

# Casos de aplicaciones empresariales de Web Mapping y SIG Libre.

Lewis Clark <sup>(1)</sup>, Javier Sánchez Ramírez <sup>(2)</sup>

- (1) Departamento Spatial Solutions, Genasys II Spain, S.A.U., C/ Ventura de la Vega, 5. 28014 Madrid, [lclark@genasys.com](mailto:lclark@genasys.com)
- (2) Departamento Spatial Solutions, Genasys II Spain, S.A.U., C/ Ventura de la Vega, 5. 28014 Madrid, [jsanchez@genasys.com](mailto:jsanchez@genasys.com)

## RESUMEN

*La incorporación Software Libre para Geomática (FOSS4G), para la explotación de Información Geoespacial en Sistemas de Información (SI) empresariales es una tendencia inexorable. Aunque estas tecnologías se están difundiendo con rapidez en entornos de empresas especializadas, Universidades, Administraciones Públicas y Centros Tecnológicos, todavía es algo incipiente en grandes empresas, especialmente en aquellas no directamente relacionadas con las tecnologías de los SIG.*

*El objetivo de esta presentación será mostrar cómo se está consiguiendo introducir el software de SIG libre en el mundo empresarial, con tres casos de éxito. El primero es un desarrollo tradicional para una compañía tipo 'utility' donde el cliente define una funcionalidad y contrata su desarrollo. El segundo es un modelo de contratación de servicios. La tercera es una aplicación para la administración pública. En los tres casos, el uso de software libre ha permitido ofrecer soluciones exitosas para los requerimientos de los clientes (tanto funcionales como de rendimiento), y óptimas en coste.*

**Palabras clave:** SIG, Software libre, MapFish, MapServer, GeoWebCache, PostGIS, OpenLayers, MapFish, Ubuntu, MapInfo

## **ABSTRACT**

*The use of Open Source GIS Software in the information systems mix of private sector companies is without doubt a growing trend. However, the speed of advance does not match that of universities, the public sector and some small specialized technology companies.*

*The objective of this presentation is give some examples of how Open Source GIS software is starting to be used in major companies. The first case is a utility company where the project has followed a fairly conventional path of development whereby functionality is defined, developed and then installed at the client's data-centre. The second case is a hosted solution where the client pays for the service annually. The third example consists of a regional government project which is included here due to its technical similarity to the other two projects.*

*In all cases the use of open source software both met the user requirements and reduced the overall project costs.*

**Key words:** *SIG, Software libre, MapFish, MapServer, GeoWebCache, PostGIS, OpenLayers, MapFish, Ubuntu, MapInfo*

## **INTRODUCCIÓN**

En este artículo se describe dos aplicaciones de Web Mapping desarrolladas para empresas multinacionales españolas y una aplicación desarrollada para la administración pública. Los tres casos son muy diferentes, tanto en la forma de vender como la forma de realizar los proyectos aunque todos comparten la misma base de software GIS.

En el primer caso el software ha llegado a formar parte de una mezcla compleja de sistemas espaciales que incluye Smallworld, Integraph y MapInfo. Cada sistema realiza una serie de tareas específicos, desarrollados a medida a lo largo de varios años con su correspondiente coste de inversión.

En el segundo caso el sistema vive aislado de los sistemas de información del cliente en modo *hosting*. El cliente, una de las empresas líderes en seguridad, contrata un servicio y solo requiere un acceso Internet para usar el sistema.

En el tercer caso se instaló en el centro de datos del cliente en modo *"hosting"*.

En los tres casos, para el cliente, el hecho de usar software Open Source es irrelevante. Sin embargo, para una empresa consultora como Genasys, es fundamental por las ventajas de coste y funcionalidad.

## CASO 1: REDES DE TELECOMUNICACIONES

### Descripción

Este ejemplo trata de una aplicación Web de consulta para una red de telecomunicaciones. La empresa había sido cliente durante varios años, realizando proyectos GIS con software comercial.

