Dialog(Activity activity){
        this.activity = activity;
        alerta = new Alerta(activity);
        red = new Network(activity, activity);
    }

    public void MensajeDialog(String mensaje){
        //Creacion del dialog
        AlertDialog.Builder builderNotificacion = new AlertDialog.Builder(activity,
R.style.TransparentDialog);
        builderNotificacion.setTitle("Nota");
        builderNotificacion.setMessage(new
EstiloTexto().setTextStyleItalic(mensaje));
```

```
//Agregar botones de accion dependiendo del grupo actual
builderNotificacion.setNegativeButton("Cerrar", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        // TODO Auto-generated method stub
    }
});
builderNotificacion.show();
}

public void showCompartir(String titulo, int id, String idUsuario){
    compartirVistaDialog(titulo, id, idUsuario);
    getUsuarios(idUsuario);
}

@TargetApi(Build.VERSION_CODES.HONEYCOMB)
private void compartirVistaDialog(String titulo, final int id, final String idUsuario){
    //Agregamos el layout al dialog
    LayoutInflater inflater = activity.getLayoutInflater();
    View dialoglayout = inflater.inflate(R.layout.dialog_layout, null);
    builder = new AlertDialog.Builder(activity, R.style.TransparentDialog);
    builder.setView(dialoglayout);
    lv = (ListView) dialoglayout.findViewById(R.id.usuarios);
    final CheckBox sms = (CheckBox) dialoglayout.findViewById(R.id.enviaSMS);
    final EditText msj = (EditText) dialoglayout.findViewById(R.id.mensaje);
    EditText et = (EditText) dialoglayout.findViewById(R.id.filtro_usuario);
    //Agregamos un filtro al editText
    et.addTextChangedListener(new TextWatcher() {
        @Override
        public void onTextChanged(CharSequence cs, int arg1, int arg2, int arg3) {
            // When user changed the Text
            adapter.getFilter().filter(cs);
        }
        @Override
        public void beforeTextChanged(CharSequence arg0, int arg1, int arg2, int arg3) {
            // TODO Auto-generated method stub
        }
        @Override
        public void afterTextChanged(Editable arg0) {
            // TODO Auto-generated method stub
        }
    });
});
```

```
//Titulo y botones de accion
builder.setTitle(titulo);
builder.setPositiveButton("Aceptar", new DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface dialog, int which) {
        // TODO Auto-generated method stub
        //Items seleccionados
        String item = "";
        for(int i = 0; i<lv.getChildCount(); i++){
            CheckBox c = (CheckBox) lv.getChildAt(i);
            if(c.isChecked())
                item += (item.equals("")?"":",") + c.getText().toString();
        }

        if(!item.equals(""))
            compartirVista(item, id, msj.getText().toString(),
sms.isChecked(), idUsuario);
        else
            alerta.Mensaje("No has seleccionado ningun usuario");
    }
});
builder.setNegativeButton("Cancelar", new DialogInterface.OnClickListener()
{

    @Override
    public void onClick(DialogInterface dialog, int which) {
        // TODO Auto-generated method stub
    }
});
}

private void getUsuarios(final String idUsuario){
    //Verificamos el acceso a la red
    if (red.networkAvailable()==true){
        //Dialogo de progreso
        final ProgressDialog dialog = new ProgressDialog(activity,
R.style.TransparentDialog);
        dialog.setMessage("Cargando valores...");
        dialog.setCancelable(false);
        dialog.setCanceledOnTouchOutside(false);
        dialog.setIndeterminate(true);
        dialog.show();
    }
}
```

```

Thread t = new Thread() {
    public void run() {
        //Consulta del Webservice
        Post post = new Post();
        JSONArray datos;
        datos = post.getServerData(null, Diccionario.URL +
"?clave=usuariosCompartir&user="+idUserio, false);
        try {
            if (datos != null && datos.length() > 0) {
                //Listado de usuarios
                usuarios = new String[datos.length()];
                for (int x = 0; x < datos.length(); x++) {
                    JSONObject json_data =
datos.getJSONObject(x);
                    Iterator<?> fila =
json_data.keys();
                    while (fila.hasNext()) {
                        String key = (String)
fila.next();
                        if(key.equals("result")){
                            result =
json_data.getString(key).equals("error") ? 0 : 1;
                        }
                        else
                            usuarios[x] =
json_data.getString(key);
                    }
                }
            }
            } else {
                alerta.Mensaje("No hay usuarios para
compartir...");
            }
        }
        catch (Exception e) {
            //alerta.Mensaje("Error: "+e.toString());
        }
        dialog.dismiss();
        activity.runOnUiThread(new Runnable() {
            public void run() {
                //Mensaje de respues

```

```

        if(result == 0){
            alerta.Mensaje("Error al cargar
datos");
        }
        else{
            //Agregar usuarios al listView y
abrir ventana de dialogo
            adapter = new
ArrayAdapter<String>(activity, R.layout.listview_item_dialog, R.id.check, usuarios);
            lv.setAdapter(adapter);
            builder.show();
        }
        result = 0;
    }
});
}
};
t.start();
}
}
}

```

```

private void compartirVista(final String usuariosCompartir, final int vista, final String
mensaje, final boolean sms, final String idUsuario){
    //Verificamos el acceso a la red
    if (red.networkAvailable()==true){
        //Dialogo de proceso
        //
        final ProgressDialog dialog = ProgressDialog.show(activity, "",
"Enviando datos", true);
        final ProgressDialog dialog = new ProgressDialog(activity,
R.style.TransparentDialog);
        dialog.setMessage("Enviando datos...");
        dialog.setCancelable(false);
        dialog.setCanceledOnTouchOutside(false);
        dialog.setIndeterminate(true);
        dialog.show();
        Thread t = new Thread() {
            public void run() {
                //Consulta del Webservice
                Post post = new Post();
                JSONArray datos;
                ArrayList<String> parametros = new
ArrayList<String>();
            }
        };
        t.start();
    }
}

```

```
/*
 * Clase que contiene las variables globales de la App
 * @author José De Jesús Rosales Banderas
 */

package app.telemetry;

import java.util.HashMap;

import android.content.Context;

public class Diccionario {

    /* Tablas para actualizar/bajar datos del sistema */
    static public String tablasActualizables[] = {
        "sesion"
    };

    /* Consultas */
    static public HashMap<String, String> query = new HashMap<String, String>(){
        {
            put("usuarioConectado", "select idusuario from sesion");
            put("getGrupos", "select nombre,descripcion,idgrupo from (select case when
nombre='Compartido' then 0 else 1 end orden,nombre,descripcion,idgrupo from grupos) t
order by orden,nombre");
            put("getIdGrupo", "select idgrupo from grupos where nombre='<grupo>");
            put("existeTema", "select idtema from tema where idtema=<tema>");
            put("getVistas", "select v.* from vista v inner join grupo_vista gv on
gv.idvista=v.idvista inner join grupos g on g.idgrupo=gv.idgrupo where g.nombre like
'<grupo>' order by v.vista");
            put("getVistald", "select v.* from vista v where v.idvista = <vista>");
            put("getTema", "select * from tema where idtema=<tema>");
            put("getValores", "select valor,hora from valores where idvista=<vista> order
by hora");
            put("getVistasDelete", "select v.idvista from vista v inner join grupo_vista gv
on gv.idvista=v.idvista inner join grupos g on g.idgrupo=gv.idgrupo where g.nombre like
'<grupo>");
            put("getTemasDelete", "select distinct v.idtema from vista v inner join
grupo_vista gv on gv.idvista=v.idvista inner join grupos g on g.idgrupo=gv.idgrupo where
g.nombre like '<grupo>");
            put("getVistaUsaTema", "select count(*) from vista where idtema=<tema>");
            put("existeVista", "select idvista from vista where idvista=<vista>");
        }
    };
}
```

```
        put("getMensajeVista", "select mensaje from vista where idvista=<vista>");
        put("vistas", "select * from vista where idvista=<vista>");
        put("getNombreUsuario", "select nombre from sesion");
    }
};

