

# Un visualizador web de mapas 3D teselados basados en modelos de CityGML

Rubén Béjar<sup>1</sup>, David Gayán-Asensio<sup>1</sup>, Javier Eced-Cerdán<sup>1</sup>, Juan López-de  
Larrínzar-Galdámez<sup>2</sup>, Pedro R. Muro-Medrano<sup>1</sup>

<sup>1</sup>Departamento de Informática e Ingeniería de Sistemas.  
Universidad Zaragoza

{rbejar,dgayan,javierec,prmuro}@unizar.es

<sup>2</sup>GeoSpatiumLab, S.L.

{juanlg}@geoslab.com

## Resumen

En este artículo se presenta un sistema para la visualización web de modelos tridimensionales creados en el lenguaje CityGML. La visualización sólo requiere un navegador web, en cualquier plataforma y sin necesidad de complementos o *plug-ins*, y está basada en teselas creadas en perspectiva isométrica, de manera que la interfaz de usuario es muy simple y las prestaciones de la aplicación muy satisfactorias.

**Palabras clave:** CityGML. 3D. Visualización. Web. Open Layers.

## 1 Introducción

Este artículo presenta una aplicación web que permite a los usuarios la visualización interactiva de ciudades en tres dimensiones (3D) desde diferentes puntos de vista. Esta aplicación funciona en un navegador web sin necesidad de instalar ningún conector adicional. Las ciudades que se pueden visualizar deben estar modeladas en el formato CityGML [1]. CityGML es un lenguaje, basado en GML, que permite representar, almacenar e intercambiar modelos de ciudades y paisajes 3D.

La aplicación se compone de una parte cliente y de una parte servidor. La parte cliente está basada en la biblioteca libre Open Layers<sup>1</sup>. Ésta es una biblioteca JavaScript que permite acceder a servicios web de mapas en dos dimensiones (2D). Estos mapas pueden estar teselados (divididos en imágenes de tamaño prefijado) para hacer más rápida su visualización. Para realizar la aplicación descrita en este artículo se ha adaptado Open Layers al caso de mapas en tres dimensiones y teselados.

Los servicios web de mapas teselados se pueden basar en un servicio de mapas estándar y en una caché de almacenamiento de teselas. El cliente web accede a la caché y ésta devuelve la tesela solicitada si la tiene almacenada; si no, la solicita a un servicio web de mapas estándar y la almacena. En este proyecto se ha modificado una caché de teselas hasta ahora destinada a datos en 2D, para que soporte datos 3D.

La parte del servidor de la aplicación es un servidor de visualización 3D que se ha desarrollado para crear las teselas. Este servicio se compone de un servidor web embebido y un componente de visualización 3D ya existente que se encarga de la importación del fichero CityGML a visualizar y el pintado del mismo.

## 2 Diseño del sistema

La Figura 1 muestra una vista arquitectural del sistema. En esta vista se muestra una aplicación web, basada en OpenLayers, que se conecta a una caché de teselas solicitando el área geográfica a visualizar. La caché devuelve las teselas correspondientes a ese área geográfica si ya las tiene generadas, o bien las genera delegando la petición a un servicio web de visualización de datos CityGML en 3D al que se le pide que devuelva una vista isométrica de los datos.

