

## PROBLEMAS Y SOLUCIONES EN EL DESARROLLO DE LA IDE DE EXTREMADURA

Alberto. Aparicio Ríos<sup>(1)</sup>, Analista de Sistemas

<sup>(1)</sup> Centro de Información Cartográfica y Territorial. Servicio de Ordenación del Territorio. Dirección General de Transportes, Ordenación del Territorio y Urbanismo. Consejería de Fomento, Vivienda, Ordenación del Territorio y Urbanismo. Avda. de las Comunidades, s/n. 06800 Mérida. [alberto.aparicio@juntaextremadura.net](mailto:alberto.aparicio@juntaextremadura.net)

### RESUMEN

*Este documento expone algunos de los problemas que encontrados a lo largo del desarrollo del Visualizador y los Servicios de la IDE Extremadura, así como las soluciones aportadas por los técnicos encargados de su diseño, desarrollo y mantenimiento. Estos mismos problemas seguramente hayan afectado a otras organizaciones y/o profesionales interesados en la interoperabilidad de la información geoespacial, por ello esperamos que sea de utilidad las soluciones propuestas.*

**Palabras clave:** Visualizador, WMS, IDE, Extremadura, soluciones.

Fecha: octubre 2012

### ABSTRACT

*This document outlines some of the problems encountered during the development of SDI Extremadura's Display and Services and the solutions provided by the technicians responsible for the design, development and maintenance. These problems certainly affect other organizations and/or professionals interested in the interoperability of geospatial information, so we expect the proposed solutions to be useful.*

**Key words:** Display, WMS, SDI

Date: October 2012

## I DESCRIPCIÓN

Uno de los puntos más complicados a la hora de desarrollar un proyecto como la IDE Extremadura <sup>[1]</sup> ha sido plasmar de forma gráfica todo aquello que la teoría de servicios WMS, INSPIRE y demás normativas nos exigen a las organizaciones productoras de datos.

Tras varios años intentando cumplir con todas ellas quizá sería bueno poner en común todos aquellos problemas que nos hemos ido encontrando así como las soluciones propuestas para que sirvan de ayuda a la gente que se inicia en este mundo.

Desde nuestro punto de vista, quizá lo más importante de una IDE, además de su interoperabilidad, la estandarización... sea la facilidad de uso para el usuario. Por muchos servicios que ofrezcamos, si el usuario final no utiliza la IDE, no hemos logrado nada. Es por ello que, tras encuestar a personal del sector y usuarios finales de varios tipos, lo que más preocupa a la gente es la velocidad de uso. A veces no es operativo para quien necesita información el hecho de tener que esperar varios minutos a que se cargue el mapa, o a que se

refresque la página, o no poder utilizar un servicio WMS porque está caído. Para evitar estos problemas lo primero que abordamos fue la estructura de los servidores del Gobierno de Extremadura. Una vez hecho esto intentamos acelerar al máximo los servicios y el visor, para aprovechar el rendimiento que estas máquinas nos ofrecían. Por último se están añadiendo utilidades que hacen el trabajo del usuario externo mucho más sencillas.

## 1.1 Optimización de los servidores

En primer lugar instalamos dos máquinas con Linux, trabajando directamente con la consola, en lugar de utilizar el entorno gráfico, para poder aprovechar al máximo sus capacidades. Estudiando los rendimientos de la anterior versión en Windows nos dimos cuenta de que, tanto en velocidad de respuesta como en disponibilidad, no tenían ni punto de comparación.

El software utilizado para servicios de mapas es principalmente Mapserver<sup>[2]</sup>. Además de ser gratuito ofrece un gran rendimiento y bastantes formas de personalización que más tarde se explicarán.

Posteriormente se incorporó un balanceador hardware que actúa de frontal ante estos dos equipos. Con ello conseguimos que las peticiones de mapas se repartan entre dos máquinas acelerando al doble las respuestas de cara al usuario. También esto nos sirve para reducir los tiempos de inactividad del servicio. Es cierto que a veces puede caerse una máquina, pero entraría en funcionamiento la otra. Además podemos llevar a cabo el mantenimiento de un servicio mientras el balanceador dirige todas las peticiones al otro, y a la inversa posteriormente, haciendo que la disponibilidad sea cercana al 100%.

## 1.2 Optimización de servicios WMS

Como ya se ha comentado, el software de publicación de servicios WMS elegido ha sido Mapserver 6.0, el cual nos ofrece servicios WMS acordes al estándar OGC WMS 1.3.0<sup>[3]</sup>.