//Ip del dervidro
static public String ip = "pnh.gob.mx";
// static public String ip = "172.16.2.151";
    /* URL del sistema web*/
static public String Server = "http://" + ip + "/sistema_telemetria/";
//Direccion web service
static public String URL = "Services/WebServiceAndroid.php";
//Variables para el listview
public static String TITULO = "titulo";
public static String DESCRIPCION = "descripcion";
public static String WEB = "web";
public static String USUARIO = "usuario";

//Metodo obtener un query
static public String getQuery(String llave){
    return query.get(llave);
}

//Metodo para ver la informacion del sistema
static public void info(Context context){
    Alerta a = new Alerta(context);
    a.setText("Servidor: " + Server).showAlerta();
}
}
```

Dispositivo.java

```
package app.telemetry;
```

```
import android.annotation.SuppressLint;
import android.app.Activity;
import android.graphics.Point;
import android.view.Display;
```

```
@SuppressWarnings("NewApi")
public class Dispositivo {
    private Activity activity;
    private Display display;
    private Point size;

    public Dispositivo(Activity activity){
        this.activity = activity;
        this.display = this.activity.getWindowManager().getDefaultDisplay();
    }
    this.size = new Point();
    this.display.getSize(this.size);
}

//Ancho en pixeles
public float getWidth(){
//    Log.i("W", ""+this.size.x);
    return this.size.x;
}

//Alto en pixeles
public float getHeight(){
//    Log.i("H", ""+this.size.y);
    return this.size.y;
}
}
```

EstiloTexto.java

```
package app.telemetria;
```

```
import android.graphics.Typeface;
import android.text.SpannableString;
import android.text.style.StyleSpan;
```

```
public class EstiloTexto {
```

```
    public static CharSequence setTextStyleItalic(CharSequence text) {
        try{
            final StyleSpan style = new StyleSpan(Typeface.ITALIC);
            final SpannableString str = new SpannableString(text);
```

```
        str.setSpan(style, 0, text.length(), 0);
        return str;
    }
    catch(Exception e){
        return text;
    }
}
}
```

FormatoFecha.java

```
package app.telemetria;

import java.text.FieldPosition;
import java.text.Format;
import java.text.ParsePosition;
import java.text.SimpleDateFormat;
import java.util.Date;

class FormatoFecha extends Format {
    //private String format = "dd/MM/yy HH:mm:ss";
    private String format = "HH:mm:ss";
    private SimpleDateFormat dateFormat = new SimpleDateFormat(format);

    @Override
    public StringBuffer format(Object obj, StringBuffer toAppendTo, FieldPosition pos) {
        long timestamp = ((Number) obj).longValue();
        Date date = new Date(timestamp);
        return dateFormat.format(date, toAppendTo, pos);
    }

    @Override
    public Object parseObject(String source, ParsePosition pos) {
        return null;
    }
}
```

Grafica.java

```
*****  
package app.telemetria;  
  
import java.util.ArrayList;  
  
import android.content.Context;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.util.Log;  
  
import com.androidplot.ui.AnchorPosition;  
import com.androidplot.ui.SizeLayoutType;  
import com.androidplot.ui.SizeMetrics;  
import com.androidplot.ui.XLayoutStyle;  
import com.androidplot.ui.YLayoutStyle;  
import com.androidplot.xy.BarFormatter;  
import com.androidplot.xy.BarRenderer;  
import com.androidplot.xy.BoundaryMode;  
import com.androidplot.xy.LineAndPointFormatter;  
import com.androidplot.xy.SimpleXYSeries;  
import com.androidplot.xy.XYPlot;  
import com.androidplot.xy.XYSeries;  
import com.androidplot.xy.XYStepMode;  
  
public class Grafica {  
    public XYPlot plot;  
    private BarFormatter barraFormato = null;  
    private LineAndPointFormatter serieFormato = null;  
    private XYSeries serie;  
    private int anchoBarra = 1;  
    private int colorSerie = 0;  
    private int colorGrid = 0;  
    private int colorTitulo = 1;  
    private int id;  
    private String serietxt;  
    private String vista;  
    private Context context;  
  
    Grafica(Context context, String tituloGrafica, int id){  
        this.id = id;  
        this.vista = tituloGrafica;  
        this.context = context;  
    }  
}
```

```
plot = new XYPlot(context, tituloGrafica);
plot.setDomainValueFormat(new FormatoFecha());
    }

    public int getIdGrafica(){
        return this.id;
    }

    public String getSerieNombre(){
        return this.serie.txt;
    }

    public String getViewaNombre(){
        return this.vista;
    }

    public XYPlot getGrafica(){
        return plot;
    }

    public XYPlot redibujar(){
        plot.redraw();
        return plot;
    }

    public Grafica setRangoY(double auxMin, double auxMax, double incremento){
        plot.setRangeBoundaries(auxMin, BoundaryMode.FIXED, auxMax,
BoundaryMode.FIXED);
        plot.setRangeStep(XYStepMode.INCREMENT_BY_VAL, incremento);
        return this;
    }

    public LineAndPointFormatter setBordeyRellenoSerie(String colorBordeHex, String
colorRellenoHex, int ancho, boolean area){
        colorSerie = hexToRgb(colorBordeHex, 160);
        serieFormato = new LineAndPointFormatter(hexToRgb(colorBordeHex,
200),0x000000,hexToRgb(colorRellenoHex, area==true? 100 : 0),null);
        Paint paint = serieFormato.getLinePaint();
        paint.setStrokeWidth(ancho);
        serieFormato.setLinePaint(paint);
        return serieFormato;
    }
}
```

```
public BarFormatter setBordeyRellenoBarra(String colorBordeHex, String
colorRellenoHex, int ancho){
    colorSerie = hexToRgb(colorBordeHex, 160);
    barraFormato = new BarFormatter(hexToRgb(colorRellenoHex, 100),
hexToRgb(colorBordeHex, -1));
    anchoBarra = ancho;
    return barraFormato;
}

public Grafica setColorEtiquetasXY(String colorXHex, String colorYHex){
//Color de labels XY
plot.getGraphWidget().getDomainLabelPaint().setColor(hexToRgb(colorYHex, -1));
plot.getGraphWidget().getRangeLabelPaint().setColor(hexToRgb(colorXHex, -1));
    return this;
}

public Grafica setColorGrid(String lineaXHex, String lineaYHex, String fondoHex){
    colorGrid = hexToRgb(fondoHex, -1);
//Color de fondo de la grafica
plot.getGraphWidget().getGridBackgroundPaint().setColor(colorGrid);
//Color de la linea XY
plot.getGraphWidget().getDomainGridLinePaint().setColor(hexToRgb(lineaYHex, -1));
plot.getGraphWidget().getRangeGridLinePaint().setColor(hexToRgb(lineaXHex, -1));
    return this;
}

public Grafica setColorPanel(String fondoHex, String BordeHex){
//Color de fondo contenedor de la grafica
plot.getGraphWidget().getBackgroundPaint().setColor(hexToRgb(fondoHex, -1));
//Color de fondo contenedor principal
plot.getBackgroundPaint().setColor(hexToRgb(fondoHex, -1));
//Borde del contenedor principal
plot.getBorderPaint().setColor(hexToRgb(BordeHex, -1));
    return this;
}

public Grafica setColorTitulo(String colorHex){
//Color del titulo de la grafica
colorTitulo = hexToRgb(colorHex, -1);
plot.getTitleWidget().getLabelPaint().setColor(colorTitulo);
}
```



