Generación automática de visualizadores web 3D

Mediante el uso de los estándares WMS, WMC y KML

Fonts, Oscar

Resumen

Los visualizadores cartográficos web incorporan cada vez mayor funcionalidad y usabilidad, y se han convertido en la carta de presentación de los diversos geoportales que componen la IDE. A su vez, Google Earth ha popularizado la visualización tridimensional de contenido geográfico, personalizable ha facilitado enormemente la integración de visualizadores geográficos tridimensionales con otros contenidos web. Es natural, pues, que en la IDE exista interés en la visualización 3D, y se encuentren experiencias en la explotación de Google Earth y KML para este propósito [1]. Generalmente, la visualización 3D se ofrece como complemento a la visualización 2D, que continúa siendo la forma preferida de consulta cartográfica para la mayoría de aplicaciones [2].

El diseño, implantación y mantenimiento de un visualizador 3D puede suponer un esfuerzo extra para los equipos técnicos de un geoportal IDE. Se presenta aquí una técnica que permite la generación automática de vistas 3D a partir de una vista 2D, utilizando conjuntamente los estándares del OGC WMS (Web Map Service) [3], WMC (Web Map Context) [4] y KML (Keyhole Markup Language) [5].

El Web Map Context es un formato estándar para codificar un *contexto* de mapa [6]. El *contexto* describe la *vista cartográfica* de mapa que tiene un usuario en un momento dado: lista de capas activas e inactivas, y su orden de visualización, así como el ámbito geográfico (BBOX) y las medidas de la ventana de visualización. De este modo, WMC permite guardar determinada vista, y recuperarla más adelante, transmitirla o replicarla en otros visualizadores.

Para demostrar la viabilidad de esta técnica se presenta una aplicación llamada WMC2KML (ver demostración en [7]) que utiliza WMC para tomar una instantánea del estado de un visualizador 2D cualquiera que implemente WMC (por ejemplo, los basados en OpenLayers), y transformar automáticamente este *contexto* en un archivo KML, lo que permite replicar la misma vista en un globo virtual 3D, sea la versión Google Earth de escritorio, el 'plug-in' web de Google Earth, o cualquier otro visualizador que soporte KML.

PALABRAS CLAVE

Visualizadores, 3D, KML, WMC.

-=-=-=-

CONTACTOS

Oscar FONTS
fonts@uji.es
Universitat Jaume I
GeoInfo Group

INTRODUCCIÓN

Aunque para gran parte de las aplicaciones cartográficas la visualización en 2D es suficiente, e incluso preferible [2], existe un interés creciente por la visualización cartográfica 3D, utilizando como plataforma lo que se ha dado en llamar *globos virtuales* [8]. Con la aparición del *plug-in* de Google Earth para navegadores web[1], es posible la inclusión de una vista 3D personalizable en un navegador de forma sencilla.

Existe una gran cantidad de visualizadores web 2D desplegados en geoportales y nodos de la IDE en cuyo núcleo albergan OpenLayers[2]. Diseña un visualizador 3D personalizado y consistente con la información mostrada en su análogo 2D puede suponer un trabajo añadido de programación y coordinación entre ambos visores.

Este trabajo propone un método que sea capaz de generar una réplica 3D de los visualizadores 2D existentes, de forma automática, basándose en los estándares OGC inherentes a ambos paradigmas de visualización. El visualizador 3D generado automáticamente replicará la configuración de capas WMS, su estado de visualización, y el ámbito geográfico del visualizador 2D original. De este modo, se pretende minimizar el esfuerzo de implantación de estos nuevos visores, así como de coordinación entre los contenidos de ambos.

ARQUITECTURA

La idea subyacente es muy sencilla, y se basa en la constatación de los siguientes hechos:

a) La mayoría de visualizadores web modernos están basados en la librería OpenLayers.
b) OpenLayers implementa el estándar WMC, formato XML diseñado para codificar la distribución de capas y estado de una vista compuesta de mapa en un instante dado.
c) Esta misma información es posible describirla en formato KML.
d) La mayoría de visualizadores 3D utilizan KML como formato de intercambio de información geográfica.

