



**CURSO DE EXPERTO EN DESARROLLO Y GESTIÓN DE  
SISTEMAS DE INFORMACIÓN GEOGRÁFICA (31-EX-75)**

*Estudios propios de la Universidad de Cantabria*

**MATERIALES DOCENTES**

*Introducción a MapGuide Open Source*

**ASIGNATURA 11**

**PUBLICACIÓN EN SERVIDORES DE MAPAS:  
MAPGUIDE OPEN SOURCE**



**EMILIO GÓMEZ FERNÁNDEZ**

emiliogf@altergeosistemas.com

## Índice de contenido

1. Introducción.....	3
2. Arquitectura de MapGuide.....	4
3. Instalando MapGuide.....	5
4. MapGuide Maestro.....	10
4.1 El entorno de trabajo.....	12
4.2 Las fuentes de datos.....	17
4.3 Las capas de datos.....	23
4.4 El Mapa.....	38
4.5 El diseño del visor cartográfico.....	50
4.5 La aplicación Fusion.....	55
5. Generar el paquete de datos y cargarlo en el servidor.....	57
6 Cargar un paquete en el servidor de mapas.....	58
7. Visualizar el proyecto en el navegador.....	60
8. Enlaces de interés.....	63



Este obra está bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 España.

## 1. Introducción

MapGuide Open Source (en adelante simplemente MapGuide) es una plataforma geoespacial basada en servidor que facilita al usuario desarrollar e implementar de forma rápida aplicaciones webs cartográficas para la visualización y consulta de mapas y datos espacialmente georreferenciados en un entorno web. Permite la creación de aplicaciones web enriquecidas con funcionalidades que, por lo general, únicamente están disponibles en Sistemas de Información Geográfica (SIG) de escritorios.

MapGuide posibilita crear visores cartográficos con funciones como la selección de elementos, la consulta de atributos, creación de áreas de influencia (*buffers*), mediciones, etc. En el lado del servidor MapGuide incluye una base de datos XML para almacenar y administrar el contenido, soportando los formatos de archivos espaciales, bases de datos y estándares más conocidos. La plataforma MapGuide se puede implementar bajo sistemas operativos Linux y Microsoft Windows, siendo compatible con los servidores web Apache e Internet Information Service (IIS). Así mismo, su Interfaz de Programación de Aplicaciones (API) permite desarrollar aplicaciones web en lenguajes de programación PHP, .Net, Java y JavaScript.

MapGuide fue inicialmente desarrollado por la empresa Autodesk -muy conocida por su software de Diseño Asistido por Ordenador AutoCAD, entre otros- bajo el nombre de Autodesk MapGuide. Hasta el año 2006 fue un servidor de mapas comercial. Es en ese año cuando Autodesk publica Autodesk MapGuide Enterprise 2007 bajo una licencia de código libre LGPL. Este cambio de postura de la empresa creadora del software se encuentra, entre otras, en el echo de que Autodesk MapGuide 6.5, la última versión antes del cambio de licencia, empezaba a ser obsoleta al implementar un visor basado en ActiveX, por lo que únicamente funcionaba con el navegador web Microsoft Internet Explorer y obviaba al cada vez mayor número de usuarios de otros navegadores. Así mismo, el avance de nueva tecnología para la visualización de datos geográficos que irrumpían con fuerza -como es el caso de los servicios OGC como el WMS o el WFS- o el uso cada vez mayor de AJAX, hacía demasiado costoso actualizar este software por lo que decidieron desarrollar una versión de MapGuide totalmente nueva.