```
        plot.getTitleWidget().getLabelPaint().setTextSize(this.context.getResources().getDimension(R.dimen.tamanoTextoGrafica));
        plot.setTitle(this.vista);
        return this;
    }

    public Grafica setPie(String texto){
        plot.getDomainLabelWidget().getLabelPaint().setColor(colorTitulo);

        plot.getDomainLabelWidget().getLabelPaint().setTextSize(this.context.getResources().getDimension(R.dimen.tamanoTextoGrafica));
        plot.setDomainLabel(texto);
        return this;
    }

    public Grafica addSerie(ArrayList<Long> x, ArrayList<Double> y, String nombre, String tipo){
        serietxt = nombre;
        serie = new SimpleXYSeries(x, y, nombre);
        leyenda(nombre);
        plot.clear();

        if(tipo.equals("Barras")){
            plot.addSeries(serie, barraFormato);
            BarRenderer renderer = (BarRenderer)plot.getRenderer(BarRenderer.class);
            renderer.setBarWidth(anchoBarra);
        }
        else if(tipo.equals("Lineas") || tipo.equals("Areas")){
            plot.addSeries(serie, serieFormato);
        }

        return this;
    }

    private void leyenda(String nombre){
        //Transparencia del recuadro
        Paint bgPaint = new Paint();
        bgPaint.setColor(this.colorGrid);
        bgPaint.setStyle(Paint.Style.FILL);
        bgPaint.setAlpha(100);
        plot.getLegendWidget().setBackgroundPaint(bgPaint);
        //Espacio entre texto y recuadro
```

```
plot.getLegendWidget().setPadding(3, 0, 0, 0);
//Color del texto
plot.getLegendWidget().getTextPaint().setColor(this.colorSerie);
//Posicion del recuadro
plot.getLegendWidget().position(5,  XLayoutStyle.ABSOLUTE_FROM_RIGHT,  35,
YLayoutStyle.ABSOLUTE_FROM_BOTTOM, AnchorPosition.RIGHT_BOTTOM);

plot.getLegendWidget().getTextPaint().setTextSize(this.context.getResources().ge
tDimension(R.dimen.tamanoTextoGrafica));
//Ajuste del recuadro al nombre, 11 por cada letra
plot.getLegendWidget().setSize(new  SizeMetrics(30,  SizeLayoutType.ABSOLUTE,
(nombre.length()*(this.context.getResources().getBoolean(R.bool.isTablet)?8:10)),
SizeLayoutType.ABSOLUTE));
}

private int hexToRgb(String colorHex, int alfa) {
    int color = 0;
    try{
        if(alfa == -1){
            color = Color.rgb(
                //alfa, //Transparencia
                Integer.valueOf( colorHex.substring( 1, 3 ), 16 ), //Rojo
                Integer.valueOf( colorHex.substring( 3, 5 ), 16 ), //Verde
                Integer.valueOf( colorHex.substring( 5, 7 ), 16 ) //Azul
            );
        } else {
            color = Color.argb(
                alfa, //Transparencia
                Integer.valueOf( colorHex.substring( 1, 3 ), 16 ), //Rojo
                Integer.valueOf( colorHex.substring( 3, 5 ), 16 ), //Verde
                Integer.valueOf( colorHex.substring( 5, 7 ), 16 ) //Azul
            );
        }
    }
    catch(Exception e){
        Log.i("Color Exception", e.toString());
        color = 0;
    }

    return color;
}
}
```

```
*****  
ListViewAdapter.java  
*****  
  
package app.telemetry;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
import android.app.Activity;  
import android.text.Html;  
import android.text.Spanned;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.webkit.WebSettings;  
import android.webkit.WebView;  
import android.widget.BaseAdapter;  
import android.widget.TextView;  
  
public class ListViewAdapter extends BaseAdapter  
{  
    public ArrayList<HashMap> list;  
    Activity activity;  
  
    public ListViewAdapter(Activity activity, ArrayList<HashMap> list) {  
        super();  
        this.activity = activity;  
        this.list = list;  
    }  
  
    @Override  
    public int getCount() {  
        // TODO Auto-generated method stub  
        return list.size();  
    }  
  
    @Override  
    public Object getItem(int position) {  
        // TODO Auto-generated method stub  
        return list.get(position);  
    }  
}
```

```
@Override
public long getItemId(int position) {
    // TODO Auto-generated method stub
    return position;
}

private class ViewHolder {
    TextView titulo;
    TextView descripcion;
    WebView webView;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    // TODO Auto-generated method stub
    ViewHolder holder;
    LayoutInflater inflater = activity.getLayoutInflater();

    if (convertView == null)
    {
        convertView = inflater.inflate(R.layout.listview_item, null);
        holder = new ViewHolder();
        holder.webView = (WebView) convertView.findViewById(R.id.frame);
        WebSettings webSettings = holder.webView.getSettings();
        holder.webView.setVerticalScrollBarEnabled(false);
        holder.webView.setHorizontalScrollBarEnabled(false);
        holder.webView.setVisibility(View.VISIBLE);
        webSettings.setLoadWithOverviewMode(true);
        webSettings.setUseWideViewPort(true);
        webSettings.setBuiltInZoomControls(false);
        webSettings.setSupportZoom(true);
        webSettings.setJavaScriptEnabled(true);
        holder.titulo = (TextView) convertView.findViewById(R.id.titulo);
        holder.descripcion = (TextView) convertView.findViewById(R.id.descripcion);
        convertView.setTag(holder);
    }
    else
    {
        holder = (ViewHolder) convertView.getTag();
    }

    HashMap map = list.get(position);
```

```
holder.webView.loadUrl("http://pnh.gob.mx/sistema_telemetria/GUI/GraficaMin.php?grupo=" + ((String) map.get(Diccionario.WEB)) + "&usuario=" + ((String) map.get(Diccionario.USUARIO)));
    holder.titulo.setText((String) map.get(Diccionario.TITULO));
    Spanned html = Html.fromHtml((String) map.get(Diccionario.DESCRIPCION));
    holder.descripcion.setText(html);

    return convertView;
}
}
```

LoginActivity.java

```
package app.telemetria;

import java.util.Iterator;
import org.json.JSONArray;
import org.json.JSONObject;
import android.animation.Animator;
import android.animation.AnimatorListenerAdapter;
import android.annotation.TargetApi;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.ContentValues;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.View;
import android.view.WindowManager;
import android.view.View.OnClickListener;
import android.view.View.OnKeyListener;
import android.widget.Button;
import android.widget.EditText;

public class LoginActivity extends Activity {
    private DataManager DBManager = null;
```

```
private Activity act;
private Network red;
private int result;
private String user, pass;
private Alerta alerta;
private EditText et_user, et_pass;
private Button iniciar;
private View mLoginFormView, mLoginStatusView;

@Override
protected void onCreate(Bundle savedInstanceState) {
    //Configurar app en pantalla completa
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    //Iniciar BD
    DBManager = new DataManager(getApplicationContext(), this);
    DBManager.open();
    //Inicializar variables
    red = new Network(getApplicationContext(), this);
    act = this;
    //Elementos graficas
    alerta = new Alerta(this);
    et_user = (EditText) this.findViewById(R.id.user);
    et_pass = (EditText) this.findViewById(R.id.password);
    iniciar = (Button) this.findViewById(R.id.login);

    getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STA
TE_ALWAYS_HIDDEN);
    //Agregar listener al boton
    iniciar.setOnClickListener(new OnClickListener() {
        public void onClick(View arg0) {
            login();
        }
    });
    //Agregar listener al boton aceptar en el campo password
    et_pass.setOnKeyListener(new OnKeyListener() {
        public boolean onKeyDown(View arg0, int keyCode, KeyEvent event) {
            if (event.getAction() == KeyEvent.ACTION_DOWN &&
keyCode == KeyEvent.KEYCODE_ENTER) {
                if (!et_user.getText().toString().equals("")) &&
!et_pass.getText().toString().equals(""))
```

```
                login();
            }
            return false;
        }
    });
    //Verificar si hay una sesion
    sesionActiva();
}

private void sesionActiva(){
    //Consultamos si hay usuario conectado
    String idsesion = DBManager.consultaDato("usuarioConectado", null);
    if(idsesion != null){
        //Ir al la actividad principal
        Intent intent = new Intent(LoginActivity.this, MainActivity.class);
        startActivity(intent);
        act.finish();
    }
}

private void login() {
    //Obtener string usuario y contraseña
    user = et_user.getText().toString();
    pass = et_pass.getText().toString();
    //Validacion
    if (!user.equals("") && !pass.equals("") && red.networkAvailable()==true){
        //Dialogo de proceso
        final ProgressDialog dialog = new ProgressDialog(act,
R.style.TransparentDialog);
        dialog.setMessage("Validando información...");
        dialog.setIndeterminate(true);
        dialog.setCancelable(false);
        dialog.setCanceledOnTouchOutside(false);
        dialog.show();
//        dialog.show(act, "", "Validando información...", true);