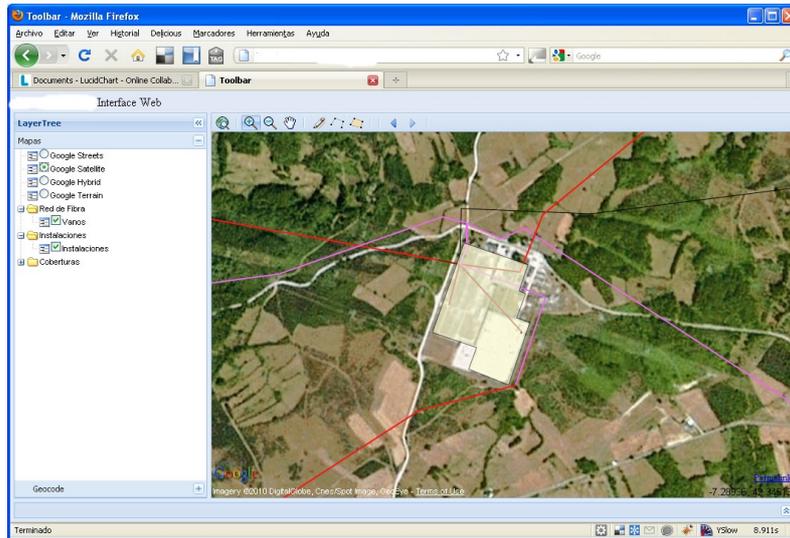


Figura 1: Aplicación Web – Datos de Red.

El siguiente diagrama refleja la arquitectura inicial del cliente, con una mezcla de diferentes sistemas comerciales, cada de lo cuales llevaba una inversión importante de años-hombre en la customización en función de las requerimientos específicos de la empresa.

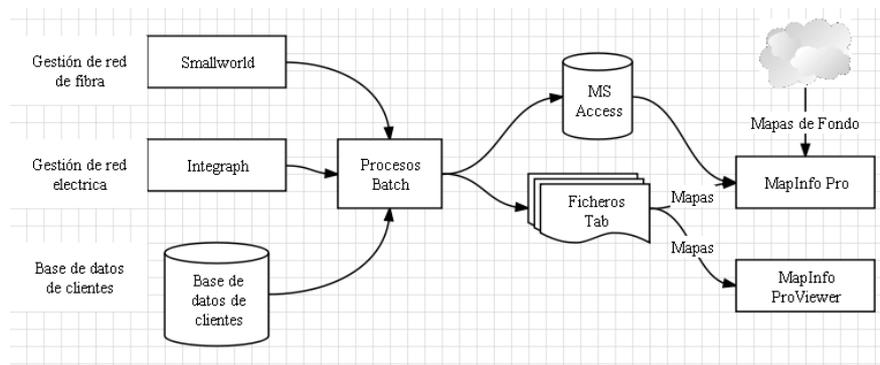


Figura 2: Arquitectura global.

Para la aplicación Web se montó dos servicios WMS con MapServer, uno para dibujar coberturas desde ficheros jpg georeferenciados y otro para los elementos de red desde mapas en formato MapInfo (TAB).

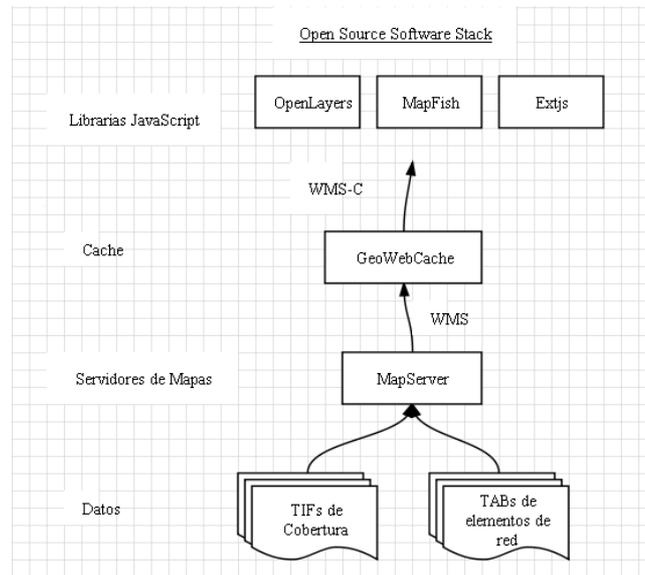


Figura 3: Software Stack - Open Source GIS.