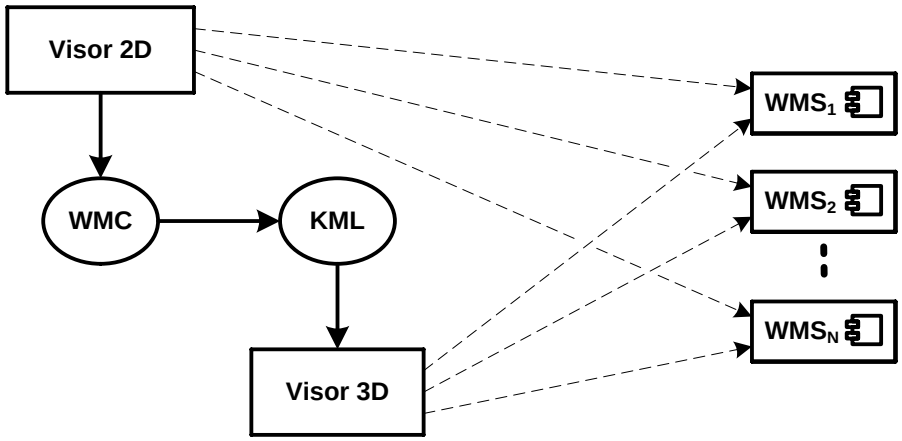

Figura 1: Transformación de contexto 2D a 3D; ambos visores acceden a WMS remotos

[1] Google Earth plugin: http://earth.google.com/plugin/
[2] Proyecto OpenLayers: http://www.openlayers.org

Por tanto,

- Para trasladar un contexto de mapa de un visor 2D a un visor 3D, basta con hacer una transformación del formato WMC al formato KML, siendo ambos lenguajes XML.
- Esta transformación será aplicable a la mayoría de implementaciones existentes.

La transformación, que hemos dado en llamar `WMC2KML`, y que se publica como un servicio web, también puede ser de utilidad para otras aplicaciones de escritorio basadas en los estándares mencionados.

En cualquier caso, la información cartográfica se recuperará de forma remota a través de servicios WMS.

ESTÁNDARES UTILIZADOS

En esta sección se analizarán los estándares OGC WMC y KML utilizados en la propuesta. No se expondrán exhaustivamente todos los elementos de ambos estándares, sino aquellos elementos relevantes para el objeto de este trabajo: transformar automáticamente un contexto de mapa a una vista 3D, estableciendo vínculos entre ambos.

Web Map Context (WMC)

WMC [4] es un formato de intercambio de vistas ente agentes OGC. WMC persiste el estado de visualización de una vista multicapa, denominado contexto, en un formato independiente del visualizador.

Contiene los siguientes elementos (se ha marcado con un asterisco* aquellos elementos no presentes en el estándar original, pero que introduce OpenLayers para una codificación más completa del contexto):

- Parámetros generales:
  - Ámbito geográfico de la vista actual, dependerá del nivel de zoom la zona donde esté centrada la vista (elemento *BoundingBox*).

- Lista de capas (colección de elementos *Layer*). Para cada una:
  - Título de la capa que se muestra al usuario (elemento *Title*).
  - Muestra si está activa o inactiva (parámetro *hidden*).
  - Indica el tipo de servicio del que se obtienen los datos (elementos *Server* y *OnlineResource*). En nuestro caso, sólo seleccionaremos las capas de tipo `OGC:WMS`.
  - Nombre interno de la capa (elemento *Name*), que corresponde al parámetro `layers` de WMS.
  - Extensión total de la capa, o dominio geográfico (elemento *Extension/maxExtent*, específico de los contextos generados con OpenLayers).

