

# Servicios de visualización INSPIRE basados en teselas vectoriales

Una aproximación a los servicios de mapas basados en teselas vectoriales en el contexto de INSPIRE

**LÓPEZ, Emilio; BÉJAR, Rubén; BARRERA, Jesús; LOPEZ-PELLICER, Francisco J.; RODRÍGUEZ, Antonio F.; ABAD, Paloma**

Las teselas vectoriales son representaciones de datos geográficos vectoriales que cubren una extensión rectangular. Son similares a las teselas de mapas ráster (de imagen) en cuanto a su intención, que es la de proporcionar soporte para servicios de mapas rápidos que puedan aprovechar la prerrenderización, el uso de cachés y las peticiones concurrentes. Se diferencian en un aspecto fundamental: en lugar de proporcionar una imagen prefijada y con un estilo cartográfico definido para unos datos geográficos, son representaciones vectoriales que pueden manipularse en las aplicaciones cliente que acceden a esos servicios de mapas.

Es importante resaltar que son representaciones pensadas para la visualización y que, por tanto, su entorno de aplicación está relacionado con los servicios de visualización y no con los de descarga. Los datos subyacentes, por ejemplo los objetos geográficos, se seleccionarán, combinarán, simplificarán y recortarán para generar teselas vectoriales pequeñas que puedan ser entregadas a dispositivos y a aplicaciones cliente relativamente simples, en las que se les asignarán estilos cartográficos diferentes en función de las necesidades del usuario.

Existe una especificación pública y abierta para teselas vectoriales, definida por Blake Thompson (Mapbox), la «*Mapbox vector tiles specification*», que ya soporta bastantes paquetes de *software*, tanto clientes como servidores de mapas (véase <https://github.com/mapbox/awesome-vector-tiles>).

Desde el punto de vista de las interfaces de los servicios, no hay diferencia entre las teselas vectoriales y las de imagen, simplemente están en formatos distintos (por ejemplo, en lugar de en PNG, pueden estar codificadas en GeoJSON). Esto significa que se pueden suministrar a través de cualquier interfaz de servicio de mapas que soporte teselas de imagen (p.ej. OGC WMTS), aunque el servidor que implemente esa interfaz tiene que proporcionar dicho soporte específico para teselas vectoriales.

Este artículo examina las ventajas e inconvenientes de las teselas vectoriales y hace algunas propuestas técnicas para poner en marcha servicios de visualización INSPIRE basados en ellas. Estas propuestas están actualmente siendo examinadas por el subgrupo técnico permanente del grupo de mantenimiento e implementación de INSPIRE (MIG-T). En ellas se describe cómo poner en marcha un servidor que soporte el perfil INSPIRE de WMTS 1.0.0 con teselas vectoriales teniendo en cuenta los posibles problemas que pueden surgir. También se presenta un prototipo de *software* que se está usando para validar esas propuestas.

## PALABRAS CLAVE

Teselas vectoriales, Servicios de visualización, WMTS, INSPIRE.

## INTRODUCCIÓN

Desde que, en 2005, Google lanzó Google Maps y popularizó el uso de mapas en navegadores de Internet basados en teselas ráster, han surgido multitud de estándares, herramientas y aplicaciones que permiten crear teselas, servir las de manera interoperable a través de Internet y aprovecharlas

en clientes de escritorio o de web, en móviles inteligentes y tabletas.

Las teselas vectoriales, a diferencia de las teselas ráster, incluyen objetos geográficos con geometrías vectoriales, generalmente simplificados con respecto a los datos a partir de los que se han generado, en lugar de representaciones cartográficas predefinidas en forma de imagen. De este modo, la tecnología de las teselas vectoriales ofrece diversas ventajas frente a las teselas ráster que son tan comunes hoy en día:

- Las teselas vectoriales pueden ser más ligeras que las ráster dependiendo del formato en el que se sirvan (MapBox indica que en la media, aproximadamente, un 75 % menores), lo que se traduce en un menor consumo de ancho de banda, algo que es especialmente interesante en planes de datos móviles, y mayor rapidez en su descarga.
- La visualización de los datos se realiza en el cliente (aplicación, móvil o de escritorio, o navegador de Internet), dando así total flexibilidad a los desarrolladores de aplicaciones a la hora de dar estilos a sus mapas.
- Los datos asociados a cada elemento del mapa llegan al cliente, lo que permite que se puedan hacer consultas muy rápidas sobre ellos (por ejemplo, para mostrar *map tips*), seleccionar las capas visibles o ajustar la interfaz de usuario al mapa que el usuario quiere ver. Por supuesto, es el proveedor de datos el que decide la representación vectorial de sus datos que se va a servir, pudiendo elegir qué objetos geográficos se sirven, con qué precisión y qué atributos se van a incluir y cuáles no.

La principal desventaja de las teselas vectoriales es que requieren que las aplicaciones de usuario sean más potentes, y por tanto más pesadas, y que los dispositivos que las muestran tengan mayor capacidad de procesamiento. Esto hoy en día no representa un problema para la mayor parte de los usuarios gracias al ritmo de avance de estas tecnologías.