Las coberturas, que son datos estáticos, pasaron por un proceso de conversión previa mientras los datos de elementos de red, que cambian constantemente, se leen de MapServer sin conversión.

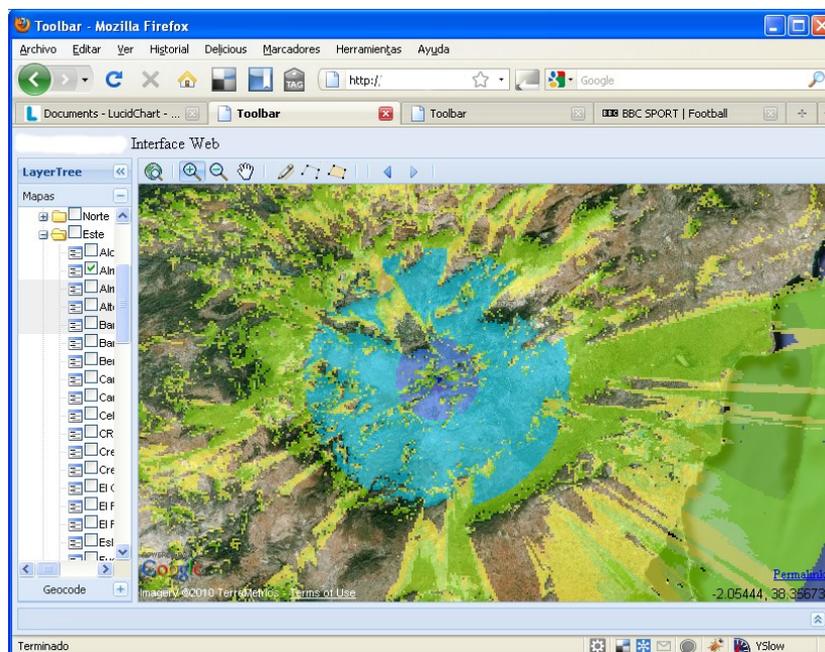


Figura 4: Gestión de coberturas..

La conversión de ficheros de JPG a TIF se realizó en *batch* con la herramienta “*mogrify*” y aunque el *mapfile* de MapServer lleva el nombre de fichero TIF, se lee datos de georeferencia desde un fichero .TAB.

```
LAYER
NAME "cobertura_31"
CONNECTIONTYPE OGR
DATA "/home/genasys/mapserv1/maps_tif/cobertura_31.tif"
STATUS ON
TYPE RASTER
PROJECTION
  "init=epsg:23030"
END
OPACITY 60
CLASSITEM "[pixel]"
OFFSITE 255 255 255
END
```

```
!table
!version 300
!charset WindowsLatin1

Definition Table
File "cobertura_31.tif"
Type "RASTER"
(432966.3042415888,4778791.2963154241) (0,0) Label "Pt 1",
(752508.48445840064,4778791.2963154241) (636,0) Label "Pt 2",
(432966.3042415888,4554207.4055026649) (0,447) Label "Pt 3"
CoordSys Earth Projection 8, 28, "m", -3, 0, 0.9996, 500000, 0
Units "m"
```

Figura 5. Coberturas – MapServer mapfile + TAB file.

### Interfaz de usuario

El árbol de capas que se ve en la parte izquierda de la pantalla en *Figura 4* tiene más de 200 nodos y no ha dado problemas aunque es un valor muy encima del valor máximo recomendado en los foros de OpenLayers.

Dicho árbol se genera dinámicamente con un *script* PHP en el servidor para facilitar el mantenimiento de la aplicación.