En la actualidad existen dos versiones de MapGuide. La ya señalada Autodesk MapGuide Enterprise en su versión comercial, y por otro lado MapGuide Open Source como software libre. Cabe resaltar que ambas versiones son prácticamente el mismo servidor de mapas y únicamente se diferencian en que sobre la versión *open source* Autodesk ha añadido unas características adicionales que básicamente son el uso de bibliotecas de código propietario para el acceso a algunas fuentes en sustitución a las de código libre usadas en la versión *open source*, la implementación de soporte para la visualización de archivos DWG -el cual sigue siendo un formato CAD cerrado de

Autodesk- o la posibilidad de poder añadir desde el *back-end*<sup>1</sup> servidores de apoyo al servidor cartográfico principal.

MapGuide permite acceder a datos espaciales vectoriales ESRI ArcSDE, MySQL, Microsoft SQL Server, PostgreSQL/PostGIS, SDF, Shapefile, ODBC, DWF, DWG, WFS, WMS y Oracle Spatial a través de la tecnología FDO data access, además de los formatos soportados mediante OGR.

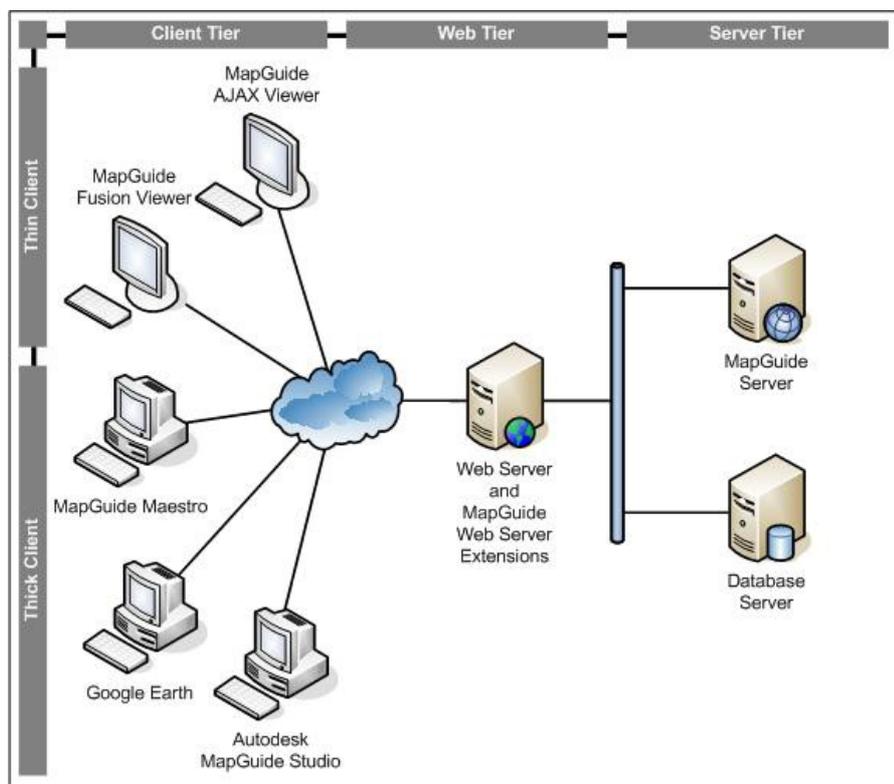
En cuanto a capas ráster se puede acceder a archivos BMP, CAL, ECW, JPG, PNG, SID, TGA y TIF, además de los soportados por la librería GDAL (estos últimos en solo lectura).

Las imágenes de salida permitidas al generar las teselas del mapa son JPG, PNG, PNG8 y GIF.

Los estándares OGC de salida soportados son WMS, WFS y GML.

## 2. Arquitectura de MapGuide

De forma muy resumida podemos decir que MapGuide está desarrollado por tres componentes que a su vez conforman tres niveles:



Fuente: <http://mapguide.osgeo.org>

- 1 En diseño web *back-end* hace referencia a las páginas de gestión del sitio web, a las cuales solo está permitido el ingreso a los administradores, en contraposición al *front-end* o parte pública que puede ver cualquier usuario que acceda al sitio.