En principio cualquier formato de geodatos vectoriales se podría utilizar para crear teselas vectoriales, y éstas se podrían suministrar a través de servicios existentes con interfaces estandarizadas. Sin embargo, están surgiendo alternativas especializadas ajustadas a la problemática concreta de las teselas vectoriales:

- A partir de datos del proyecto OpenStreetMap, diversos proyectos llevan un tiempo ofreciendo teselas vectoriales en distintos formatos [1].
- La empresa Mapbox ha creado una especificación abierta para teselas vectoriales [2], que se ha convertido en un estándar *de facto* dentro de la comunidad de mapas en Internet, como demuestra la cantidad de servidores, clientes y aplicaciones de utilidad que se han construido alrededor de esta especificación, incluyendo los principales clientes de visualización de mapas como Openlayers [3].
- La comunidad de desarrolladores de Geoserver ha desarrollado extensiones que pueden ser instaladas en este servidor para ofrecer los mapas como teselas en varios formatos, incluyendo el de Mapbox [4].
- Algunas empresas privadas también están empezando a incluir el soporte a esta especificación abierta en sus productos, como es el caso de ESRI [5].
- Herramientas de SIG comienzan a incluir la posibilidad de cargar teselas vectoriales. Éste es el caso, por ejemplo, de QGIS [6].

Parece que la apuesta por el teselado vectorial es clara, de manera que probablemente en un futuro próximo el uso de esta tecnología se encuentre bastante extendido, tal y como ha sucedido en los últimos años en el ámbito de los servicios de mapas basados en teselas ráster. Por este motivo, el Centro Nacional de Información Geográfica, como principal órgano responsable de la IDEE, está trabajando en la puesta en marcha de un servicio de mapas basado en teselas vectoriales y en la definición de una serie de recomendaciones para alinear la construcción de este tipo de servicios con

las guías técnicas definidas por la Directiva INSPIRE. El resto de este artículo resume el trabajo realizado hasta ahora en esta línea.

## PONIENDO EN MARCHA UN SERVICIO DE TESELAS VECTORIALES

Para la construcción de un servicio de mapas capaz de servir sus capas como teselas vectoriales se ha tomado como referencia la información del Mapa Base de España, que actualmente se está publicando mediante un servicio de mapas (WMS) [7] y un servicio teselado (WMTS) [8] conformes a las especificaciones de la Directiva INSPIRE para los servicios de visualización. El Mapa Base de España recoge la cartografía procedente de diversas bases de datos geográficas de diferentes escalas: información procedente de las Bases Topográficas (500, 50 y 25), NGBE (Nomenclátor Geográfico Básico de España) y SIGLIM (Sistema de Información Geográfica de Líneas Límite), información geográfica de referencia de hidrografía y redes de transporte, información de Cartocidad y el Modelo digital del terreno con paso de malla de 5 metros. Por este motivo, se trata de un ejemplo perfecto para demostrar la viabilidad de las teselas vectoriales, al combinar múltiples tipos de objetos geográficos, con geometrías diversas y a distintas escalas.

A la hora de seleccionar la tecnología para poner en marcha el servicio, existen hoy en día varias opciones. En [9] se recoge un estudio bastante detallado sobre diferentes alternativas actuales que permiten construir un servicio de este tipo. No obstante, a pesar de que en él no se menciona GeoServer, se ha optado por esta tecnología para la publicación del Mapa Base de España como teselas vectoriales. GeoServer<sup>1</sup> es un *framework* escrito en Java que permite a los usuarios compartir y editar información geográfica. Diseñado para tener la máxima interoperabilidad posible, publica datos utilizando estándares a partir de bases de datos espaciales y diversos formatos de ficheros tanto vectoriales como ráster. Además, se distribuye bajo licencia GNU General Public License (GPL).

La elección de Geoserver para la construcción del servicio de teselas vectoriales del Mapa Base de España se ha basado en los siguientes aspectos:

- Geoserver lleva integrada una versión de la solución libre de creación de teselas GeoWebCache<sup>2</sup>, por lo que no requiere la instalación de herramientas extra para generar las teselas. GeoWebCache además almacena eficientemente los datos del teselado vectorial. Dado que el estilo es realizado por el cliente, no por el servidor, GeoWebCache sólo necesita almacenar una tesela vectorial para todos los estilos diferentes.
- Geoserver es el *software* sobre el que se ha creado el actual servicio WMTS del Mapa Base de España del Instituto Geográfico Nacional de España lo que permite, por un lado, reaprovechar buena parte del trabajo de configuración realizado en la construcción del servicio original a la hora de definir el detalle de cada capa en función de la escala y, por otro, la publicación de un solo servicio capaz de servir una misma información como teselas ráster o vectoriales, según la petición realizada por los clientes de visualización.
- La configuración de la generación de teselas es relativamente sencilla y no requiere que el personal encargado del mantenimiento del servicio tenga conocimientos técnicos adicionales.