        Thread t = new Thread() {
            public void run() {
                //Consulta del Webservice
                Post post = new Post();
                JSONArray datos;
```

```

        datos = post.getServerData(null, Diccionario.URL +
"?clave=login&user=" + (user.replace(" ", "%20")) + "&pass=" + (pass.replace(" ", "%20")),
false);

        try {
            if (datos != null && datos.length() > 0) {
                //Obtencion del resultado
                for (int x = 0; x < datos.length(); x++) {
                    ContentValues values = new
ContentValues();

                    datos.getJSONObject(x);

                    JSONObject json_data =
json_data.keys();

                    Iterator<?> fila =

                    while (fila.hasNext()) {
                        String key = (String)

                            if(!key.equals("result"))
                                values.put(key,

                            else
                                result =

                    }
                    //Si son correctos los datos

                    if(result == 1)

                        DBManager.insertar("sesion", values);
                }

            } else {
                alerta.Mensaje("Error al consultar los
datos...");

                et_user.setText("");
                et_pass.setText("");
            }
        }
        catch (Exception e) {
            //alerta.Mensaje("Error: "+e.toString());
        }
        dialog.dismiss();
        act.runOnUiThread(new Runnable() {

```



```

public void run() {
    //Mensaje de error o cambio de
    pantalla
    if(result == 0){
        alerta.Mensaje("Error, el
        usuario y/o contraseña no son correctos");
        et_user.setText("");
        et_pass.setText("");
    }
    else{
        //Iniciar actividad principal
        Intent intent = new
        Intent(LoginActivity.this, MainActivity.class);
        startActivity(intent);
        act.finish();
        DBManager.close();
    }
}
});
}
};
t.start();
}
else{
    alerta.Mensaje("Escriba su usuario y/o contraseña");
}
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    //super.onCreateOptionsMenu(menu);
    //getMenuInflater().inflate(R.menu.login, menu);
    return true;
}

/**
 * Shows the progress UI and hides the login form.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB_MR2)
private void showProgress(final boolean show) {
    // On Honeycomb MR2 we have the ViewPropertyAnimator APIs, which allow
    // for very easy animations. If available, use these APIs to fade-in
    // the progress spinner.

```

```

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB_MR2) {
    int shortAnimTime = getResources().getInteger(
        android.R.integer.config_shortAnimTime);

    mLoginStatusView.setVisibility(View.VISIBLE);
    mLoginStatusView.animate().setDuration(shortAnimTime)
        .alpha(show ? 1 : 0)
        .setListener(new AnimatorListenerAdapter() {
            @Override
            public void onAnimationEnd(Animator
animation) {
                mLoginStatusView.setVisibility(show ?
View.VISIBLE
                : View.GONE);
            }
        });

    mLoginFormView.setVisibility(View.VISIBLE);
    mLoginFormView.animate().setDuration(shortAnimTime)
        .alpha(show ? 0 : 1)
        .setListener(new AnimatorListenerAdapter() {
            @Override
            public void onAnimationEnd(Animator
animation) {
                mLoginFormView.setVisibility(show ?
View.GONE
                : View.VISIBLE);
            }
        });
} else {
    // The ViewPropertyAnimator APIs are not available, so simply show
    // and hide the relevant UI components.
    mLoginStatusView.setVisibility(show ? View.VISIBLE : View.GONE);
    mLoginFormView.setVisibility(show ? View.GONE : View.VISIBLE);
}
}
}
}

```

MainActivity.java

```
package app.telemetry;
```

```
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Iterator;  
import java.util.Timer;  
import java.util.TimerTask;
```

```
import org.json.JSONArray;  
import org.json.JSONObject;
```

```
import android.app.Activity;  
import android.app.AlertDialog;  
import android.app.ProgressDialog;  
import android.content.ContentValues;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.support.v7.app.ActionBarActivity;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.WindowManager;  
import android.widget.AdapterView;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.AdapterView.OnItemClickListener;
```

```
public class MainActivity extends ActionBarActivity {  
    public static Activity mainActivity;  
    private DataManager DBManager = null;  
    private Activity activity;  
    private Network red;  
    private int result, resultTimer = 0;  
    private String idUsuario, valores[], matrizValores[][];  
    private ListView lista;  
    private Alerta alerta;  
    private boolean dialogAbierto = false;  
    private AlertDialog.Builder builderNotificacion;  
    private Timer timer;  
    private ArrayList<HashMap> list;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    //Configurar app en pantalla completa
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //Iniciar BD
        DBManager = new DataManager(getApplicationContext(), this);
        DBManager.open();
        //Obtener usuario logeado
        idUsuario = DBManager.consultaDato("usuarioConectado", null);
        //Obtener nombre del usuario logeado
        String nombre = DBManager.consultaDato("getNombreUsuario", null);
        setTitle(getTitle() + " - (" + nombre + ")");
        //Inicilizacion de variables globales
        activity = this;
        mainActivity = this;
        red = new Network(getApplicationContext(), this);
        alerta = new Alerta(this);
        //Cargar lista de grupos
        cargarLista();
    }
```

```
private void cargarLista(){
    //Consultar si hay grupos almacenados
    lista = (ListView) this.findViewById(R.id.lista_grupos);
    matrizValores = DBManager.consulta("getGrupos", null);
    if(matrizValores!=null){
        //Mapa
        list = new ArrayList<HashMap>();
        for(int i=0; i<matrizValores.length; i++){
            HashMap temp = new HashMap();
            temp.put(Diccionario.TITULO, matrizValores[i][0]);
            temp.put(Diccionario.DESCRIPCION, matrizValores[i][1]);
            temp.put(Diccionario.WEB, matrizValores[i][2]);
            temp.put(Diccionario.USUARIO, idUsuario);
            list.add(temp);
        }
        //Adaptador Personalizado
        ListViewAdapter adapter = new ListViewAdapter(this, list);
        lista.setAdapter(adapter);
    }
```

```
    }
    else{
        //Obtener lista de grupos
        getLista();
    }
    //Agregar listener al listView
    lista.setOnItemClickListener(new OnItemClickListener(){
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
            // TODO Auto-generated method stub
            //Iniciar la siguiente actividad
            Intent intent = new Intent(MainActivity.this, Vistas.class);
            intent.putExtra("grupo",
lista.getItemAtPosition(position).toString().substring(lista.getItemAtPosition(position).toSt
ring().indexOf("titulo=")+7, lista.getItemAtPosition(position).toString().length()-1));
            intent.putExtra("actualizar", false);
            startActivity(intent);
        }
    });
}

private void getLista(){
    //Verificamos el acceso a la red
    if (red.networkAvailable()==true){
        //Dialogo de proceso
//
        final ProgressDialog dialog = ProgressDialog.show(activity, "",
"Cargando valores", true);
        final ProgressDialog dialog = new ProgressDialog(activity,
R.style.TransparentDialog);
        dialog.setMessage("Cargando valores...");
        dialog.setIndeterminate(true);
        dialog.setCancelable(false);
        dialog.setCanceledOnTouchOutside(false);
        dialog.show();
        Thread t = new Thread() {
            public void run() {
                //Consulta del Webservice
                Post post = new Post();
                JSONArray datos;
                datos = post.getServerData(null, Diccionario.URL +
"?clave=gruposVista&user="+idUser, false);
                try {
```