Por un lado la parte destinada al servidor de mapas. Consta de uno o más servidores MapGuide que realizan todo el trabajo pesado, tales como la consulta y actualización de datos, la representación del mapa, etc. En este nivel se pueden agregar tantos servidores adicionales como fuesen necesarios, en función del número de usuarios simultáneos que se conectarán al servicio. Aunque no es obligatoria por lo general la configuración típica suele constar de un servidor MapGuide destinado a ofrecer el mapa propiamente dicho y por otro lado un servidor donde se almacena la base de datos espacial y a la que se conecta MapGuide para solicitar la información geográfica.

Por otro lado está el nivel destinado al servidor web. Son uno o varios servidores que trabajan en colaboración para recoger la peticiones de información de los usuarios y las procesan con ayuda de los servidores MapGuide.

Y finalmente el nivel de los dispositivos clientes que se conectan a MapGuide para visualizar la cartografía. El usuario visualiza la cartografía y hace peticiones a MapGuide a través de un visor de mapas AJAX<sup>2</sup>. La tecnología AJAX, tan en boga actualmente en el desarrollo web, permite al navegador realizar peticiones asincrónicas al servidor. A efectos prácticos para un usuario esto le permite solicitar información al servidor en segundo plano, sin bloquear la página y tener que esperar a que esta se recargue con cada petición.

También existe un visor basado en el formato de Autodesk DWF (Design Web Format) pero que al requerir de ActiveX tiene un uso limitado al solo poder utilizarse esta tecnología en el navegador Microsoft Internet Explorer como ya hemos comentado, por lo que su uso no está muy extendido.

### 3. Instalando MapGuide



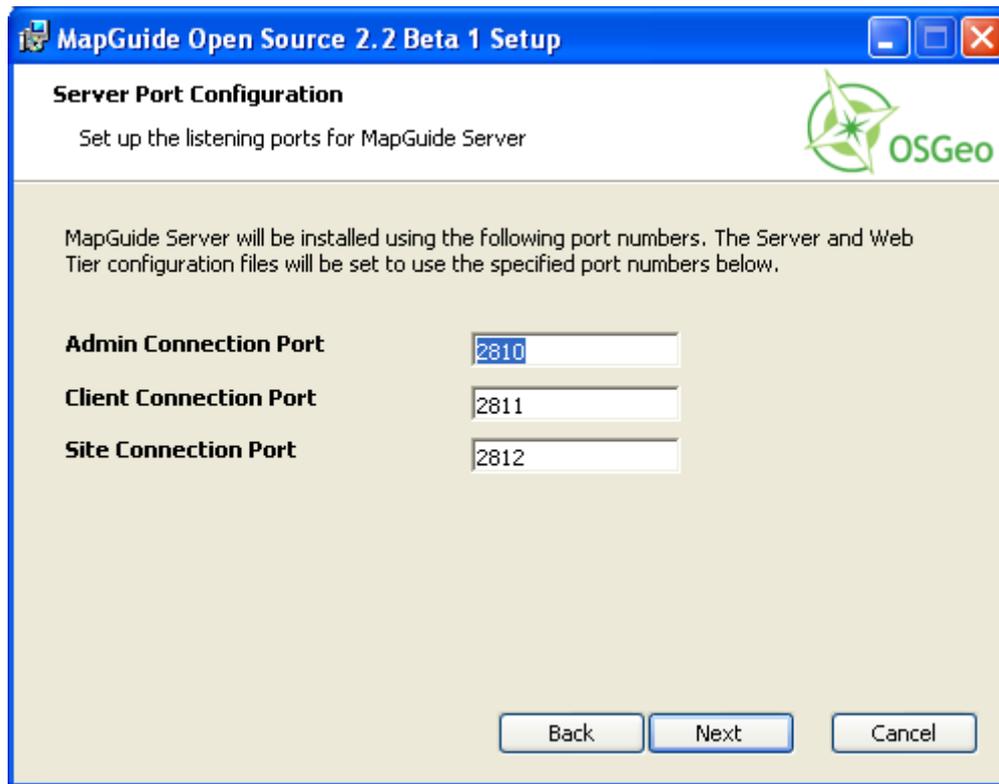
Descarga MapGuide Open Source desde el siguiente enlace: <http://mapguide.osgeo.org/download>. En este tutorial utilizaremos la versión MapGuide Open Source 2.2 Beta.