## CASO 2: SEGUIMIENTO DE RECURSOS

### Descripción

Este ejemplo trata de una aplicación de seguimiento de recursos, ubicado en modo *hosting*, que utiliza los siguientes paquetes de software:

- OpenLayers + MapFish
- GeoWebCache
- MapServer
- MapFish Server
- PostGIS
- Ubuntu

Los recursos móviles que se requiere localizar mandan mensajes SMS al servidor que registra los datos en Postgres. La aplicación Web lee las posiciones y refresca automáticamente sus ubicaciones en el mapa.

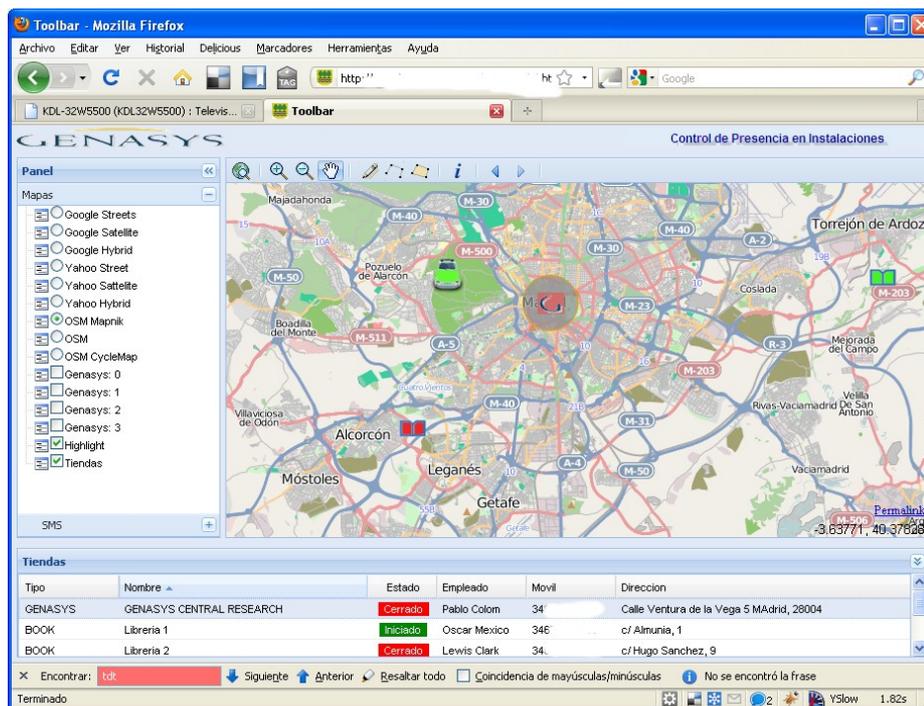


Figura 6: Gestión de recursos – Interfaz Web.

Pasos previos para poder localizar un recurso.

- Dar de alta el móvil en la base de datos.
- Instalar una aplicación Java en el móvil.

La aplicación en el móvil se encarga de mandar los SMS automáticamente, de tal manera que el usuario no tiene que escribir manualmente el mensaje.

Formas de localizar el móvil:

- Con etiquetas RFID en una ubicación fija.
- Con GPS o localización por celda.

En el primer caso la posición de las etiquetas se da de alta en la base de datos de tal manera que cada etiqueta tiene su propio identificador mientras en el segundo caso el móvil utiliza los servicios de localización ya existentes en el móvil o en la red del operador.

## Arquitectura

En el lado servidor se ha desarrollado una aplicación PHP para recibir los SMS, escribir en la base de datos y devolver la respuesta al móvil. Los mensajes pasan por el GPPm (Genasys Positioning Platform) que puede añadir la posición del móvil.