Keyhole Markup Language (KML)

KML [5] permite reproducir la información de contexto utilizando sus propios elementos, aunque en algún caso la traslación no es trivial.

El contexto codificado en KML contendrá los siguientes elementos:

- Parámetros generales:
  - Ámbito geográfico de la vista actual. En este caso, posición inicial de la cámara en un escenario 3D (elemento *LookAt*).

- Lista de capas (colección de elementos *Folder*). Para cada una:
  - Título de la capa que se muestra al usuario (elemento *Name*).
  - Muestra si está activa o inactiva (elemento *visibility*).
  - Estructura *superoverlay*, que muestra las capas a diferentes niveles de detalle recursivamente (contenido en un elemento *NetworkLink*). Consta de:
    - Ámbito geográfico para el primer de detalle (elemento *Region/LatLonAltBox*, específico de los contextos generados con OpenLayers).
    - Enlace al primer nivel de detalle. Se pasan como parámetros el mencionado ámbito geográfico, y la URL de base del servicio WMS donde recuperar las imágenes (elemento *Link/href*).

Proceso de transformación WMC2KML

Cabe destacar que la relación de algunos elementos entre ambos formatos no es directa.

En cuanto al **ámbito geográfico para las capas**, debe tenerse en cuenta que KML trabaja en el sistema de referencia EPSG:4326. En caso de que el contexto venga definido en otro sistema de referencia, debe transformarse. Para esta implementación se ha utilizado una calculadora geodésica simplificada que reproyecta de UTM a coordenadas geográficas, y aplica una transformación de datum de 7 parámetros, de ED50 a WGS84 [9].

En cuanto a la **transformación del ámbito de visualización**, se parte de un rectángulo contenedor (BBOX) que ha de transformarse a una posición de cámara según los parámetros del elemento *LookAt*: Latitud y longitud del punto central hacia el que se dirige, distancia de la cámara a dicho punto, y ángulo bajo el que se observa el terreno. Deduciendo la distancia focal de la cámara, puede calcularse fácilmente una vista cenital equivalente al BBOX original [10] [11].

Por último, **las capas** en el visualizador 3D **siguen un esquema de pirámide multirresolución** (*quadtile*). El ámbito geográfico de las imágenes que componen cada nivel de resolución, así como la petición WMS completa, deben codificarse directamente en KML. Para ello, se ha recurrido a un segundo servicio web que hemos llamado `Superoverlay`. Este servicio construye una petición WMS para una región a un nivel de detalle dado, y enlaza a las 4 subregiones que conformarán el nivel de detalle inmediatamente superior. Siguiendo una estructura recursiva y mediante el uso del elemento KML *NetworkLink*, se pueden ir construyendo regiones y peticiones WMS a medida que son requeridas por el motor de visualización 3D [12].

DISEÑO

En esta sección se muestran las clases principales que componen los dos servicios web presentados, `WMC2KML` y `Superoverlay`, así como la secuencia de inicialización del visualizador 3D.

Diagrama de clases

En la siguiente figura se muestran las principales clases de la aplicación. El punto de partida es la clase WMS2KML, que despliega un servicio web capaz de recibir un documento WMC enviado por HTTP POST desde un cliente remoto, y devolver un KML que contenga un contexto equivalente.

Figura 2: Diagrama de las clases principales

Para su transformación, se llama a una clase WMCReader que construye un contexto en memoria, utilizando las clases:

- Context, que contiene el ámbito de visualización y la colección de capas asociadas.
- Layer, que contiene los detalles de contexto sobre cada capa (dominio geográfico, URL del servicio WMS, y opciones de visualización).

WMCReader también se encarga de transformar el sistema de referencia del ámbito de visualización, en caso que sea necesario. Para ello, hace uso de una calculadora geodésica (no representada en la figura anterior por simplicidad).