Para la incorporación en Geoserver de la gestión de teselas vectoriales, es necesario añadir una extensión creada por la comunidad de desarrolladores de la plataforma<sup>3</sup>. De este modo, se añade al servidor la capacidad de producir teselas vectoriales en tres formatos: GeoJSON, TopoJSON y MVT. Los dos primeros son formatos más legibles para el usuario, mientras que el último, implementación de la especificación de MapBox, es preferible para su utilización en aplicaciones finalistas.

Una vez instalada la extensión, simplemente hay que habilitar los formatos anteriores en aquellas capas que se desean servir como teselas vectoriales. Por lo tanto, al efectuar dicha operación sobre

---

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

<sup>2</sup> <http://www.geowebcache.org/>

<sup>3</sup> <http://docs.geoserver.org/latest/en/user/extensions/vectortiles/index.html>

las mismas capas del Mapa Base de España que hasta el momento se ofrecían como teselas ráster a través del WMTS, se obtiene al mismo tiempo su publicación como teselas ráster con las mismas restricciones de visualización en cuanto a niveles de escala que estuviesen ya configuradas.

A la hora de explotar el servicio, se puede hacer uso de las dos interfaces que ofrece el propio Geoserver: la interfaz WMTS, en la que la única diferencia con respecto a las teselas ráster será el formato solicitado para la respuesta, y la interfaz TMS [10], definida por OSGeo.

#### Ejemplo de petición WMTS:

```
http://[host]:[port]/wms?LAYERS=ignbasetodo&FORMAT=application/x-  
protobuf;type=mapbox-  
vector&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&STYLES=&SRS=EPSG:900913&BBOX=-  
20037508.34,10018754.17,-10018754.17,20037508.34&WIDTH=256&HEIGHT=256
```

#### Ejemplo de petición TMS:

```
http://[host]:[port]/tms/1.0.0/ignbasetodo@EPSG:900913@pbf/10/500/636.pbf
```

Ahora bien, a pesar de la sencillez a la hora de configurar el servicio, la utilización de Geoserver también ha desvelado algunas limitaciones.

La primera limitación está relacionada con la gestión de capas que realiza el servidor. En Geoserver, cada capa está vinculada a una única tabla en la base de datos. Además, se pueden crear agrupaciones de capas, de manera que con una única petición se sirven simultáneamente datos de diferentes capas individuales. Esta es la estrategia utilizada en el servicio del Mapa Base de España, en el que se tiene una capa agrupada, donde la visualización de las capas individuales que la componen se determina basándose en estilos. Así, se muestran geometrías más o menos detalladas para los mismos tipos de objeto geográfico en función de la escala de visualización: a unas escalas se muestran los elementos de las capas procedentes del BTN500 y, a otras, las de la BTN25. Esta característica, que constituye una clara ventaja a la hora de configurar el servicio, no lo es tanto en el momento de utilizarlo. El motivo es que para poder identificar un elemento dentro de una tesela vectorial, es necesario conocer el tipo de objeto geográfico al que pertenece y la identificación de dichos tipos que ofrece Geoserver no es otra que la denominación de las capas individuales que están configuradas en el servicio. Este hecho implica que no se tiene un único tipo de objeto geográfico, por ejemplo «carreteras», sino varios («carreteras25», «carreteras500», etc.) para identificar un mismo tipo de objeto del mundo real, lo cual complica la utilización del servicio por parte de los clientes web.

Aunque el problema anterior se podría solventar utilizando otra tecnología diferente para construir el servidor, se ha optado por trabajar en una solución que combine todos los mismos tipos de objetos geográficos de diferentes escalas en una misma tabla y restrinja la visibilidad de sus elementos a través de reglas de estilo basadas en atributos, de tal modo que solo exista una capa simple por tipo de objeto geográfico en el servicio y, de esta manera, sea más sencillo de explotar por los clientes de visualización.

La otra limitación que se ha encontrado en Geoserver es que no ofrece la interfaz XYZ [11], una de las más utilizadas, sino la que más, a la hora de utilizar mapas basados en teselas.

Del mismo modo que en el caso anterior, este problema se podría solucionar recurriendo a otra solución para poner en marcha el servidor que ofrezca la interfaz XYZ. Sin embargo, se ha optado por aprovechar todo el trabajo realizado con Geoserver mediante la construcción de un *proxy* que transforme las peticiones entre una interfaz y otra. Para ello, se ha partido del trabajo desarrollado por Chris Whong<sup>4</sup> que ofrece un *script* para publicar un servicio TMS como interfaz XYZ y se ha adaptado a los propósitos del experimento. El único cambio que requiere el paso de una interfaz a otra es la siguiente conversión en la coordenada Y:

$$y = (2^z) - y - 1$$

<sup>4</sup> <https://gist.github.com/chriswhong/f492cc317c661001f41bcf0795162717>