---

<sup>1</sup><http://openlayers.org/>

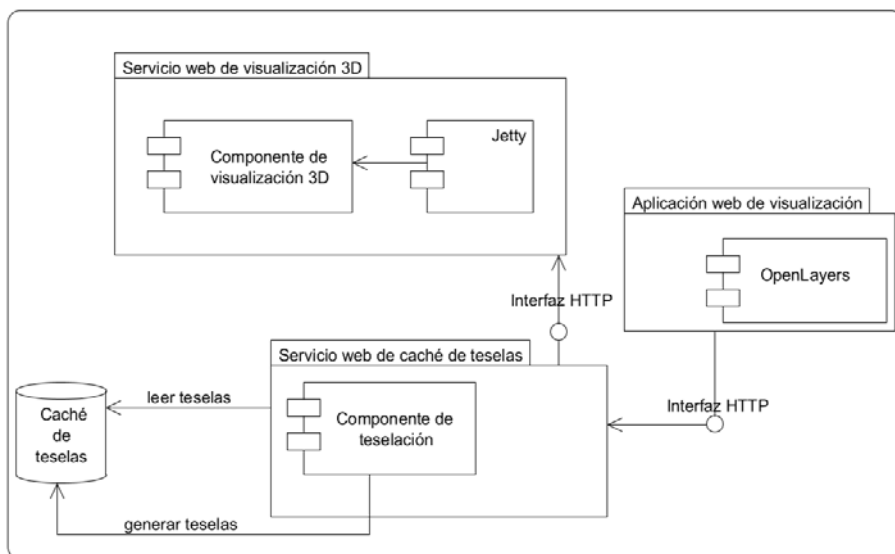


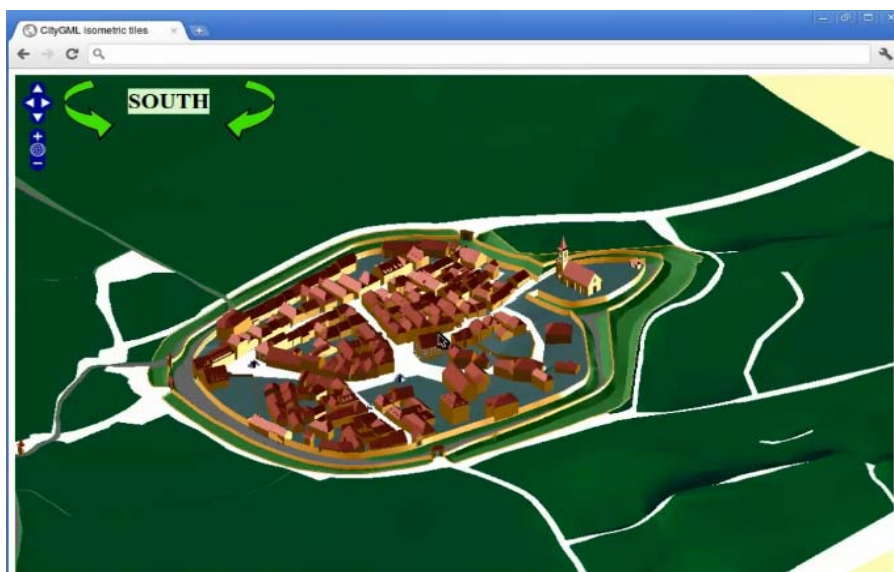
Figura 1: Componentes del sistema

El servicio web de visualización de datos CityGML en 3D está basado en el visualizador de código abierto Aristoteles Viewer [2], desarrollado en Java. Para el acceso al servicio web de visualización de datos CityGML en 3D se ha creado una interfaz similar a la operación *GetScene* del borrador de la especificación, todavía en borrador, del servicio web 3D del OGC [3].

Para aprovechar al máximo las posibilidades ofrecidas por Open Layers para la visualización de mapas teselados, se ha optado por crear cada vista de una ciudad 3D (norte, sur, este y oeste) como una capa distinta. En lugar de usar la herramienta por defecto de Open Layers para el cambio de capa, que está diseñada para capas 2D superpuestas, se ha enlazado la funcionalidad del cambio de capa/vista con dos botones con flechas en perspectiva que facilitan al usuario entender que se va a producir un “giro” de la ciudad visualizada.

### 3 Resultados

La Figura 2 muestra la aplicación en funcionamiento. La funcionalidad de la misma es similar a la de cualquier visualizador sencillo de mapas en 2D, con la posibilidad de desplazar el mapa y cambiar el nivel de zoom. Además se han incorporado dos botones que permiten girar la vista. Los datos que se muestran son un modelo virtual de la ciudad de Ettenheim, Alemania, creado por el *Research Center Karlsruhe*, del *Institute for Applied Computer Science*<sup>2</sup>. Se puede ver un video del sistema en funcionamiento en <http://www.youtube.com/watch?v=PFrszcnw1FM>.