La clase Context se encarga de codificar el contexto en KML, construyendo el elemento LookAt [10, 11], y las carpetas para cada una de las capas. Cada Layer, a su vez, construye la estructura básica de su estructura KML Superoverlay [12].

A medida que el usuario navega por la vista 3D, se requerirá la descarga de imágenes WMS a más alta resolución. El fragmento de código KML necesario para ello se construye mediante una llamada a la clase Superoverlay.

Secuencia de inicialización

Mostramos aquí la secuencia de inicialización que se tiene lugar a partir del evento "ver en 3D" hasta que el visualizador 3D está inicializado, tal como se muestra en la siguiente figura:

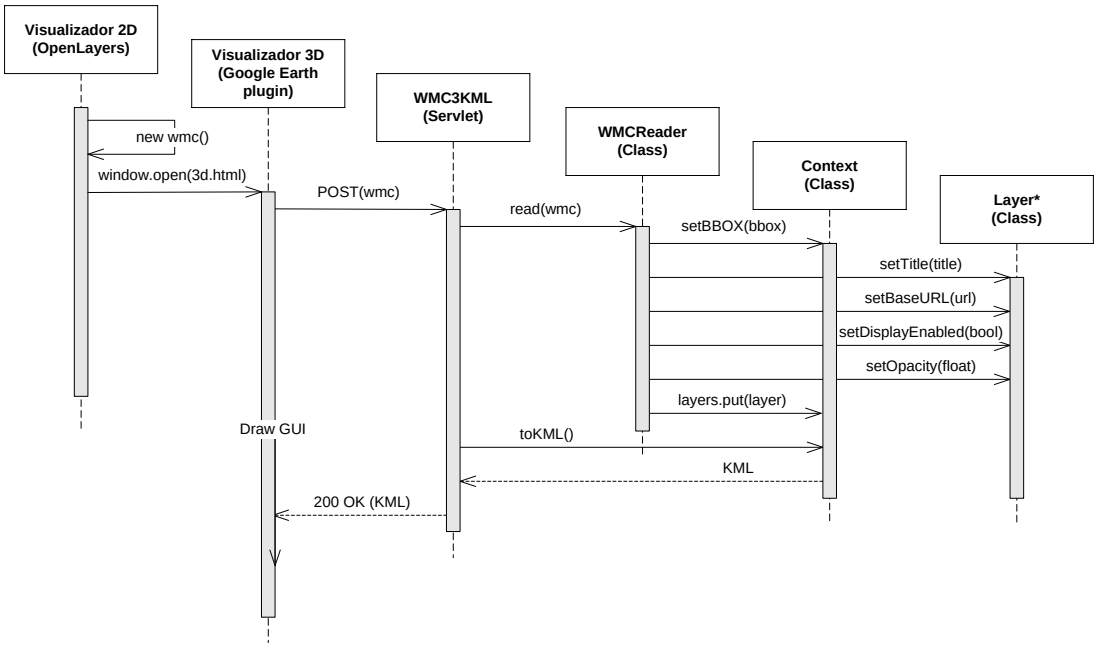

Figura 3. Secuencia de inicialización del visor 3D

En primer lugar, el visualizador 2D serializará su estado actual utilizando `wmc`. Desde el visualizador 2D se crea una nueva ventana que contendrá el visualizador 3D. Éste llama al servicio `WMS2KML` pasando el contexto como parámetro.

En el servidor, el servicio `WMS2KML` utilizará la clase `WMCReader` para crear un objeto de tipo `Context`, donde se guardarán las coordenadas `BBOX` de la vista actual. Si el sistema de referencia entre el visor 2D y el visor 3D no es el mismo, en este momento `Context` llamará a una calculadora geodésica (no representada en la figura) para transformar dicho `BBOX`. A continuación, `WMCReader` creará tantos objetos `Layer` como capas se hallen en el contexto, e informará sus parámetros:

- Título de la capa, para mostrar al usuario.
- URL base del servicio WMS, que contiene todos los parámetros de la petición WMS excepto el BBOX.
- Indicador de si la capa está activada.
- Transparencia de la capa, en un valor entre 0 y 1.