Este software nos permite trabajar muy rápidamente con una gran variedad de fuentes de datos, como pueden ser shapefile, TIFF o postgis. Para un óptimo rendimiento partimos del código fuente y lo recompilamos para la máquina que estamos utilizando. Incluso si lo instalamos como fast cgi podremos mantener abiertas las conexiones con los archivos y con las bases de datos, consiguiendo un rendimiento incluso mayor.

Una vez hecho esto, los servicios ofrecidos pueden optimizarse siguiendo varias pautas:

- **Proyecciones permitidas:** Para realizar las reproyecciones se utiliza la librería Proj.4<sup>[4]</sup>. Esta nos permite hacer un montón de conversiones entre miles de sistemas de referencia, de los cuales a nosotros solamente nos interesarán unos pocos. Eliminando el resto de referencias a los SRS no utilizados o bien situando las más utilizadas al inicio, conseguimos optimizar los tiempos de carga.
- **Creación de índices espaciales en servicios que utilizan imágenes raster:** Se trabaja con un shapefile de fondo que nos indica en cada caso cuál es el raster que se aplica en cada zona del mapa. Esto es útil, por ejemplo, trabajando con ortofotos. La directiva utilizada es TILEINDEX “ruta al shape”.
- **Hay que evitar mostrar capas cuando no sean útiles.** Para ello podemos limitar las escalas de visualización utilizando las directivas MINSCALEDENOM y MAXSCALEDENOM, consiguiendo así mostrar aquellas capas que realmente sean útiles a una cierta capa y evitando malgastar recursos cuando no sean necesarias.

- Lo mismo ocurre con la extensión geográfica. De nada nos sirve darle la extensión de España si solamente tenemos datos de la zona de Extremadura. Limitando la zona al máximo conseguiremos evitar peticiones a los servidores si estamos en zonas alejadas de la misma. Para ello usaremos el EXTENT ó “ows\_extent”, dentro de los metadatos del servicio.
- A la hora de crear varias capas dentro de un fichero MAP, lo lógico sería colocar las más usadas al inicio. También dentro de una capa, si se utilizan varios CLASS para distinguir los datos en categorías, el hecho de colocar las más utilizadas al inicio hará que consigamos mejores tiempos de respuesta.
- Cuando utilizamos shapes como fuentes de datos es bueno crear índices espaciales, sobre todo si se trata de ficheros grandes. La utilidad “shptree” que provee Mapserver nos puede ser de gran utilidad.
- Un paso próximo será la inclusión de servicios pre cacheados del tipo WMTS ó WMS-C. Con utilidades como Mapproxy, muy ligada a Mapserver, podríamos generar una rejilla de imágenes a partir del servicio WMS inicial, de tal forma que las peticiones que lleguen posteriormente no tengan que resolverse de nuevo generando el mapa, sino que se le reenvíen las imágenes ya cargadas. Con este sistema se obtiene un rendimiento muy superior al obtenido directamente por el servicio, aunque tiene la desventaja de que necesita gran capacidad de almacenamiento. Hemos hecho varios estudios para intentar optimizar este almacenamiento, y la solución encontrada ha sido formatear el disco donde se vaya a almacenar la caché utilizando un sistema de archivos ReiserFS, el cual está muy optimizado para millones de archivos muy pequeños, que son las características de los archivos de salida de las cachés, por lo que se adapta perfectamente.

### **I.3 Optimización del visualizador OpenLayers**

La librería OpenLayers <sup>[5]</sup> nos permite optimizar la carga de los mapas de ciertas maneras:

- Cuando añadimos una capa necesitamos especificar al menos una url de origen de los datos. En algunos navegadores se limita el número máximo de peticiones simultáneas a un servidor, pudiendo ocurrir que se pida solamente la mitad del área visual de cada vez si tenemos tileadas las peticiones, teniendo que esperar a que lleguen unas para poder pedir las otras. Puede solucionarse añadiendo varias URL's a dicha capa, bien con varios dominios o bien utilizando servidores virtuales a tal efecto.
- Lo óptimo sería tener un servicio cacheado (WMS-C) ó WMTS utilizando la misma rejilla que el visor está pidiendo, con los mismos niveles de zoom para evitar tener que hacer modificaciones a las imágenes cacheadas antes de devolverlas al cliente. Para ello el parámetro scales de openlayers nos puede ser bastante útil, no teniendo que limitarnos a la subdivisión en escalas que se hacía en versiones previas.