*Figura 2: Aplicación de visualización en acción*

Se han realizado algunas pruebas, muy preliminares, de rendimiento para facilitar posteriores estimaciones sobre el tiempo de llenado de las cachés de teselas a partir de diferentes modelos en CityGML. Para ello se han tomado modelos disponibles en Internet con 3014 objetos (Ettenheim), 1918 objetos (Waldbruecke) y 813 objetos (Munich) y se ha medido el tiempo que cuesta crear teselas en distintos niveles de zoom. La Figura 3 muestra algunos de los resultados. Como era de esperar, a mayor número de objetos más tiempo cuesta generar las teselas. También era de esperar que conforme el zoom es más cercano, cuesta menos tiempo generar las teselas porque el componente de pintado tiene que tener menos objetos en

<sup>2</sup><http://www.iai.fzk.de/www-extern/index.php?id=1040&L=1>

cuenta. A partir de un cierto nivel de zoom, el tiempo de pintado de las teselas ya no varía mucho en los sucesivos niveles.

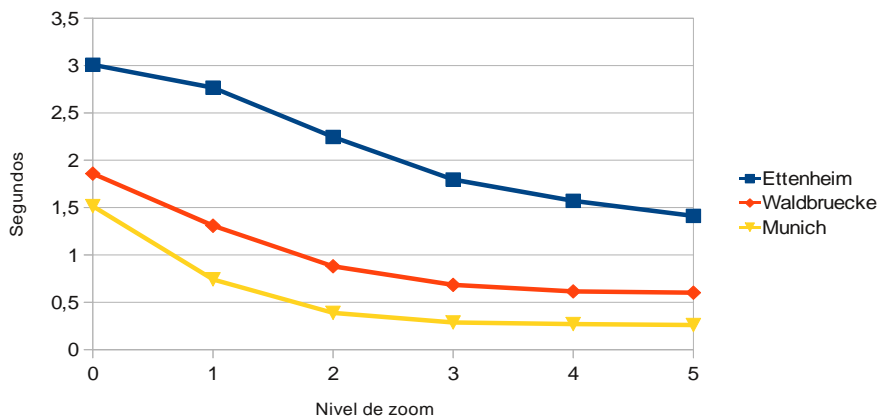


Figura 3: Pruebas de rendimiento en la creación de teselas

## 4 Conclusiones

Se ha diseñado e implementado un sistema para la visualización en una aplicación web de modelos 3D creados en el lenguaje CityGML. Se ha elegido una estrategia basada en teselas pregeneradas y en una perspectiva isométrica para permitir que la aplicación funcione en cualquier navegador, sin la necesidad de complementos o *plug-ins*, y que las prestaciones percibidas por el usuario sean buenas. La flexibilidad que permiten otras aproximaciones a la visualización 3D en cuanto al posicionamiento y orientación de la cámara virtual se cambian por sencillez de uso, cumplimiento de estándares y prestaciones. Con una interfaz y un punto de vista similar el de las imágenes aéreas oblicuas que ofrecen algunos callejeros y proveedores de mapas en Internet, creemos que esta aproximación es adecuada para mostrar información 3D de manera sencilla y ofreciendo buenas prestaciones incluso en computadores poco potentes (incluyendo *smartphones* y *tablets*), aunque indudablemente ciertos usos y ciertos usuarios seguirán necesitando más flexibilidad y otro tipo de interfaces más complejas para trabajar con este tipo de información.

Como trabajo futuro, el diseño del sistema nos permite abordar la creación de teselas a partir de modelos 3D no basados en CityGML sustituyendo el componente basado en Aristoteles Viewer por otros, lo que nos abre la puerta a poder visualizar información procedente de otras fuentes.

## **Agradecimientos.**

Este trabajo ha sido parcialmente financiado por el Gobierno de España mediante los proyectos TIN2009-10971 y “España Virtual” (ref. CENIT 2008-1030), el Centro Nacional de Información Geográfica (CNIG) y GeoSpatiumLab S.L.

## **Referencias**

- [1] Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C. (eds.): OpenGIS® City Geography Markup Language (CityGML) Encoding Standard. 2008.
- [2] Dörschlag, D., Drerup, J. Institute of Geodesy and Geoinformation Bonn – Department of Geoinformation: Aristoteles Viewer, <http://www.ikg.uni-bonn.de/aristoteles/index.php/Aristoteles>
- [3] Schilling, A., Kolbe, T.H. (eds.): Draft for Candidate OpenGIS® Web 3D Service Interface Standard. 2010.