```

        if (datos != null && datos.length() > 0) {
            //Obtener datos
            valores = new String[datos.length()];
            for (int x = 0; x < datos.length(); x++) {
                ContentValues values = new ContentValues();
                JSONObject json_data =
datos.getJSONObject(x);
                Iterator<?> fila =
json_data.keys();
                while (fila.hasNext()) {
                    String key = (String)
fila.next();
                    if(key.equals("result")){
                        result =
json_data.getString(key).equals("error") ? 0 : 1;
                    }
                    else
                        valores[x] =
json_data.getString(key);
                    if(!key.equals("result"))
                        values.put(key,
json_data.getString(key));
                }
                //Insertar grupos en la BD
                if(result == 1)
                    DBManager.insertar("grupos", values);
            }
        } else {
            alerta.Mensaje("Error al consultar los
datos...");
        }
    }
    catch (Exception e) {
        //alerta.Mensaje("Error: "+e.toString());
    }
    dialog.dismiss();
    activity.runOnUiThread(new Runnable() {
        public void run() {

```



```

private void verificarCompartidos(){
    try{
        //Verificamos el acceso a la red
        if (red.networkAvailable()==true){
            Thread t = new Thread() {
                public void run() {
                    //Consulta del Webservice
                    Post post = new Post();
                    JSONArray datos;
                    datos = post.getServerData(null,
Diccionario.URL + "?clave=alertaVistaCompartida&user="+idUserario, false);
                    try {
                        if (datos != null && datos.length() > 0)
{
                            //Obtencion de la respuesta
                            for (int x = 0; x < datos.length();
x++) {
                                JSONObject json_data =
                                datos.getJSONObject(x);
                                Iterator<?> fila =
                                json_data.keys();
                                while (fila.hasNext()) {
                                    String key =
                                    (String) fila.next();

                                    if(key.equals("alerta")){

                                        resultTimer = json_data.getString(key).equals("1") ? 1 : 0;
                                        }
                                    }
                                }
                                } else { }
                            }
                        catch (Exception e) {
                            //Log.i("Error ",e.toString());
                        }
                    }
                    activity.runOnUiThread(new Runnable() {
                        public void run() {
                            //Si hay notificacion abrimos el
dialog

```



```

        if(resultTimer == 1){
            notificacionDialog();
        }

        resultTimer = 0;
    }
    });
    }
};
t.start();
}
}
catch(Exception e){ }
}

private void notificacionDialog(){
    try{
        //Si el dialog no esta abierto
        if(dialogAbierto==false){
            dialogAbierto = true;
            //Creacion del dialog
            builderNotificacion = new AlertDialog.Builder(activity);
            builderNotificacion.setTitle("Notificación");
            builderNotificacion.setMessage("¡Tienes una nueva vista
compartida!");

            //Agregar botones de accion dependiendo del grupo actual
            builderNotificacion.setPositiveButton("Ver Compartidos", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // TODO Auto-generated method stub
                    //Ver compartidos
                    Intent intent = new Intent(MainActivity.this,
Vistas.class);

                    intent.putExtra("grupo", "Compartido");
                    intent.putExtra("actualizar", true);
                    startActivityForResult(intent, 1);
                    dialogAbierto = false;
                }
            });
            builderNotificacion.setNegativeButton("Cerrar", new
DialogInterface.OnClickListener() {
                @Override

```

```
        public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            dialogAbierto = false;
        }
    });
    builderNotificacion.show();
}
}
catch(Exception e){ }
}

@Override
protected void onStart() {
    super.onStart();
    //En cuanto inicie la ventana crear timer para revisar las notificaciones
    timer = new Timer();
    timer.scheduleAtFixedRate(new TimerTask(){
        @Override
        public void run() {
            // TODO Auto-generated method stub
            verificarCompartidos();
        }
    }, 0, 20000);
}

@Override
protected void onStop() {
    super.onStop();
    //Cuando pare la ventan cancelar el timer de notificaciones
    timer.cancel();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
```

```
// as you specify a parent activity in AndroidManifest.xml.
int id = item.getItemId();
//Cerrar sesion
if (id == R.id.cerrar_sesion) {
    DBManager.delete("sesion", "idusuario", idUsuario, false);
    DBManager.delete("grupos", null, null, true);
    DBManager.delete("grupo_vista", null, null, true);
    DBManager.delete("vista", null, null, true);
    DBManager.delete("valores", null, null, true);
    DBManager.delete("tema", null, null, true);
    DBManager.close();
    Intent intent = new Intent(MainActivity.this, LoginActivity.class);
    startActivity(intent);
    this.finish();
    return true;
}
//Actualizar grupos
else if(id == R.id.actualizar) {
    DBManager.delete("grupos", null, null, true);
    DBManager.delete("grupo_vista", null, null, true);
    getLista();
}
return super.onOptionsItemSelected(item);
}
}
```

Neetwork.java

```
package app.telemetry;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.InetAddress;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.util.Log;
```

```

public class Network {
    private Context context;
    private Activity activity;
    private Alerta a;
    private String msj = "";
    private String msjComplemento = "";
    private boolean result = false;
    private boolean mensaje = true;

    public Network(Context context, Activity activity){
        this.context = context;
        this.activity = activity;
        this.a = new Alerta(this.context);
    }

    public void setMsjComplemento(String texto){
        this.msjComplemento = texto;
    }

    public boolean networkAvailable() {
        ConnectivityManager connectMgr = (ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE);
        if (connectMgr != null) {
            NetworkInfo[] netInfo = connectMgr.getAllNetworkInfo();
            if (netInfo != null) {
                for (NetworkInfo net : netInfo) {
                    if (net.getState() ==
NetworkInfo.State.CONNECTED) {
                        Log.d("NETWORK", "Network
available");
                        return true;
                    }
                }
            }
        }

        if(this.mensaje == true){
            a.setText("No hay red disponible" +
(msjComplemento.equals("") ? "" : ", "+msjComplemento) );
            a.showAlerta();
        }
        Log.d("NETWORK", "No network available");
    }
}

```

```
        return false;
    }

    public void setMensajeDefault(boolean mensaje){
        this.mensaje = mensaje;
    }

    public String cmdPing() {
        String str = "";
        try {
            Process process = Runtime.getRuntime().exec("/system/bin/ping -c 8 " +
Diccionario.ip);
            BufferedReader reader = new BufferedReader(new
InputStreamReader(process.getInputStream()));
            int i;
            char[] buffer = new char[4096];
            StringBuffer output = new StringBuffer();
            while ((i = reader.read(buffer)) > 0)
                output.append(buffer, 0, i);
            reader.close();

            str = output.toString();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return str;
    }

    public void ping(){
        final ProgressDialog dialog = ProgressDialog.show(activity, "", "Prueba de
conexión con el servidor", true);
        Thread t = new Thread(){
            public void run(){
                Log.d("", "conectando...");
                try {
                    String host = Diccionario.ip;
                    InetAddress addr = null;
                    try {
                        addr = InetAddress.getByName(host);
                    } catch (Exception e) {
                        result = false;
                        msj = e.toString();
                    }
                }
            }
        };
    }
}
```

```
    }
    try {
        if(addr.isReachable(5000)) {
            result = true;
            msj = "1";
        } else {
            result = false;
            msj = "0";
        }
    } catch (Exception e) {
        result = false;
        msj = e.toString();
    }
} catch (Exception e) {
result = false;
    msj = e.toString();
}

dialog.dismiss();
activity.runOnUiThread(
    new Runnable(){
        public void run(){
            if(msj.equals("1")){
                Log.i("NETWORK","alcanzado");
            }
            else if(msj.equals("0")){
                a.setText("Error, no hay conexión con
el servidor"+(msjComplemento.equals("") ? "" : ", "+msjComplemento));

                a.showAlerta();
            }
            else{
                a.setText("Error,
"+msj+(msjComplemento.equals("") ? "" : ", "+msjComplemento));
                a.showAlerta();
            }
            msj = "";
        }
    });
};
t.start();
```

```
    }

    public String getMsj(){
        return this.msj;
    }

    public boolean getResultProcess(){
        String str = "";
        while(str==""){
            str = getMsj();
        }
        return result;
    }
}
```

Post.java

```
package app.telemetry;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import android.util.Log;

class Post {
    private InputStream is = null;
    private String respuesta = "";

    private void conectaPost(ArrayList<?> parametros, String URL) {
        ArrayList<BasicNameValuePair> nameValuePairs;
        try {
```