### **I.4 Modificaciones en el visor de la IDE Extremadura**

Con el fin de facilitar la vida al usuario y hacer que la búsqueda de información y la obtención de resultados sean más sencillas, se han incorporado algunos cambios personalizados en el visor:

- Consulta gráfica de información: Utilizando las plantillas de Mapserver es muy sencillo mostrar las peticiones GetFeatureInfo de forma visual para que le sea fácil al usuario interpretar el resultado de sus consultas. Permite incluso accesos a metadatos, ficheros asociados a la capa, etc. Los parámetros utilizados para indicarle al fichero de configuración del servicio que va a utilizar una plantilla son HEADER, FOOTER y TEMPLATE, siendo esta última la requerida si queremos llevar a cabo este tipo de consultas. Dichos ficheros tendrán un contenido html en el que se modificarán y/o añadirán los datos en función de los resultados obtenidos en el getFeatureInfo. La plantilla HEADER ha sido utilizada para definir los estilos que van a tener las tablas, iniciar la cabecera de las mismas... y el pie o FOOTER se ha usado para cerrar las tablas o capas que se estaban utilizando. Dentro de dichas plantillas se puede hacer referencia a los campos que obtenemos del getFeatureInfo, especificados en el bloque METADATA, concretamente en la etiqueta "wms\_feature\_info\_mime\_type" "text/html". Podemos definir los campos que serán consultables con la etiqueta "gml\_include\_items" "nombre,de,campos,separados,por, comas". Posteriormente haremos referencia a ellos en la plantilla:

```
<!-- MapServer Template -->  
<table><tr><td>[HOJA5000]</td></tr></table>
```

- Para visualización de fechas no hemos sido capaces de que sea Mapserver quien publique en un formato día/mes/año, así que la solución ha sido incorporar javascript a las plantillas del getFeatureInfo que se han comentado anteriormente. Lo mismo ocurre con otros campos codificados que solamente los productores de información a veces entendemos. De esta forma cualquiera que consulte el servicio en html podrá obtener la información de una manera legible.
- Normalmente openlayers se viene usando indicándole la escala máxima, la escala mínima y el número de niveles de zoom que queremos, lo cual deriva en unas escalas indeterminadas que no son demasiado útiles (ej. 1:1703992). Para solucionarlo hemos propuesto utilizar el parámetro scales en la creación del mapa en openlayers, de tal forma que se adapta perfectamente a los niveles de escala que nosotros deseamos:

```
map = new OpenLayers.Map({  
  div: "map",  
  ...  
  scales:[5000000, 2000000, , 10000, 5000, 2500, 1500, 1000, 500, 250],  
  ...  
});+
```

- A la hora de crear nuevas capas es posible parametrizarlas con algunos efectos visuales que harán más amigable la interfaz de usuario. Un claro ejemplo es 'transitionEffect = resize', lo que hará que mientras carga una capa a un zoom determinado nos seguirá mostrando la capa en el zoom previo al que vamos a visualizar, bien estirada o bien reducida, dependiendo de si vamos a una escala mayor o menor. De esta forma evitamos que al usuario se le quede la pantalla en blanco con la consiguiente sensación de que no hay datos o algo parecido.

```
var layer = new OpenLayers.Layer.WMS(  
  "Ortofoto",
```

```
"http://www.ideextremadura.com/CICTEX/EXTREMADURA_BASE",  
{layers: 'EXTREMADURA_BASE'},  
{transitionEffect: 'resize'  
});
```