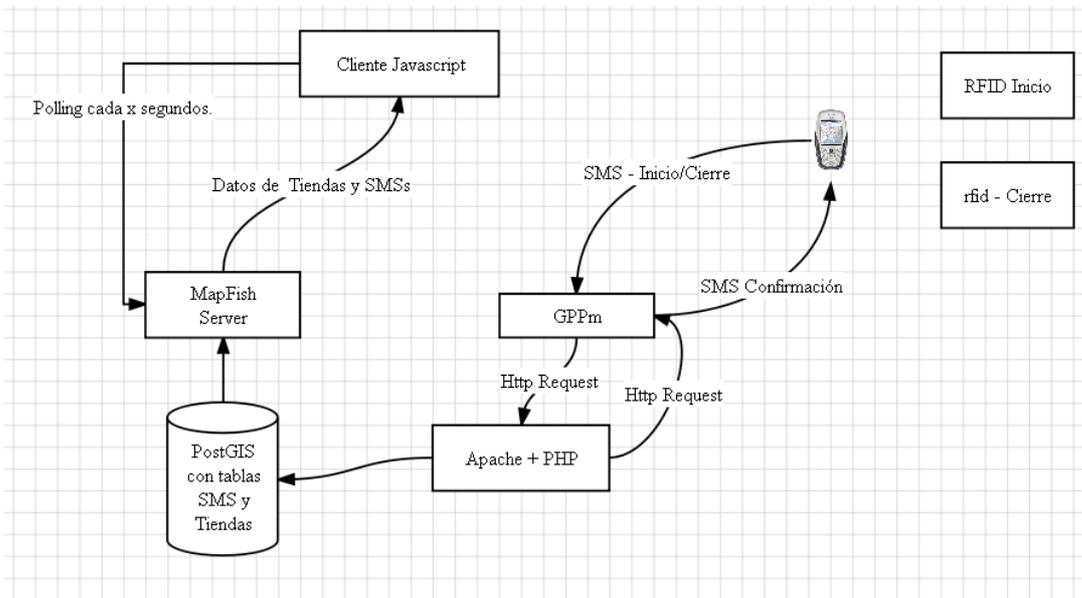


Figura 7: Arquitectura.

MapFish Server, versión 1.2 proporciona el API que el cliente *JavaScript* llama cada 5 segundos para mantener las posiciones actualizadas en el *display*.

El flujo de información empieza con un SMS en el móvil. El GPP recibe el mensaje y lo convierte en una petición HTTP GET. El URL de la aplicación que recibe el GET estaba previamente dado de alta en el GPP y asociado con una palabra clave del SMS.

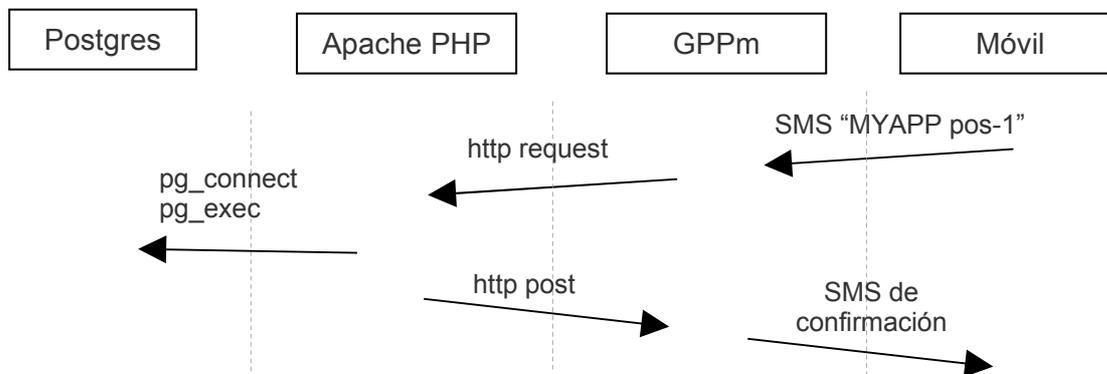


Figura 8: Flujo de mensajes.