```
HttpClient httpClient = new DefaultHttpClient();
HttpPost httppost = new HttpPost(URL);
nameValuePairs = new ArrayList<BasicNameValuePair>();

if (parametros != null) {
    for (int i = 0; i < parametros.size() - 1; i += 2) {
        nameValuePairs.add(new BasicNameValuePair((String)parametros.get(i),
(String)parametros.get(i + 1)));
    }

    httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
}

HttpResponse response = httpClient.execute(httppost);
HttpEntity entity = response.getEntity();
is = entity.getContent();
} catch (Exception e) {
    is = null;
Log.e("log_tag", "Error in http connection " + e.toString());
} finally { }
}

private void getRespuestaPost() {
try {
    BufferedReader reader = new BufferedReader(
        new InputStreamReader(is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }

    is.close();
    respuesta = sb.toString();
    Log.e("log_tag", "Cadena JSon " + respuesta);
} catch (Exception e) {
    Log.e("log_tag", "Error converting result " + e.toString());
}
}

@SuppressWarnings("finally")
private JSONArray getJSONArray() {
    JSONArray jArray = null;
```



```
public boolean onDoubleTap(MotionEvent e) {
    this.vista.startDetalle();
    return true;
}

@Override
public void onLongPress(MotionEvent e){
    this.vista.showCompartir();
}

@Override
public boolean onSingleTapConfirmed(MotionEvent e) {
    this.vista.mensajeDialog();
    return true;
}
}
```

Vistas.java

```
package app.telemetria;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Timer;
import java.util.TimerTask;

import org.json.JSONArray;
import org.json.JSONObject;

import android.annotation.TargetApi;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
```

```
import android.support.v7.app.ActionBarActivity;
import android.util.Log;
import android.view.GestureDetector;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup.LayoutParams;
import android.view.WindowManager;
import android.widget.LinearLayout;

public class Vistas extends ActionBarActivity {
    private String idUsuario, grupo, idGrupo;
    private boolean tablet = true, dialogAbierto = false, actualizar = false, esCompartido
= false;
    private int result = 0, resultTimer = 0, x = 0;
    private LinearLayout row1, row2, contenedorRow2, contenedorRows;
    private DataManager DBManager = null;
    private Activity activity;
    private Network red;
    private Alerta alerta;
    private Dispositivo dispositivo;
    private double ln_x, ln_y;
    private AlertDialog.Builder builderNotificacion;
    private Timer timer;
    private int idVistas[], idPresionada, posPresionada;
    private ArrayList<Grafica> plots = new ArrayList<Grafica>();
    private      ArrayList<ArrayList<Long>>      serieFechas      =      new
ArrayList<ArrayList<Long>>();
    private      ArrayList<ArrayList<Double>>      serieValores      =      new
ArrayList<ArrayList<Double>>();

    @Override
    public void onCreate(Bundle savedInstanceState) {
        //Configurar app en pantalla completa
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_vistas);
        //Obtener grupo
        Intent i = getIntent();
        grupo = i.getStringExtra("grupo");
        actualizar = i.getBooleanExtra("actualizar", false);
```

```
//Inicilizar BD
    DBManager = new DataManager(getApplicationContext(), this);
    DBManager.open();
    //Obtener usuario logeado
    idUsuario = DBManager.consultaDato("usuarioConectado", null);
    //Obtener nombre del usuario logeado
    String nombre = DBManager.consultaDato("getNombreUsuario", null);
    setTitle(getTitle() + " - (" + nombre + ")");
    //IdGrupo
    idGrupo = DBManager.consultaDato("getIdGrupo", new HashMap<String,
String>(){{put("<grupo>", grupo);}});
    //Inicilizacion de variables globales
    activity = this;
    red = new Network(getApplicationContext(), this);
    alerta = new Alerta(this);
    //Contenedores de las graficas
    contenedorRows = (LinearLayout) this.findViewById(R.id.contenedorRows);
    contenedorRow2 = (LinearLayout) this.findViewById(R.id.contenedorRow2);
    row1 = (LinearLayout) this.findViewById(R.id.row1);
    row2 = (LinearLayout) this.findViewById(R.id.row2);
    //Verificacion tamaño del dispositivo y ajuste
    tablet = getResources().getBoolean(R.bool.isTablet);
    dispositivo = new Dispositivo(this);
    int divisor = 2;
    double porcentaje = 0.08;
    if(!tablet){
        divisor = 1;
        porcentaje = 0.12;
        contenedorRows.removeView(contenedorRow2);
    }
    //Ajuste de dimensiones
    ln_x = (dispositivo.getWidth() - (5*divisor)) / divisor;
    ln_y = (dispositivo.getHeight() - (dispositivo.getHeight()*porcentaje)) / divisor;
    //Verificamos si es grupo es el compartido
    esCompartido = grupo.equals("Compartido");
    //Cargar vistas
    cargarVistas();
}

private void cargarVistas(){
    //Si se envia el forzado de actualizacion
    if(actualizar==true)
        borraDatos();
}
```

```

//Consulta si hay vistas almacenadas
    final String vistas[][] = DBManager.consulta("getVistas", new
HashMap<String, String>(){{put("<grupo>", grupo)}});
    if(vistas!=null){
        //Recorrer vistas, posiciones:
idvista,idtema,tipo_vista,sensor,vista,max,min,mensaje,enviado
        for(x=0; x<vistas.length; x++){
            //Obtencion del tema, posiciones: idtema, fondo_panel,
borde_panel, fondo_grid, grid_ejex, grid_ejey, etiquetas_ejex, etiquetas_ejey, borde_serie,
fondo_serie, color_titulo, ancho_barra, ancho_linea
            String temas[][] = DBManager.consulta("getTema", new
HashMap<String, String>(){{put("<tema>", vistas[x][1])}});
            String tema[] = temas[0];
            //Obtencion de valores, posiciones: valor, hora
            String valores[][] = DBManager.consulta("getValores", new
HashMap<String, String>(){{put("<vista>", vistas[x][0])}});
            //Grafica
            Grafica graph = new Grafica(activity, vistas[x][4],
Integer.parseInt(vistas[x][0]));
            graph.setRangoY(Double.parseDouble(vistas[x][6]),
Double.parseDouble(vistas[x][5]), Math.round(((Double.parseDouble(vistas[x][5])-
Double.parseDouble(vistas[x][6]))/4) * 100) /100);
            graph.setColorPanel(tema[1], tema[2]);
            graph.setColorGrid(tema[4], tema[5], tema[3]);
            graph.setColorEtiquetasXY(tema[6], tema[7]);
            if(vistas[x][2].equals("Barras"))
                graph.setBordeyRellenoBarra(tema[8], tema[9],
Integer.parseInt(tema[11]));
            else
                graph.setBordeyRellenoSerie(tema[8], tema[9],
Integer.parseInt(tema[12]), vistas[x][2].equals("Areas")?true:false);
            graph.setColorTitulo(tema[10]);
            //Agregar grafica al array
            plots.add(graph);
            //Agregar series a los array
            ArrayList<Long> fechas = new ArrayList<Long>();
            ArrayList<Double> valor = new ArrayList<Double>();
            for(int i=0; i<valores.length; i++){
                valor.add(Double.parseDouble(valores[i][0]));

                fechas.add(Long.parseLong(valores[i][1]));
            }
            //Agregar quien lo envio solo en grupo compartido

```