En el primer paso tendremos que elegir el tipo de servidor web sobre el que queremos instalar MapGuide. Si tenemos instalado *Internet Information Service* (IIS) en nuestro ordenador se nos dará la opción de elegir entre este o Apache. En nuestro caso elegiremos el servidor de código abierto Apache.

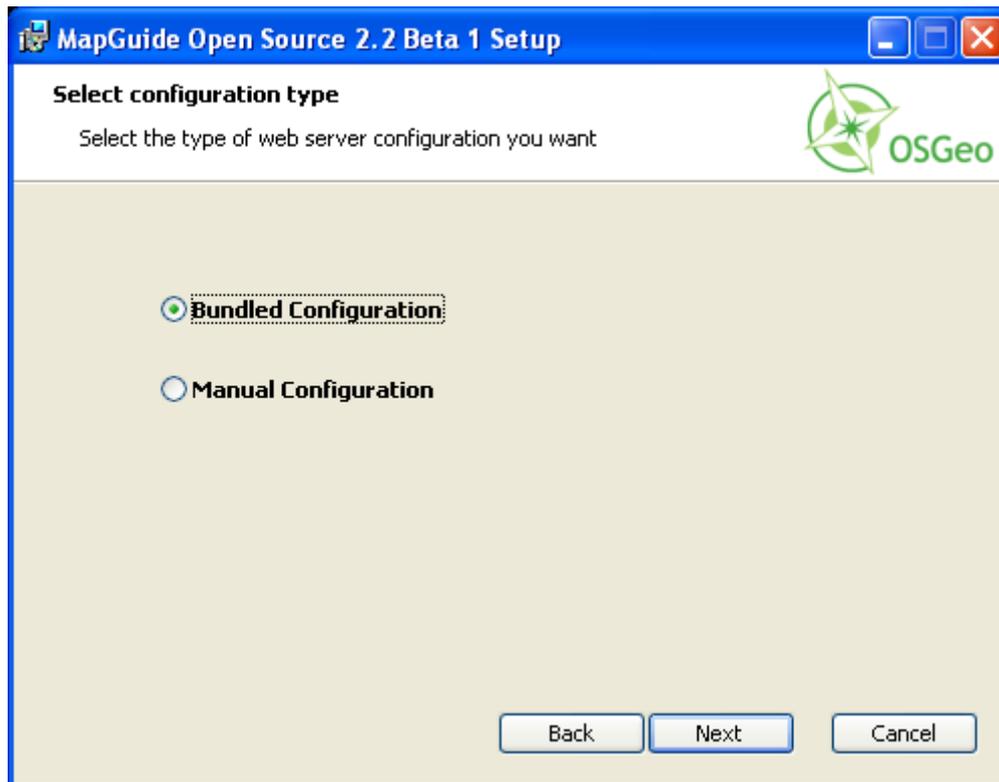
A continuación nos informará de los puertos que usará el servidor de mapas MapGuide. No modificaremos ninguno y continuaremos adelante.