## PROPONIENDO RECOMENDACIONES PARA LOS SERVICIOS DE VISUALIZACIÓN EN INSPIRE

La guía técnica de INSPIRE para servicios de visualización [12] describe cómo usar WMS y WMTS en servicios INSPIRE, junto a ciertos aspectos relacionados con el rendimiento de servicio. A continuación se proponen una serie de recomendaciones sobre la inclusión de las teselas vectoriales en dicha guía técnica, que han sido trasladadas al Grupo de Implementación y Mantenimiento (MIG) de INSPIRE para su evaluación. Dado que las teselas vectoriales no cuentan todavía con un estándar único y asentado al que ajustarse, no consideramos oportuno incluir su recomendación dentro de una sección específica del documento, sino como un Anexo. En dicho Anexo se describiría cómo usar un formato de teselas vectoriales en las peticiones de un WMTS, así como en las interfaces TMS y XYZ.

Con respecto a los servicios WMTS, nuestras recomendaciones van dirigidas tanto a la operación «Get View Service Metadata» como a la operación «Get Map». De este modo, en referencia a la operación «Get View Service Metadata», la principal ampliación que debería revisarse es la relativa a los metadatos de la operación «GetTile». El requisito 82 de la última versión de la guía técnica establece que dichos metadatos deben documentarse con el siguiente elemento del documento de *Capabilities*:

```
<ows:Operation name="GetTile">
```

En dicho elemento se indica que al menos el formato PNG o el formato GIF debe ser ofrecido por el servicio. Consideramos que este requisito debería, por lo tanto, ser ampliado para dar cabida a los formatos de codificación de las teselas vectoriales. Aunque cualquier formato vectorial podría usarse para codificar este tipo de teselas, la especificación de Mapbox parece ser el estándar *de facto*, siendo además compatible con mucho *software* en uso. Por lo tanto, el requisito 82 podría extenderse indicando que, tal y como se describe en la especificación de Mapbox (versión 2.1), el tipo MIME para la operación GetTile debería ser *application/vnd.mapbox-vector-tile* y los ficheros deberían tener la extensión *.mvt*, aunque, dado que algunos servidores usan actualmente *aplicación/x-protobuf;type=mapbox-vector* y *.pbf* como extensión, sería interesante también contemplar esa opción. Adicionalmente, hay otros formatos alternativos que tienen al menos algún uso y el requisito 82 podría también incluirlos: GeoJSON, cuyo tipo MIME es *application/json;type=geojson* y la extensión es usualmente *.json* o *.geojson*; TopoJSON, cuyo tipo MIME es *application/json;type=topojson* y la extensión del fichero suele ser *.json* o *.topojson*.

Dentro de la operación «Get View Service Metadata» también proponemos revisar las recomendaciones relativas a estilos y leyendas, valorando su cabida dentro de las teselas vectoriales.

La definición del estilo se establece actualmente en el requisito 90. Los estilos tienen una interpretación diferente en las teselas vectoriales con respecto a la que se da para las teselas ráster, ya que el estilo en este caso es responsabilidad del cliente. Una opción podría ser suponer que, si se solicita un cierto estilo para un determinado tipo de objeto geográfico, en este contexto significa que la tesela vectorial contiene toda la información necesaria para permitir la aplicación de dicho estilo en el cliente. Por ejemplo, si la representación de ciertos elementos se basa en el atributo «tipo de camino», las teselas vectoriales que se obtengan, deberían contener este atributo para poder aplicar el estilo correspondiente. Los detalles pueden considerarse fuera del alcance de la guía técnica de los servicios de visualización, ya que dependen en gran medida de los tipos de datos definidos en otros documentos INSPIRE.

Con respecto a las leyendas, actualmente quedan recogidas en el requisito 91. Las leyendas definidas por el servidor no tienen sentido si se usan las teselas vectoriales, ya que pueden y deben definirse en las aplicaciones cliente. Dado que este elemento es opcional, nuestra recomendación es no incluir ninguna leyenda para las teselas vectoriales en el lado del servidor.

En cuanto a la operación «Get Map», las directrices INSPIRE la emparejan con la operación «GetTile» de la especificación WMTS de OGC. Sugerimos extender los valores permitidos que se pueden usar en el parámetro FORMAT para dar cabida a las teselas vectoriales. Estos valores serían los mismos que se han comentado en el caso del requisito 82:

- aplicación / vnd.mapbox-vector-tile

- application / x-protobuf; type = mapbox-vector
- aplicación / json; type = geojson
- aplicación / json; type = topojson

Dentro de las directrices INSPIRE para los servicios de visualización, el actual anexo A introduce algunas recomendaciones sobre la utilización de la interfaz TMS. Proponemos que estas recomendaciones se extiendan con el objeto de incluir el uso de teselas vectoriales a través de dicha interfaz. Del mismo modo que para los servicios WMTS, en el caso de la interfaz TMS sería preciso establecer ciertas pautas para las operaciones «Get View Service Metadata» y «Get Map».