Finalmente, la colección de capas se asocia al `Context`. Una vez `Context` contiene toda la información, su método `toKML()` genera el KML que será enviado al cliente. Este KML contiene una colección de `Placemarks` (capas), que son el punto de partida para la construcción de los respectivos *superoverlays* (ver apartado *Keyhole Markup Language*).

IMPLEMENTACIÓN

Puede encontrarse una demostración en línea de esta aplicación en [7].

Parte cliente 2D

Se compone de un control de OpenLayers con una colección de capas WMS.

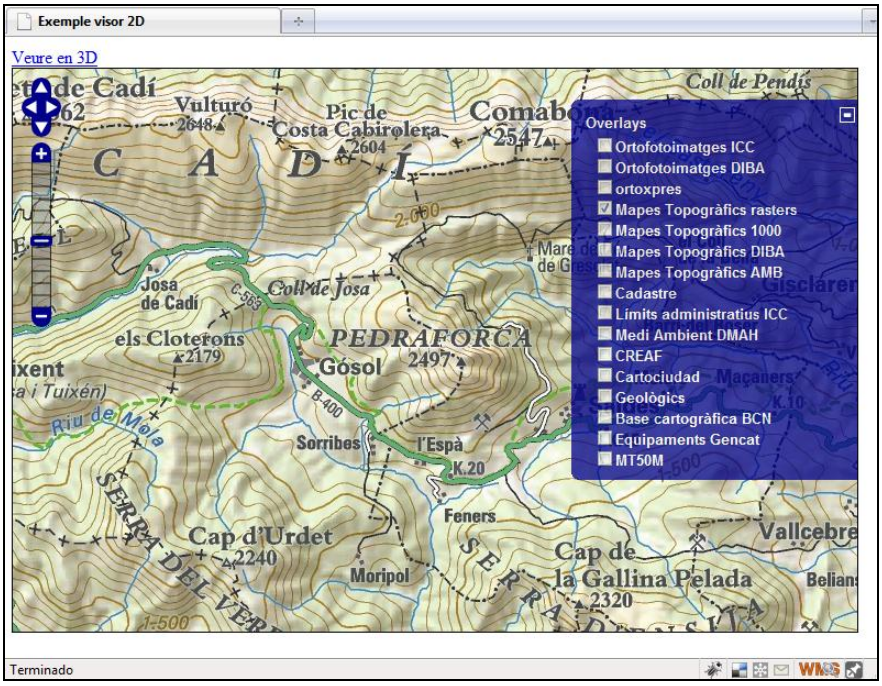

Figura 4. Ejemplo de vista 2D.

Componentes de la vista.

Parte cliente 3D

Se basa en el plugin de Google Earth. La tabla de contenidos (capas) se construye utilizando una versión modificada de la librería KMLTree [13].