---

<sup>2</sup> También conocido como DHTML o HTML dinámico y que es básicamente HTML + Javascript

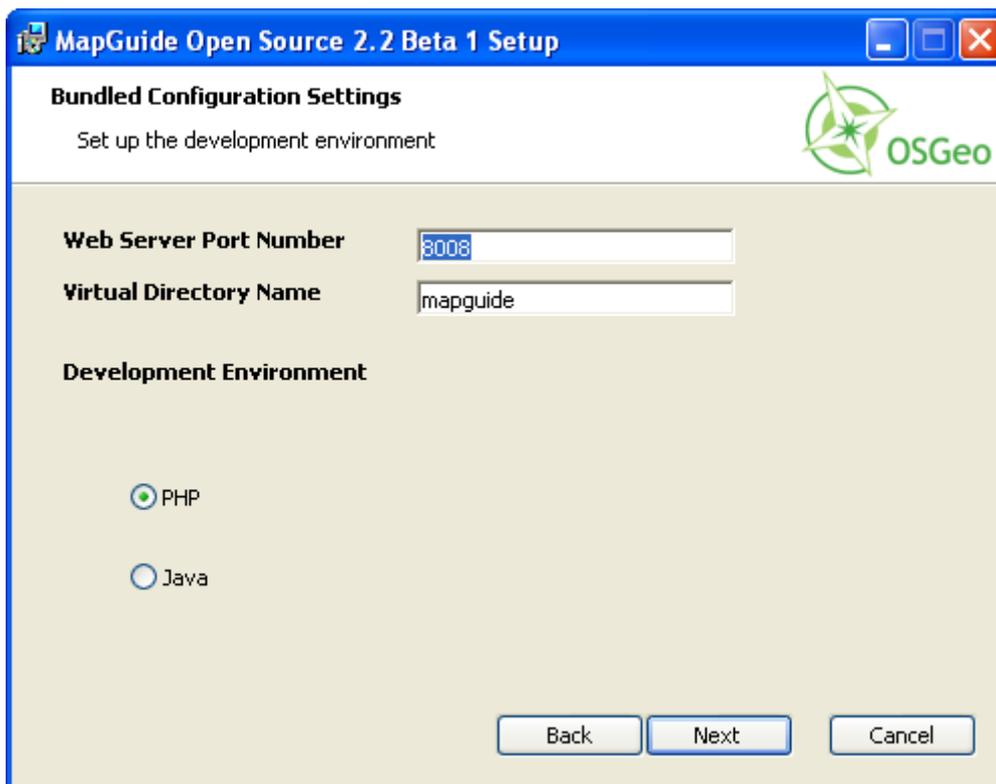


En la siguiente ventana tendremos que elegir entre una configuración de instalación ya integrada de Apache, PHP y Tomcat (Bundled Configuration) o si queremos instalar y configurar manualmente MapGuide (Manual Configuration). Elegiremos la primera opción.



En el siguiente paso se señala el número de puerto que en este caso utilizará el servidor web Apache (8008) y el nombre del directorio virtual (mapguide). Utilizaremos los que vienen por defecto. Además, el asistente de instalación preguntará por el lenguaje de programación para el entorno de desarrollo, donde seleccionaremos PHP.<sup>3</sup>