La operación «Get View Service Metadata» podría ser la predeterminada al acceder a la URL base del servicio. La respuesta sería un documento que proporcione metadatos de descripción sobre el servicio (elemento *TileMapService*) y enumere los tipos de teselas disponibles (*TileMaps*).

```
<TileMapService version="1.0.0" services="http://[host]:[port]/">

  <Title>Mapa Base de España</Title>

  <Abstract>Cartografía procedente de diversas bases de datos geográficas del
  IGN: Base Cartográfica Nacional escala 1:500.000 (BCN500), Base Topográfica Nacional
  escala 1:100.000 (BTN100), Base Topográfica Nacional 1:25.000 (BTN25), Nomenclátor
  Geográfico Básico de España (NGBE), Sistema de Información Geográfica de Líneas
  Límite (SIGLIM), Información Geográfica de Referencia (IGR) del IGN en materia de
  transportes y CartoCiudad.</Abstract>

  <TileMaps>

    <TileMap title="Todas las capas de IGNBase" srs="EPSG:4326"
    profile="global-geodetic"
    href="http://[host]:[port]/tms/1.0.0/IGNBaseTodo@EPSG%3A4326@utfgrid"/>

    <TileMap title="Todas las capas de IGNBase" srs="EPSG:4326"
    profile="global-geodetic"
    href="http://[host]:[port]/tms/1.0.0/IGNBaseTodo@EPSG%3A4326@png"/>

    <TileMap title="Todas las capas de IGNBase" srs="EPSG:4326"
    profile="global-geodetic"
    href="http://[host]:[port]/tms/1.0.0/IGNBaseTodo@EPSG%3A4326@jpeg"/>

    <TileMap title="Todas las capas de IGNBase" srs="EPSG:4326"
    profile="global-geodetic"
    href="http://[host]:[port]/tms/1.0.0/IGNBaseTodo@EPSG%3A4326@geojson"/>

    <TileMap title="Todas las capas de IGNBase" srs="EPSG:4326"
    profile="global-geodetic"
    href="http://[host]:[port]/tms/1.0.0/IGNBaseTodo@EPSG%3A4326@topojson"/>

    <TileMap title="Todas las capas de IGNBase" srs="EPSG:4326"
    profile="global-geodetic"
    href="http://[host]:[port]/tms/1.0.0/IGNBaseTodo@EPSG%3A4326@pbf"/>

    <TileMap title="Todas las capas de IGNBase" srs="EPSG:900913"
    profile="global-mercator"
    href="http://[host]:[port]/tms/1.0.0/IGNBaseTodo@EPSG%3A900913@utfgrid"/>

    <TileMap title="Todas las capas de IGNBase" srs="EPSG:900913"
    profile="global-mercator"
    href="http://[host]:[port]/tms/1.0.0/IGNBaseTodo@EPSG%3A900913@png"/>

    <TileMap title="Todas las capas de IGNBase" srs="EPSG:900913"
    profile="global-mercator"
    href="http://[host]:[port]/tms/1.0.0/IGNBaseTodo@EPSG%3A900913@jpeg"/>

  </TileMaps>

</TileMapService>
```

```

        <TileMap title="Todas las capas de IGBase" srs="EPSG:900913"
profile="global-mercator"
href="http://[host]:[port]/tms/1.0.0/IGBaseTodo@EPSG%3A900913@geojson"/>

        <TileMap title="Todas las capas de IGBase" srs="EPSG:900913"
profile="global-mercator"
href="http://[host]:[port]/tms/1.0.0/IGBaseTodo@EPSG%3A900913@topojson"/>

        <TileMap title="Todas las capas de IGBase" srs="EPSG:900913"
profile="global-mercator"
href="http://[host]:[port]/tms/1.0.0/IGBaseTodo@EPSG%3A900913@pbf"/>

    </TileMaps>
</TileMapService>

```

Cada *TileMap* debería ofrecer un enlace a su documento de metadatos, donde se proporcione el tipo MIME y la extensión de fichero para cada tesela (a través del elemento *TileFormat*) y la lista de niveles disponibles.

```

<TileMap version="1.0.0" tilemapservice="http://[host]:[port]/tms/1.0.0">

    <Title>Todas las capas de IGBase</Title>

    <Abstract/>

    <SRS>EPSG:4326</SRS>

    <BoundingBox minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"/>

    <Origin x="-180.0" y="-90.0"/>

    <TileFormat width="256" height="256" mime-type="application/x-protobuf"
extension="pbf"/>

    <TileSets profile="global-geodetic">

        <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/0" units-per-
pixel="0.703125" order="0"/>

        <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/1" units-per-
pixel="0.3515625" order="1"/>

        <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/2" units-per-
pixel="0.17578125" order="2"/>

        <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/3" units-per-
pixel="0.087890625" order="3"/>

        <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/4" units-per-
pixel="0.0439453125" order="4"/>

        <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/5" units-per-
pixel="0.02197265625" order="5"/>

        <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/6" units-per-
pixel="0.010986328125" order="6"/>

        <TileSet