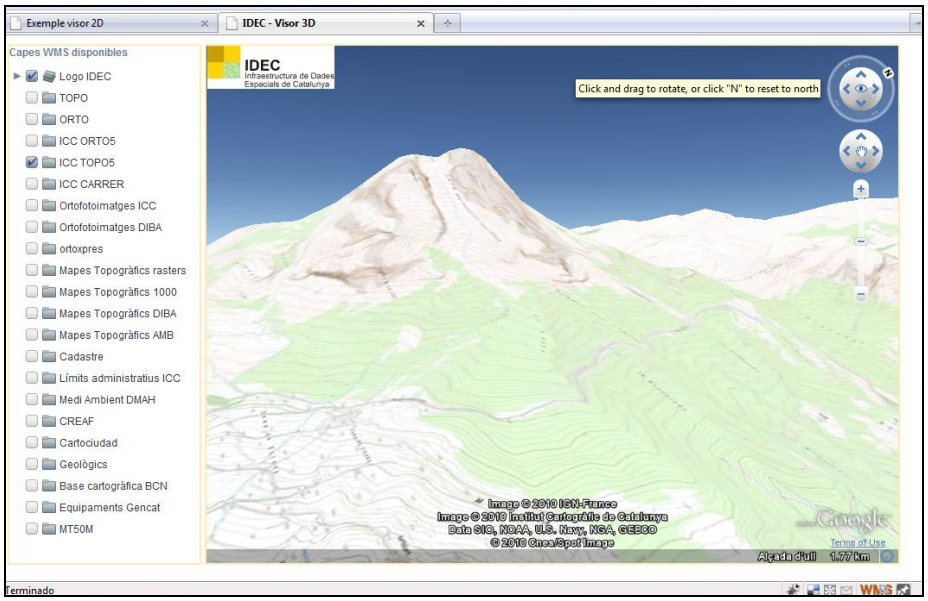

Figura 5. Ejemplo de vista 3D.

Instalación y uso

El generador de vistas 3D es una aplicación java que se distribuye en un fichero `.war`, desplegable fácilmente en cualquier contenedor de aplicaciones (por ejemplo, Tomcat 6). El código es compilable en versiones de Java 1.5 o en Java 6.

El visor 3D no contiene ficheros de configuración. Toda la información necesaria para generar el visor 3D se obtiene del fichero de contexto WMC de entrada.

Se ha supuesto que el visor 2D de origen está basado en OpenLayers 2.x. En el instante de crear el visor 3D, debe generarse la información de contexto, y abrir una nueva ventana, cuya dirección es la ruta donde se haya desplegado la aplicación, más `3d.html`. Por ejemplo, se puede crear una función que contenga estas dos instrucciones:

```
var wmc;
function open3d() {
   wmc = new OpenLayers.Format.WMC().write(map);
   window.open('http://delta.icc.cat/IDEC-Visor3D/3d.html');
}
```

Bastará con vincular el evento de usuario que deseemos (generalmente un clic en un enlace o botón) con una llamada a la función **`open3D()`**. En el ejemplo de código se ha supuesto que la aplicación 3D se encuentra en http://delta.icc.cat/IDEC-Visor3D/, y que el objeto mapa de OpenLayers se llama **`map`**. Obsérvese que es necesario instanciar una variable global de nombre `wmc` que contendrá el contexto de la vista actual en el formato *OGC Web Map Context*, que describirá la lista de capas actuales, sus parámetros (tipos de servicio, formato, URL, transparencia, visibilidad, etc), así como el ámbito geográfico (BBOX) visible en pantalla en el momento de generar el contexto.

LIMITACIONES

La primera limitación en un visor 3D de estas características es el uso de *superoverlays*. Esto implica la descarga masiva de imágenes a diferentes resoluciones, que deben poder obtenerse en un tiempo mínimo para que la navegación 3D resulte fluida. Generalmente las imágenes están preprocesadas en el sistema de referencia y a las resoluciones adecuadas al globo virtual, mediante cachés de teselas diseñadas *ex profeso* [14]. En este caso los mapas se están obteniendo a través de servicios WMS, cuyos datos no estarán cacheados. Es más, si se trata de cartografías oficiales de España y Portugal, es probable que no se encuentren en el sistema de referencia deseado, EPSG:4326, lo que implicará una reproyección y un cambio de datum al vuelo además del posible rasterizado, simbolización, y/o cambio de formato. En resumen, para la técnica de *superoverlay* aplicada, los servicios WMS son lentos, y difícilmente pueden soportar grandes cantidades de peticiones simultáneas.