Por ejemplo:

- El contenido del SMS empieza con el texto "MYAPP" y va a un número corto como "01234".
- El GPPm busca el texto en su lista de aplicaciones y encuentra una aplicación con el url <http://myserver.com/gpp.php>.
- El HTTP GET que manda el GPP incluye el número del móvil de origen, un identificador único del mensaje y otros parámetros que puede incluir la posición.
- El *script* de PHP lee el contenido del mensaje y otros parámetros de la petición y los escribe en la base de datos.
- Para cerrar el bucle, desde PHP se lanza un HTTP POST con un XML que ejecuta la función "sendSMS" del GPP.

```

<GPPR> <FUNCTION type="M_IT">sendSMS</FUNCTION>
<AUTH>
  <CLIENT>demopresencia</CLIENT>
  <CLI_PASSWD>demopresencia</CLI_PASSWD>
  <USER_LOGIN>demouser</USER_LOGIN>
  <USER_PASSWD>XXXX</USER_PASSWD>
</AUTH>
<PARAMETERS idtype="MSISDN">
  <ATTRIBUTE name="MEMBERSID">
    <OBJECT>
      <CLASS>VECTOR</CLASS>
      <CONTENTS>
        <ATTRIBUTE name="MEMBERID">
          <STRING val="sms_mo_id"/>
        </ATTRIBUTE>
      </CONTENTS>
    </OBJECT>
  </ATTRIBUTE>
  <ATTRIBUTE name="TEXT">
    <STRING val="Prueba"/>
  </ATTRIBUTE>
</PARAMETERS>
</GPPR>

```

Figura 9: Protocolo GPP para mandar SMS.interface

## Interfaz de usuario

El cliente consiste en un interfaz HTML/JavaScript basado en MapFish. En el siguiente ejemplo el panel inferior (south) tiene una lista de los puntos de control mientras el panel a la izquierda (west) tiene dos regiones con una lista de mapas y una lista de todos los mensajes SMS que entran. Al recibir un nuevo mensaje, se centra el mapa en el icono que representa el móvil o el punto de control. Si el punto está definido como tipo “edificio”, la aplicación ajusta el nivel de zoom para poder visualizar el icono encima del plano del edificio.

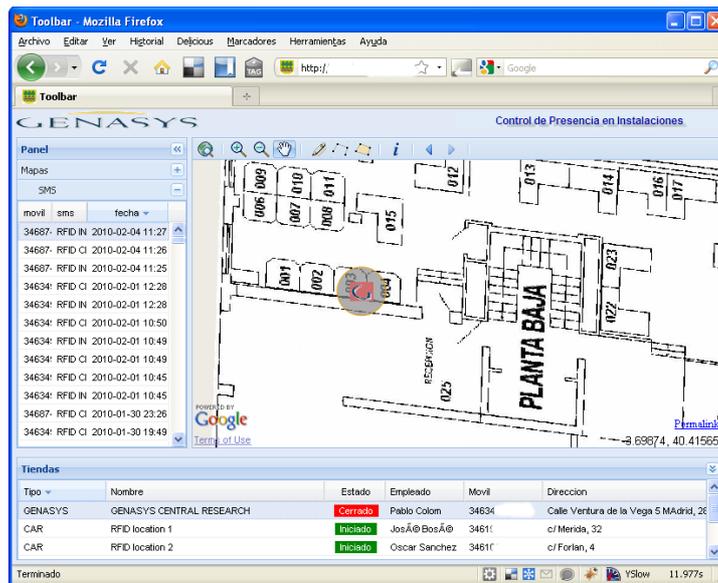


Figura 10: Interfaz Web – Nivel planta de edificio..

## Servidor de mapas de fondo

Para servir los planos y otros mapas de fondo se utiliza la combinación de MapServer/Apache con GeoWebCache/Tomcat. Los planos son ficheros TIF.

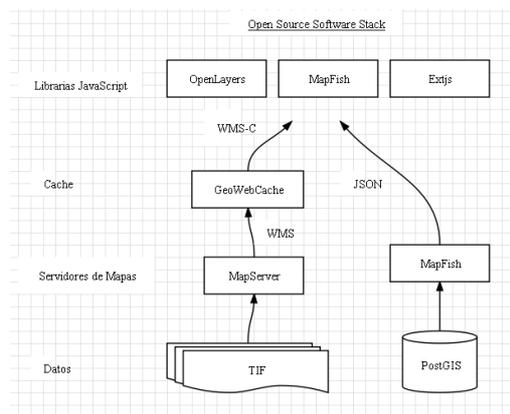


Figura 11: Software Stack.