```

```

href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/7" units-per-
pixel="0.0054931640625" order="7"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/8" units-per-
pixel="0.00274658203125" order="8"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/9" units-per-
pixel="0.001373291015625" order="9"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/10" units-per-
pixel="6.866455078125E-4" order="10"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/11" units-per-
pixel="3.4332275390625E-4" order="11"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/12" units-per-
pixel="1.71661376953125E-4" order="12"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/13" units-per-
pixel="8.58306884765625E-5" order="13"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/14" units-per-
pixel="4.291534423828125E-5" order="14"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/15" units-per-
pixel="2.1457672119140625E-5" order="15"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/16" units-per-
pixel="1.0728836059570312E-5" order="16"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/17" units-per-
pixel="5.364418029785156E-6" order="17"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/18" units-per-
pixel="2.682209014892578E-6" order="18"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/19" units-per-
pixel="1.341104507446289E-6" order="19"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/20" units-per-
pixel="6.705522537231445E-7" order="20"/>

    <TileSet
href="http://[host]:[port]/tms/1.0.0/ignbasetodo2@EPSG%3A4326@pbf/21" units-per-
pixel="3.3527612686157227E-7" order="21"/>

    </TileSets>
</TileMap>

```

En cuanto a la operación GetMap, se realizaría a través de una petición HTTP GET con la siguiente estructura:

```
http://{TileMap URL}/{z}/{x}/{y}.{extension}
```



Donde {z} es el nivel de zoom, {x} es la fila de la tesela, {y} es la columna de la tesela y {extensión} es el formato de respuesta (que es específico para cada {TileMap URL}). En el caso de las teselas vectoriales, los valores permitidos para el parámetro de extensión podrían ser mvt, pbf, geojson o topojson.

Finalmente, creemos conveniente que las directrices INSPIRE hagan alguna referencia al protocolo XYZ, por ser éste uno de los más ampliamente utilizados, por ejemplo, por el propio Google o Mapbox. Dado que la utilización de este protocolo es bastante simple, podría ser suficiente con mencionarla y hacer alusión a las características que viene recogidas en la propia especificación [11]:

- Las teselas son ficheros de 256 × 256 píxeles.
- Cada nivel de zoom es un directorio, cada columna es un subdirectorio y cada fila en esa columna es un fichero
- El formato de petición es {url}/zoom/x/y.pbf

Por último, en cuanto a la calidad de los servicios, las teselas vectoriales son generalmente más pequeñas que las teselas ráster, por lo que el rendimiento, la capacidad y la disponibilidad deberían ser similares o mejores, ya que los demás aspectos son iguales, especialmente los referidos al uso de preprocesamiento y al almacenamiento en caché. Las teselas vectoriales requieren más potencia de renderización por parte del cliente, pero, por un lado, las aplicaciones cliente están fuera del alcance de las directrices INSPIRE y, por otro lado, en la actualidad, incluso los dispositivos móviles de bajo coste son lo suficientemente potentes como para soportar esa carga de trabajo.

## DEFINIENDO ESTILOS EN EL CLIENTE DE VISUALIZACIÓN

Como se ha expuesto anteriormente, a diferencia de las teselas ráster, en las que el estilo de visualización se aplica en el servidor, en las teselas vectoriales la visualización de los datos se realiza en el cliente (aplicación, móvil o de escritorio, o navegador de Internet), dando así total flexibilidad a los desarrolladores de aplicaciones a la hora de crear estilos para sus mapas. El principal problema que deben afrontar esos desarrolladores viene impuesto por la biblioteca de visualización utilizada, en la que la configuración de los estilos puede ser más o menos difícil en función de la opción elegida. Sería interesante, por lo tanto, contar con un mecanismo estándar que facilite la definición de estilos de las teselas vectoriales y que sea aceptado por las diferentes bibliotecas de visualización, algo similar a lo que ocurre con la especificación *Styled Layer Descriptor* (SLD) [13] y los distintos *frameworks* para la publicación de información geográfica mediante servicios web estándar (GeoServer, Deegree, Mapserver, etc).

Mapbox, como principal proveedor en la actualidad de servicios web de mapas basados en teselas vectoriales y una de las empresas que más apuestan por esta tecnología, es consciente de este problema y por ello ha desarrollado una especificación abierta para la creación de documentos de definición de estilos en formato JSON. Esta especificación, denominada Mapbox Style Specification [14], describe la estructura que deben tener estos documentos, en los que se indica, entre otros aspectos, las direcciones de los servidores de mapas, las capas que se desean dibujar, cómo hacerlo (colores, grosores, utilización de iconos, etc) y cuándo (escalas máximas y mínimas, filtros en función de atributos de las entidades geográficas, etc.). Se trata de una especificación bastante potente que ofrece muchas posibilidades de configuración, por lo que, en cierto modo, podría resultar difícil de utilizar para aquellos desarrolladores no expertos en la materia.

Para facilitar la creación de documentos conforme a la especificación de Mapbox, existe un proyecto de *software* libre, denominado Maputnik<sup>5</sup>, en el que se ofrece un editor gráfico para la definición de estilos sobre teselas vectoriales. Este editor puede instalarse localmente o utilizarse en línea. Permite dar de alta nuevos servicios de mapas y editar la configuración específica de cada una de sus capas. Una vez terminada la edición, se puede exportar el resultado a formato JSON (véase la Figura 1).