Otra limitación intrínseca a esta propuesta es la incapacidad de visualizar datos vectoriales directamente, lo cual resta interactividad al contenido cartográfico visualizado. Un servicio WFS típicamente ofrece los datos en formato GML, lenguaje muy prolijo, y no diseñado para su visualización en tiempo real. En caso de querer visualizar información vectorial, debería poder obtenerse directamente en formato KML, que, entre otras cosas, permite definir simbolización, elementos tridimensionales, y características que aportan interactividad. Existen implementaciones de servicios OGC que ofrecen KML nativamente como formato de respuesta, como es el caso de GeoServer[3]. Esto restaría generalidad a la propuesta, que quedaría condicionada a un fabricante en la parte de geoservicios OGC, pero es una técnica a explotar si se desea información vectorial.

Por último, cabe destacar que el sistema de referencia espacial utilizado en el visor 2D puede no ser el mismo que el usado por el visor 3D, en cuyo caso deben transformarse las coordenadas del documento de contexto WMC al transformarlo a KML. La implementación propuesta incluye una calculadora geodésica simplificada, que convierte de coordenadas UTM y datum ED50 a coordenadas geográficas en datum WGS84. Para aportar el máximo de generalidad posible, sería preciso incluir una librería de transformación de coordenadas más completa, como Geotools[4], o PROJ.4[5].

[3] Proyecto Geoserver: http://www.geoserver.org
[4] Proyecto Geotools: http://www.geotools.org
[5] Proyecto PROJ.4: http://trac.osgeo.org/proj/

REFERENCIAS

[1] **Juan Jorge Rosales León**, GRAFCAN. *Difusión de IDECanarias a través del estándar OpenGIS®KML Encoding Estandar*. JIDEE, 2008.

[2] **Enric Rodellas, Andrés Valentín, Jordi L. Ramot, et al.** *Mesa redonda: Navegación 3D web, ¿Moda o necesidad?* VII Fórum TIG-SIG, 2008.

[3] **J. Beaujardiere (editor)**, *OpenGis Web Map Server Implementation Specificatio, 1.3.0.* Ref. OGC 06-042. Open Geospatial Consortium, 2006.

[4] **Jerome Sonnet (editor),** *OpenGIS Web Map Context Implementation Specification 1.1.* Ref. OGC 05-005. Open Geospatial Consortium, 2005.

[5] **T. Wilson.** *OpenGIS KML (Keyhole Markup Language) 2.2.0.* Ref. OGC 07-147r2. Open Geospatial Consortium, 2008.

[6] **Miguel Luaces, Mª Dolores Arteaga, Universitat de Girona-SIGTE.** *Módulo de Estándares e Infraestructuras de Datos Espaciale., Lección 7: Otros servicios y especificaciones OGC. 7.2. Web Map Context.* Apuntes del Máster Profesional UNIGIS en Gestión de SIG, 11ª Edición, 2009.

[7] Demo: http://delta.icc.cat/idecwebservices/mapawms/index.jsp?lang=es_ES

[8] **Michael Gould.** *El papel de las IDEs en globos virtuales.* V Jornadas Técnicas de la IDE de España (JIDEE), 2008.

[9] **Oscar Fonts, José Luis Valenzuela.** *Diseño de un sistema de localización basado en GPS y PDA*, 2002. Capítulo III.4. http://cataleg.upc.edu/record=b1204002~S1

[10] *KML Reference. <LookAt>*. Google Inc. http://code.google.com/apis/kml/documentation/kmlreference.html#lookat

[11] **Justin Deoliveira.** *Make a good LookAt for KML layers.* Código fuente de Geoserver. http://jira.codehaus.org/browse/GEOS-1240

[12] *KML 2.1 Tutorial. Working with regions.* Google Inc. http://code.google.com/apis/kml/documentation/kml_21tutorial.html#workingregions

[13] **Chad Burt.** KMLTree project. http://code.google.com/p/kmltree/

[14] **Oscar Fonts, Carlos Granell.** *Visualización geográfica 3D: estándares y aplicaciones.* III Jornadas de SIG Libre, Marzo 2009, ISBN 978-84-691-9409-6