## CASO 3: PUBLICACIÓN DE DATOS DE RECURSOS HIDROLÓGICOS

### Descripción

Este ejemplo trata de una aplicación de consulta y cálculo de cuencas hidrológicas. El proyecto consistía en construir una aplicación Web y modificar un software ya existente para publicar datos.

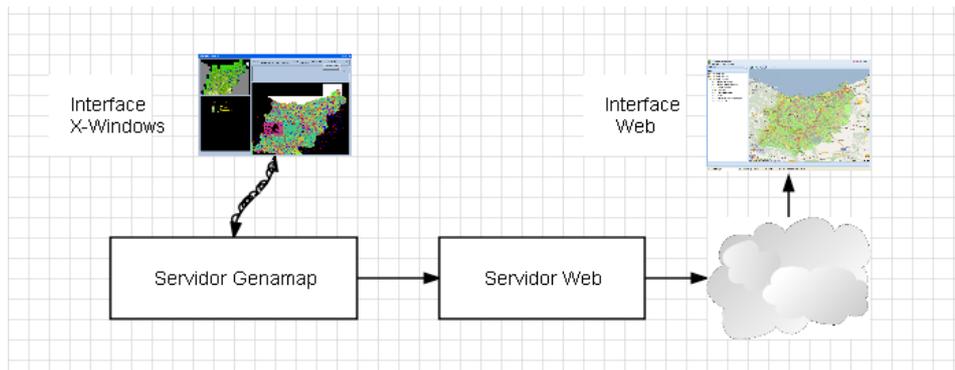


Figura 12: *Publicación de datos.*

La aplicación ya existente constaba de un sistema de gestión de información hidrológica que permitía consultar datos y ejecutar varios cálculos. El software estaba desarrollado sobre el GIS propietario Genamap.

Un nuevo menú en el interfaz X-Windows añade la opción de publicar datos, tanto mapas como datos alfanuméricos.

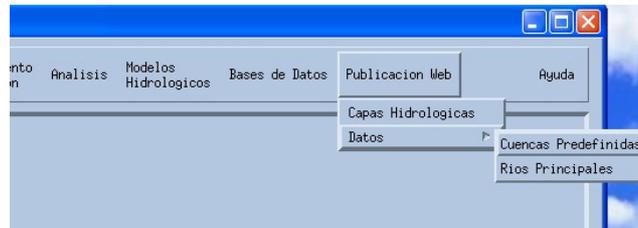


Figura 13: Aplicación Genamap – *Publicar.*

Los datos publicados se insertan en una base de datos PostGIS o en el caso de los mapas de tipos vegetación o suelo, se copia en formato *shape* para ser renderizado por MapServer. El software base incluye MapFish Server y GeoWebCache mientras el cliente se basaba en MapFish Cliente con Extjs.

### Mapas Base

Los mapas base de la aplicación aprovechan de los servicios WMS de la Diputación de Gipuzkoa que dibujan sus propios datos encima de Google maps con los mismos estilos para que aparezca un mapa continuo.