```
        if(esCompartido)
            graph.setPie("Enviado por: "+vistas[x][8]);
        //Dibujar grafica
        graph.addSerie(fechas, valor, vistas[x][3], vistas[x][2]);
    }
    //Agregar graficas a los contenedores
    agregarGraficas();
}
else{
    //Consultar Webservice
    getVistas();
}
}

private void getVistas(){
    //Verificamos el acceso a la red
    if (red.networkAvailable()==true){
        //Dialog cargando
//        final ProgressDialog dialog = ProgressDialog.show(activity, "",
"Cargando vistas", true);
        final ProgressDialog dialog = new ProgressDialog(activity,
R.style.TransparentDialog);
        dialog.setMessage("Cargando vistas...");
        dialog.setCancelable(false);
        dialog.setCanceledOnTouchOutside(false);
        dialog.setIndeterminate(true);
        dialog.show();
        Thread t = new Thread() {
            public void run() {
                //Consultamos el Webservice
                Post post = new Post();
                JSONArray datos;
                datos = post.getServerData(null, Diccionario.URL +
"?clave=series&grupo="+grupo+ "&usuario="+idUsuario, false);
                try {
                    //Hay respuesta
                    if (datos != null && datos.length() > 0) {
                        //Recorrer respuesta JSON
                        idVistas = new int[datos.length()];
                        for (int x = 0; x < datos.length(); x++) {
                            //valores para insertar
                            ContentValues valuesTema = new
ContentValues(),
```

```

ContentValues(),
new ContentValues(),
ContentValues();

datos.getJSONObject(x);

exitoso o hubo un error en la respuesta

json_data.keys();

(String) fila.next();

    if(key.equals("result")){

json_data.getString(key).equals("error") ? 0 : 1;

en la respuesta

json_data.getInt("idvista");

json_data.getDouble("min");

json_data.getDouble("max");

json_data.getString("vista"),

json_data.getString("sensor"),

json_data.getString("latitud"),

valuesValores = new
valuesGrupoVista =
valuesVista = new

//Posicion en el JSON
JSONObject json_data =
if(x==0){
//Obtenermos si fue
Iterator<?> fila =
while (fila.hasNext()) {
String key =
result =
}
}
//Terminamos en caso de error
if(result==0)
break;

//Valores recibidos en el JSON
idVistas[x] =
double auxMin =
double auxMax =
String titulo =
serie =
latitud =

```

| | | |
|--|----------------------|---|
| json_data.getString("longitud"), | longitud | = |
| json_data.getString("estacion"), | estacion | = |
| json_data.getString("sistema"), | sistema | = |
| json_data.getString("unidad"), | unidad | = |
| json_data.getString("enviado"), | enviado | = |
| json_data.getString("mensaje"), | mensaje | = |
| json_data.getString("tipo_vista"), | tipo_vista | = |
| json_data.getString("fondo_panel"), | fondo_panel | = |
| json_data.getString("borde_panel"), | borde_panel | = |
| json_data.getString("fondo_grid"), | fondo_grid | = |
| json_data.getString("grid_ejex"), | grid_ejex | = |
| json_data.getString("grid_ejey"), | grid_ejey | = |
| json_data.getString("etiquetas_ejex"), | etiquetas_ejex | = |
| json_data.getString("etiquetas_ejey"), | etiquetas_ejey | = |
| json_data.getString("borde_serie"), | borde_serie | = |
| json_data.getString("fondo_serie"), | fondo_serie | = |
| json_data.getString("color_titulo"); | color_titulo | = |
| json_data.getString("idtema"); | final String idtema | = |
| json_data.getInt("ancho_barra"); | int ancho_barra | = |
| json_data.getInt("ancho_linea"); | int ancho_linea | = |
| json_data.getString("idvista"); | final String idvista | = |


```

int idgrupo =
json_data.getInt("idgrupo");

//Insercion del tema si no existe
String existeTema =
DBManager.consultaDato("existeTema", new HashMap<String, String>(){put("<tema>",
idtema);});

if(existeTema==null){

valuesTema.put("fondo_panel", fondo_panel);

valuesTema.put("borde_panel", borde_panel);

valuesTema.put("fondo_grid", fondo_grid);
valuesTema.put("grid_ejex",
grid_ejex);
valuesTema.put("grid_ejey",
grid_ejey);

valuesTema.put("etiquetas_ejex", etiquetas_ejex);
valuesTema.put("etiquetas_ejey", etiquetas_ejey);

valuesTema.put("borde_serie", borde_serie);
valuesTema.put("fondo_serie", fondo_serie);
valuesTema.put("color_titulo", color_titulo);
valuesTema.put("ancho_barra", ancho_barra);
valuesTema.put("ancho_linea", ancho_linea);

valuesTema.put("idtema",
idtema);

int tema =
DBManager.insertar("tema", valuesTema);

Log.i("tema creado",
""+tema);

}
//Calculo rango maximo y minimo en
la grafica
if(auxMin==auxMax &&
auxMin==0){
auxMin = 0;
}

```

```

        auxMax += 4;
    }else{
        auxMin -= 4;
        auxMax += 4;
    }
    //Para las barras por default el
    minimo es 0

    if(tipo_vista.equals("Barras")){
        auxMin = 0;
    }
    //Configuracion de la grafica
    Grafica graph = new Grafica(activity,
    titulo, idVistas[x]);
    graph.setRangoY(auxMin,    auxMax,
    Math.round(((auxMax-auxMin)/4) * 100) /100);
    graph.setColorPanel(fondo_panel,
    borde_panel);
    graph.setColorGrid(grid_ejex,    grid_ejey,
    fondo_grid);
    graph.setColorEtiquetasXY(etiquetas_ejex, etiquetas_ejey);
    if(tipo_vista.equals("Barras"))
    graph.setBordeyRellenoBarra(borde_serie, fondo_serie, ancho_barra);
    else
    graph.setBordeyRellenoSerie(borde_serie,    fondo_serie,    ancho_linea,
    tipo_vista.equals("Areas")?true:false);
    graph.setColorTitulo(color_titulo);
    //Agregar grafica al array
    plots.add(graph);
    //Inserion de la vista en la BD
    valuesVista.put("latitud", latitud);
    valuesVista.put("longitud", longitud);
    valuesVista.put("estacion", estacion);
    valuesVista.put("sistema", sistema);
    valuesVista.put("unidad", unidad);
    valuesVista.put("idvista", idvista);
    valuesVista.put("idtema", idtema);
    valuesVista.put("tipo_vista",
    tipo_vista);
    valuesVista.put("sensor", serie);
    valuesVista.put("vista", titulo);

```

```

mensaje);
                                valuesVista.put("mensaje",
                                valuesVista.put("enviado", enviado);
                                //valuesVista.put("idgrupo",
idgrupo);
                                valuesVista.put("max", auxMax);
                                valuesVista.put("min", auxMin);
                                //Si ya existe la vista en otro grupo la
reemplazamo junto con los valores
                                String existeVista =
DBManager.consultaDato("existeVista", new HashMap<String, String>(){put("<vista>",
idvista)}});
                                if(existeVista!=null){
                                DBManager.delete("vista",
                                DBManager.delete("valores",
                                }
                                int vista =
DBManager.insertar("vista", valuesVista);
                                Log.i("vista creada", ""+vista);
                                valuesGrupoVista.put("idvista",
                                valuesGrupoVista.put("idgrupo",
                                int grupoVista =
DBManager.insertar("grupo_vista", valuesGrupoVista);
                                Log.i("GrupoVista creada",
                                ""+grupoVista);
                                //Recorrer valores de la serie
                                JSONArray temp = new
JSONArray(json_data.getString("serie"));
                                ArrayList<Long> fechas = new
ArrayList<Long>();
                                ArrayList<Double> valores = new
ArrayList<Double>();
                                for(int i=0; i<temp.length(); i++){
                                JSONObject valorTemp =
temp.getJSONObject(i);
                                Double valor =
valorTemp.getDouble("valor");
                                long fecha =
fechaHoraToMilisegundos(valorTemp.getString("tstamp"));

```

```

        fechas.add(fecha);
        valores.add(valor);