---

<sup>5</sup> <https://github.com/maputnik/editor>

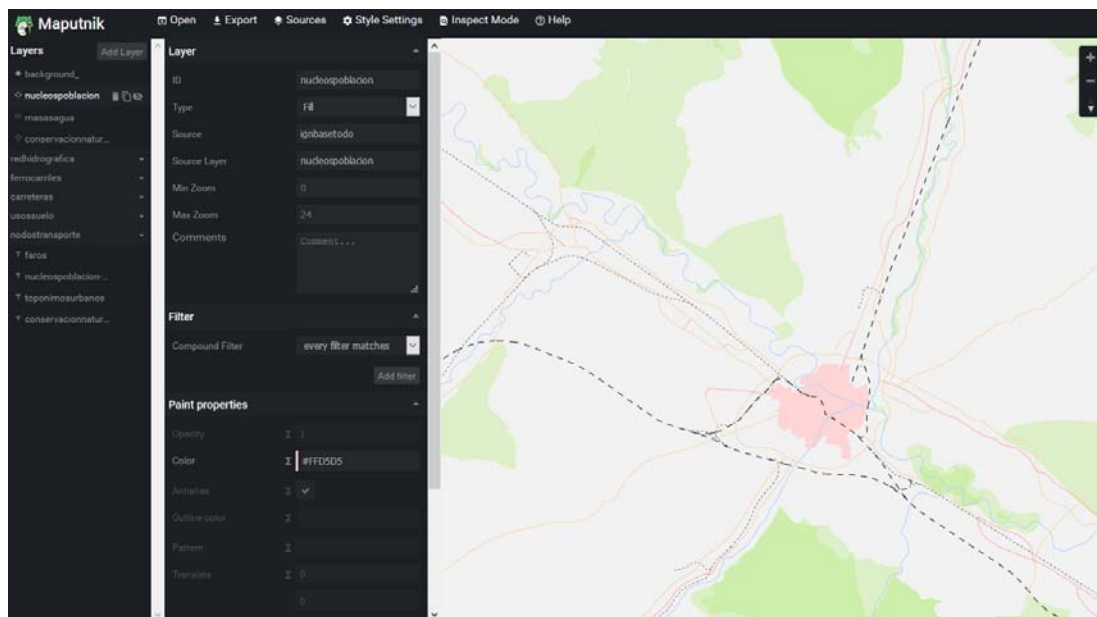


Figura 1: Maputnik - Editor de estilos en JSON

El principal problema que tiene el editor de Maputnik es que necesita como fuente de datos un servicio de mapas con interfaz XYZ. Por lo tanto, si el servicio de mapas que se desea utilizar sólo ofrece la interfaz WMTS o TMS, no se podrá emplear directamente como fuente de datos de Maputnik. Sería necesario, por ejemplo, poner en marcha un *proxy* como el que se ha comentado anteriormente que transforme las peticiones desde una interfaz a otra.

Una vez creado el fichero JSON con los estilos, se puede utilizar directamente para configurar un visualizador creado con Mapbox GL J. Aunque de momento otras bibliotecas de visualización no soportan esto de manera nativa, las más habituales cuentan con algunos *plugins* desarrollados para cargar estilos de Mapbox, con mayor o menor acierto, que se espera sigan evolucionando para permitir esta funcionalidad, cada vez más demandada.

Dentro de las pruebas realizadas con el Mapa Base de España, se ha utilizado Maputnik para definir estilos de visualización y se han probado en diferentes clientes de visualización con los siguientes resultados:

- Mapbox GL JS<sup>6</sup>: como era de esperar, la biblioteca reconoce perfectamente el fichero JSON de estilos y lo representa correctamente.
- Leaflet + *plugin*<sup>7</sup>: el resultado es bastante bueno y no se aprecian apenas diferencias con respecto a Mapbox GL JS.
- OpenLayers 4 + *plugin*<sup>8</sup>: el *plugin* tiene problemas al cargar ciertos servicios de teselas vectoriales. Además, la forma en la que está implementado el *plugin* dificulta la configuración del visor, ya que crea por sí mismo un mapa de forma asíncrona y por tanto no se le pueden definir las propiedades iniciales (el área visible, por ejemplo). El *plugin* tampoco avisa de cuándo ha finalizado esta carga, por lo que no se pueden definir a posteriori, al menos fácilmente, dichas propiedades.

<sup>6</sup> <https://www.mapbox.com/mapbox-gl-js/api/>

<sup>7</sup> <https://github.com/mapbox/mapbox-gl-leaflet>

<sup>8</sup> <https://github.com/boundlessgeo/ol-mapbox-style>

## CONCLUSIONES

En este artículo se ha presentado el trabajo realizado para poner en marcha un servicio de teselas vectoriales a partir del Mapa Base de España. El prototipo se ha desarrollado a partir de la herramienta de *software* libre Geoserver, que ha dado muy buenos resultados. No obstante, se han detectado algunas limitaciones y se han expuesto las soluciones adoptadas para superarlas.