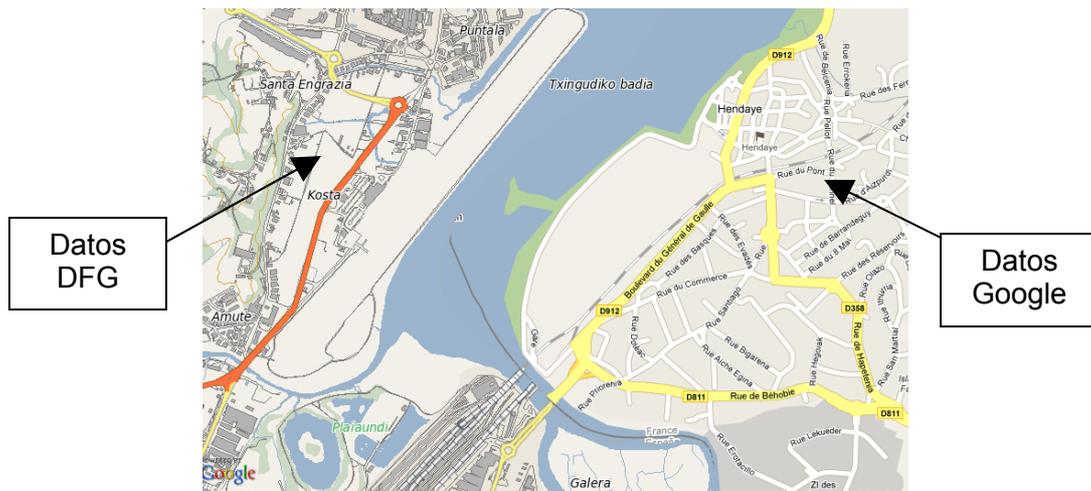


Figura 14: Superposición de datos.

## Consulta

Al cargar la página Web se descarga los vectores de las capas de cuencas para poder cambiar el color de los polígonos al pasar el cursor encima del mapa. Al seleccionar la opción de consulta se abre una ventana de atributos que incluye funciones de exportación a Excel o Shape. Para mejorar el rendimiento de la capa vectorial, al importar los datos a PostGIS se ha simplificado las líneas.

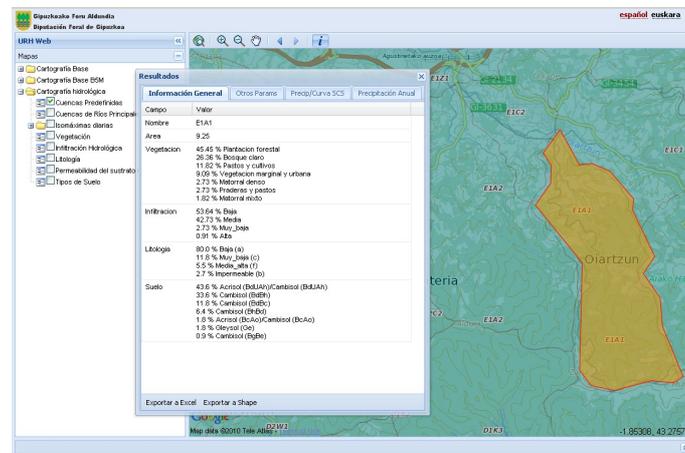


Figura 15: Capa Vectorial y la función de consulta.

## CONCLUSIONES

Después de realizar proyectos Web con diferentes tecnologías durante varios años, hemos llegado a la conclusión que para proyectos de Web Mapping el software *stack* Open Source GIS reúne una serie de características que iguala o supera el software comercial.

- Dibujo rápido.
- Estilos agradables e interfaces intuitivos.
- Generación rápido de prototipos con poco esfuerzo.
- Uso de mapas de fondo de empresas como Google, Microsoft etc. sin estar atado a su tecnología.

## REFERENCIAS

- ◆ OpenLayers: [www.openlayers.org](http://www.openlayers.org)
- ◆ GeoWebCache: [www.geowebcache.org/trac](http://www.geowebcache.org/trac)
- ◆ MapServer: [www.mapserver.org](http://www.mapserver.org)
- ◆ MapFish: [www.mapfish.org](http://www.mapfish.org)
- ◆ Posgres: [www.postgresql.org](http://www.postgresql.org)
- ◆ PostGIS: [www.postgis.org](http://www.postgis.org)
- ◆ Ubuntu: [www.ubuntu.com](http://www.ubuntu.com)
- ◆ Mogrify: <http://www.imagemagick.org/script/mogrify.php>
- ◆ Diputación Foral de Gipuzkoa : [www.gipuzkoa.net](http://www.gipuzkoa.net)