- Dada nuestra imposibilidad para configurar el `getFeatureInfo` sobre capas raster, la solución ha sido que, en nuestras capas, cuando utilizamos un raster que lleva información, como por ejemplo, `EXTREMADURA_BASE`, es el propio visor quien se encarga de modificar el nombre de la capa a la que va a preguntar anteponiéndole la cadena "INFO", de tal forma que es transparente al usuario, aunque como digo es personalizado en nuestro visor, y en principio no está disponible para servicios WMS externos.
- Para acelerar la carga de algunas capas, la petición de información a un grupo de las mismas hará que se consigan mejores velocidades a la hora de generación de los mapas, ya que en vez de 4 peticiones sobre 4 mapas haremos solamente 1 petición. La desventaja de este método es que no podremos separar la información de las capas a menos que estén a diferentes escalas, lo cual sería deseable. El hecho de que varias capas salten a unas escalas u otras y se desactiven cuando no son visibles permitirá tener un único servicio con toda la información, sin necesidad de que sea el usuario quien active y desactive las capas.
- Cuando pedimos consultas `getFeatureInfo`, es normal que a veces aparezcan enlaces, bien como texto o bien como hipervínculos. En este último caso, al hacer click sobre ellos se nos abre en la misma pestaña a menos que esté preparado el servicio para ello, cosa que difícilmente hemos encontrado. Por este motivo, el visor intercepta todas las comunicaciones con los servicios para añadir la información necesaria al texto obtenido de tal forma que la apertura de enlaces se lleve a cabo en una nueva página permitiendo mantener el estado del mapa actual.
- En algunos servicios como el caso de planeamiento digital de Extremadura, donde una misma capa está compuesta de varias (polígonos, líneas, puntos y textos), se ha conseguido unificarlas todas en una, de tal forma que al usuario le dé la sensación de que trabaja con toda la información a la vez, haciendo más fácil el entendimiento de lo que está viendo en el mapa.
- No es necesaria la utilización de proxies (en principio), para las peticiones de mapas.

## 1.5 Servicios para el usuario

Por último, los últimos cambios que se están realizando son para facilitar el trabajo a los usuarios finales de la aplicación:

- Búsqueda por polígono/parcela: A la hora de trabajar, por ejemplo, con expedientes de calificación se hace imprescindible localizar los polígonos y parcelas de los distintos municipios. Es una de las herramientas más utilizadas. La información es proveniente de catastro directamente, ya que son ellos quienes pueden ofrecerla más actualizada.

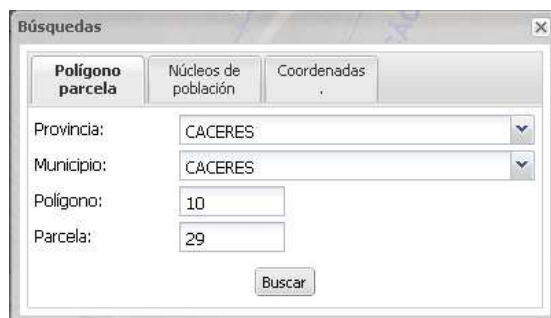


Figura 1: Funcionalidad de búsqueda por polígono/parcela.

- Generación de servicios WMS de utilidad y que nos distinguan de otras herramientas como Google Maps u Open Street Map, como pueden ser los expedientes de calificación, planeamiento urbanístico vigente, planos escaneados de ordenación municipal, información del atlas de Extremadura...
- Se están incluyendo posibilidades para que el usuario cargue sus propios shapes, kml, gml... para interactuar con la información disponible en la IDE.
- Entornos de visualización 3D, como el vuelo virtual de Extremadura (<http://www.ideextremadura.com/nasa/cictex.jnlp>) basados en todos estos servicios WMS generados previamente, donde podremos fusionar, de una forma muy vistosa todos estos servicios y datos optimizados anteriormente:

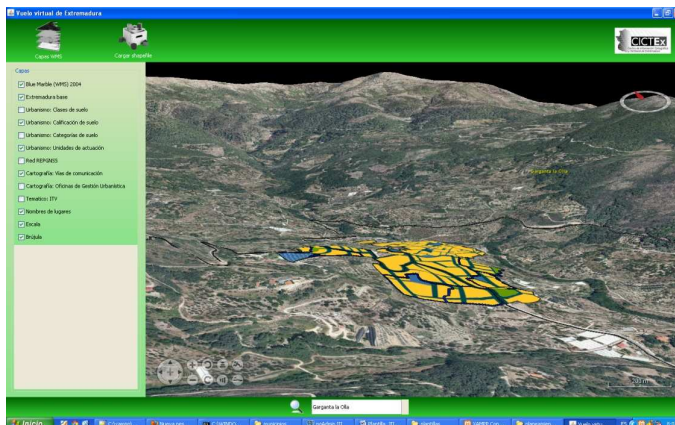


Figura 2: Vuelo virtual de Extremadura.

## Referencias.

- [1] <http://www.ideextremadura.com/Geoportal>
- [2] <http://mapserver.org/>
- [3] <http://www.opengeospatial.org/standards/wms/>
- [4] <http://trac.osgeo.org/proj/>
- [5] <http://openlayers.org>