Adicionalmente, se recogen en el artículo una serie de indicaciones que se han plasmado en la redacción de un primer borrador de recomendaciones sobre la utilización de servicios de mapas basados en teselas vectoriales que será sometido al Grupo de Implementación y Mantenimiento (MIG) de INSPIRE para su evaluación y consideración como nuevo punto a añadir dentro de futuras revisiones de las Normas de Ejecución de servicios de visualización de la Directiva.

Finalmente, se han puesto en marcha algunos prototipos de visualizadores, basados en diferentes bibliotecas para la construcción de clientes web, que permiten explotar el servicio desarrollado a partir de una definición de estilos basada en una especificación abierta.

En conclusión, se podría decir que las teselas vectoriales son una realidad que cada vez tiene más aceptación y para las que día a día van surgiendo nuevas soluciones técnicas y tecnológicas que mejoran sus prestaciones y facilitan, en gran medida, su manejo y puesta en marcha. Si la tecnología viene apoyada por la estandarización, gracias en parte a las recomendaciones que pueden hacerse a través de directrices técnicas que describan las características principales de este tipo de servicios como las proporcionadas por la Directiva INSPIRE, es muy posible que las teselas vectoriales se conviertan en poco tiempo en la manera habitual de servir mapas en Internet.

## AUTORES

Emilio LÓPEZ

*elromero@fomento.es*  
Centro Nacional de Información  
Geográfica

Rubén BÉJAR

*rbejar@unizar.es*  
Universidad Zaragoza  
Instituto de Investigación en  
Ingeniería de Aragón

Jesús BARRERA

*jesusb@geoslab.com*  
GeoSLab

Francisco J LOPEZ-PELLICER

*fjlopez@unizar.es*  
Universidad Zaragoza  
Instituto de Investigación en  
Ingeniería de Aragón

Antonio F. RODRÍGUEZ

*afrodriguez@fomento.es*  
Centro Nacional de Información  
Geográfica

Paloma ABAD

*pabad@fomento.es*  
Centro Nacional de Información  
Geográfica

## REFERENCIAS

- [1] Open Street Map: Vector Tiles, [http://wiki.openstreetmap.org/wiki/Vector\\_tiles](http://wiki.openstreetmap.org/wiki/Vector_tiles) [consultado el 19-10-2017]
- [2] Mapbox: Mapbox Vector Tile Specification: a guide to encoding tiled vector data with the Mapbox Vector Tile Specification, <https://www.mapbox.com/vector-tiles/specification/> [consultado el 19-10-2017]
- [3] GitHub: Awesome Vector Tiles, <https://github.com/mapbox/awesome-vector-tiles> [consultado el 19-10-2017]
- [4] GitHub: Extensions Vector Tiles, <https://github.com/geoserver/geoserver/tree/master/src/extension/vectortiles> [consultado el 19-10-2017]
- [5] Turner, A.: Vector Tiles Preview. In: ARCGIS blog, July 20 2015, <https://blogs.esri.com/esri/arcgis/2015/07/20/vector-tiles-preview/> [consultado el 19-10-2017]
- [6] GitHub: Vector tiles reader QGIS plugin, <https://github.com/geometalab/Vector-Tiles-Reader->

[QGIS-Plugin](#) [consultado el 19-10-2017]

- [7] Open Geospatial Consortium: Web Map Service, <http://www.opengeospatial.org/standards/wms> [consultado el 19-10-2017]
- [8] Open Geospatial Consortium: OpenGIS Web Map Tile Service Implementation Standard, <http://www.opengeospatial.org/standards/wmts> [consultado el 19-10-2017]
- [9] Norman, P.: Serving vector tiles. In: Paul's blog, November 2016, <http://www.paulnorman.ca/blog/2016/11/serving-vector-tiles/> [consultado el 19-10-2017]
- [10] OsGeo: Tile Map Service Specification, [http://wiki.osgeo.org/wiki/Tile\\_Map\\_Service\\_Specification](http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification) [consultado el 19-10-2017]
- [11] Open Street Map: Slippy map tilenames, [http://wiki.openstreetmap.org/wiki/Slippy\\_map\\_tilenames](http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames) [consultado el 19-10-2017]
- [12] Initial Operating Capability Task Force Network Services: Technical Guidance for the implementation of INSPIRE View Services: version 3.11, April 4 2013, <http://inspire.ec.europa.eu/documents/technical-guidance-implementation-inspire-view-services-1> [consultado el 19-10-2017]
- [13] Open Geospatial Consortium: Styled Layer Descriptor, <http://www.opengeospatial.org/standards/sld> [consultado el 19-10-2017]
- [14] Mapbox: Mapbox Style Specification, <https://www.mapbox.com/mapbox-gl-js/style-spec/> [consultado el 19-10-2017]