        //Insercion de valores de la serie en la
BD
        valoresValores.put("idvista", idvista);
        valoresValores.put("valor", valor);
        valoresValores.put("hora", fecha);
        int idvalor =
DBManager.insertar("valores", valoresValores);
        Log.i("valor creado",
        ""+idvalor);
    }
    //Agregar series X y Y al array
    serieValores.add(x, valores);
    serieFechas.add(x, fechas);
    //Agregar quien lo envio solo en grupo
compartido
    if(esCompartido)
        graph.setPie("Eviado: "+enviado);
    //Dibujar grafica
    graph.addSerie(fechas, valores, serie,
tipo_vista);
    }
    } else {
        alerta.Mensaje("Error al consultar los
datos...");
    }
}
catch (Exception e) {
    //alerta.Mensaje("Error: "+e.toString());
}
dialog.dismiss();
activity.runOnUiThread(new Runnable() {
    public void run() {
        //Resultado erroneo
        if(result == 0){
            alerta.Mensaje(esCompartido ?
"No hay vistas compartidas" : "Error, este grupo no tiene vistas");
        }
        else{

```

```

//Agregar graficas a los
contenedores
        agregarGraficas();
    }
    result = 0;
}
});
}
};
t.start();
}
}

private void agregarGraficas(){
    //Recorrer array de graficas
    for(int i=0; i<plots.size(); i++){
        //Creacion del contenedor individual
        LinearLayout ln = new LinearLayout(activity);
        LayoutParams lnLayoutParams = new
LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT);
        ln.setLayoutParams(lnLayoutParams);
        LayoutParams params = ln.getLayoutParams();
        //Alto y ancho para cada vista
        params.height = (int) ln_y;
        params.width = (int) ln_x;
        //Agregar grafica al contenedor
        ln.addView(plots.get(i).getGrafica());
        //Agregar listener para compartir vista
        final int temp = i;
        //Listener gestor de eventos
        final TabListener gestureDoubleTap = new TabListener(this);
        final GestureDetector gestureDetector = new GestureDetector(activity,
gestureDoubleTap);
        plots.get(i).getGrafica().setOnTouchListener(new View.OnTouchListener()
{
            @Override
            public boolean onTouch(View view, MotionEvent motionEvent) {
                idPresionada = plots.get(temp).getIdGrafica();
                posPresionada = temp;
                gestureDetector.onTouchEvent(motionEvent);
                return true;
            }
        }
    }
}

```

```
});  
//Si no es tablet agregamos 1 vista por pantalla  
if(!tablet){  
    ln.setPadding(4, 4, 4, 4);  
    row1.addView(ln);  
}  
//Si es tablet agregamos 4 vistas por pantalla  
else{  
    if((i % 4) < 2){  
        ln.setPadding(8, 10, 8, 0);  
        row1.addView(ln);  
    }  
    else{  
        ln.setPadding(8, 8, 8, 8);  
        row2.addView(ln);  
    }  
}  
}  
}  
  
public void startDetalle(){  
    Intent intent = new Intent(activity, DetalleActivity.class);  
    intent.putExtra("idVista", String.valueOf(idPresionada));  
    intent.putExtra("compartido", esCompartido);  
    activity.startActivityForResult(intent, 1);  
}  
  
@TargetApi(Build.VERSION_CODES.HONEYCOMB)  
public void mensajeDialog(){  
    if(esCompartido){  
        String texto = DBManager.consultaDato("getMensajeVista", new  
HashMap<String, String>(){put("<vista>", idPresionada+"");});  
        Dialog dialog = new Dialog(this);  
        dialog.MensajeDialog(texto);  
    }  
}  
  
public void showCompartir(){  
    Dialog dialog = new Dialog(activity);  
    dialog.showCompartir("Compartir", vista  
\ ""+plots.get(posPresionada).getVistaNombre()+ "\",  
plots.get(posPresionada).getIdGrafica(), idUsuario);
```

```
}

private long fechaHoraToMilisegundos(String times){
    String fechaHora[], fecha[], hora[];
    //Separar hora y fecha
    fechaHora = times.split(" ");
    fecha = fechaHora[0].split("-");
    hora = fechaHora[1].split(":");
    //Convertir a date y conversion a milisegundos
    Calendar calendar = Calendar.getInstance();
    calendar.set(Integer.parseInt(fecha[0]), Integer.parseInt(fecha[1])-1,
Integer.parseInt(fecha[2]), Integer.parseInt(hora[0]), Integer.parseInt(hora[1]),
Integer.parseInt(hora[2]));
    return calendar.getTimeInMillis();
}

private void borraDatos(){
    //Consulta de vistas y temas
    String vistas[][] = DBManager.consulta("getVistasDelete", new HashMap<String,
String>(){put("<grupo>", grupo)});
    String temas[][] = DBManager.consulta("getTemasDelete", new HashMap<String,
String>(){put("<grupo>", grupo)});
    //Recorrer vistas y removerlas
    if(vistas!=null)
    for(int i=0; i<vistas.length; i++){
        DBManager.deleteMul("grupo_vista", "idvista="+vistas[i][0]+ " and
idgrupo="+idGrupo);
        DBManager.delete("vista", "idvista", vistas[i][0], false);
        DBManager.delete("valores", "idvista", vistas[i][0], false);
    }

    //Recorrer temas y removerlos si no son usados
    if(temas!=null)
    for(int i=0; i<temas.length; i++){
        final String tema = temas[i][0];
        String usado = DBManager.consultaDato("getVistaUsaTema", new HashMap<String,
String>(){put("<tema>", tema)});
        if(usado==null || usado.equals("0")){
            DBManager.delete("tema", "idtema", tema, false);
        }
    }
}
}
```



```

        }
    }
    } else { }
}
catch (Exception e) {
    Log.i("Error ",e.toString());
}
activity.runOnUiThread(new Runnable() {
    public void run() {
        //Si hay notificacion abrimos el
        dialog

        if(resultTimer == 1){
            notificacionDialog();
        }

        resultTimer = 0;
    }
});
    }
};
t.start();
}
}
catch(Exception e){ }
}
private void notificacionDialog(){
    try{
        //Si el dialog no esta abierto
        if(dialogAbierto==false){
            dialogAbierto = true;
            //Creacion del dialog
            builderNotificacion = new AlertDialog.Builder(activity);
            builderNotificacion.setTitle("Notificación");
            builderNotificacion.setMessage("¡Tienes una nueva vista
compartida!");

            //Agregar botones de accion dependiendo del grupo actual
            builderNotificacion.setPositiveButton(esCompartido
"Actualizar" : "Ver Compartidos", new DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // TODO Auto-generated method stub
                    //Actualizar vistas

```

```
        grupo = esCompartido ? grupo : "Compartido";
        actualizarVistas();
        dialogAbierto = false;
    }
});
builderNotificacion.setNegativeButton("Cerrar", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        // TODO Auto-generated method stub
        dialogAbierto = false;
    }
});
builderNotificacion.show();
}
}
catch(Exception e){ }
}

@Override
protected void onStart() {
    super.onStart();
    //En cuanto inicie la ventana crear timer para revisar las notificaciones
    timer = new Timer();
    timer.scheduleAtFixedRate(new TimerTask(){
        @Override
        public void run() {
            // TODO Auto-generated method stub
            verificarCompartidos();
        }
    }, 0, 20000);
}
@Override
protected void onStop() {
    super.onStop();
    //Cuando pare la ventan cancelar el timer de notificaciones
    timer.cancel();
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.vistas, menu);
    return true;
}
```

```
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    //Cierre de sesion
    if (id == R.id.cerrar_sesion) {
        DBManager.delete("sesion", "idusuario", idUsuario, false);
        DBManager.delete("grupos", null, null, true);
        DBManager.delete("grupo_vista", null, null, true);
        DBManager.delete("vista", null, null, true);
        DBManager.delete("valores", null, null, true);
        DBManager.delete("tema", null, null, true);
        DBManager.close();
        MainActivity.mainActivity.finish();
        this.finish();
        return true;
    }
    //Actualizar vistas
    else if(id == R.id.actualizar) {
        actualizarVistas();
    }
    return super.onOptionsItemSelected(item);
}

protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    if (requestCode == 1) {
        if(resultCode == RESULT_OK){
            esCompartido = data.getBooleanExtra("compartido", false);
            if(esCompartido){
                grupo = "Compartido";
                actualizarVistas();
            }
        }
        if (resultCode == RESULT_CANCELED) {
            //Write your code if there's no result
        }
    }
}
}
```