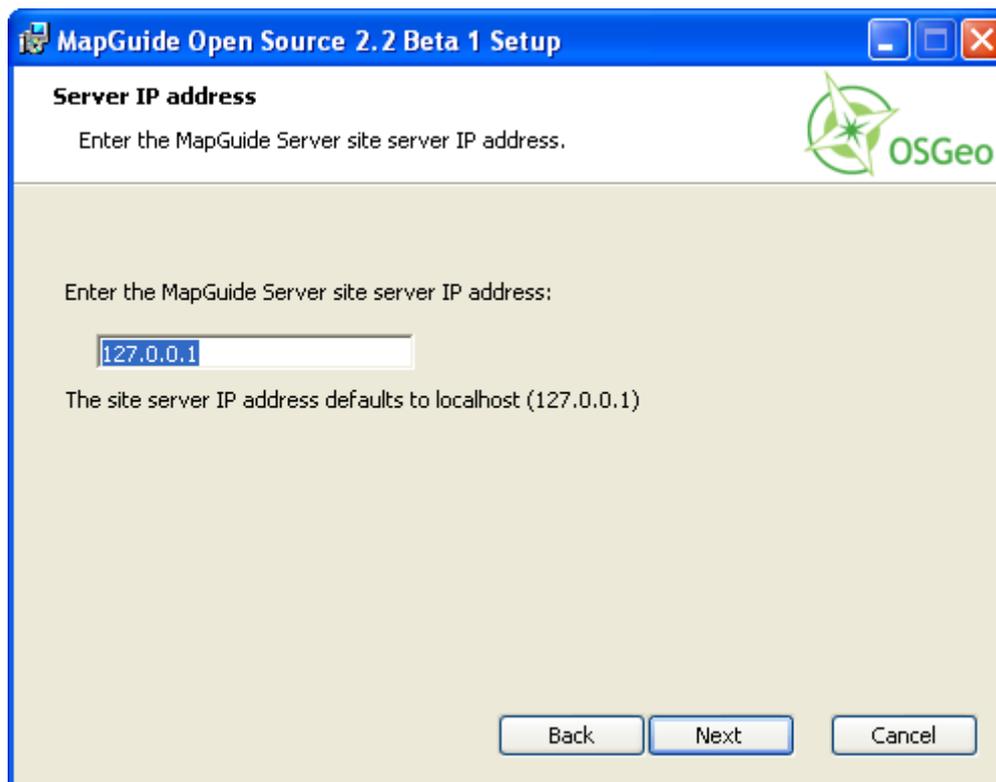
<sup>3</sup> PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.



A continuación se nos pide la IP del servidor MapGuide. Mantendremos la que nos ofrece por defecto.



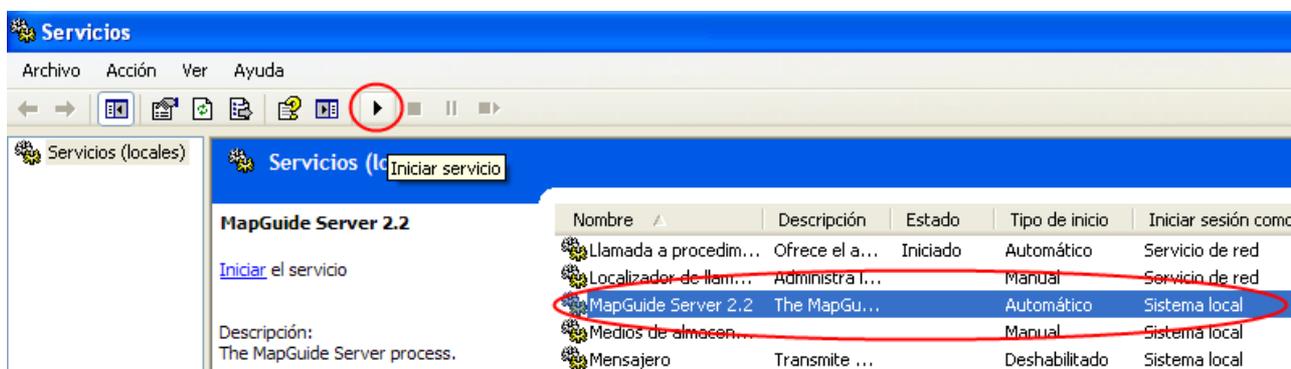
Tanto la IP ficticia 127.0.0.1 como la palabra reservada *localhost* son formas equivalentes de referirnos de manera genérica a la IP local del ordenador en que estamos trabajando.



En las siguientes ventanas del asistente conservaremos la configuración predefinida hasta el final.

Una vez terminada la instalación se pedirá reiniciar el sistema.

Tras reiniciar Windows verificaremos que tanto el servidor Apache como el servidor MapGuide están arrancados. Para ellos iremos al menú **Inicio** → **Panel de Control** → **Herramientas administrativas** → **Servicios** y comprobaremos que en los servicios **ApacheMapGuide** y **MapGuide Server** la columna *Estado* está marcada como *Iniciado*. Si no es así seleccionamos el servicio y pulsamos en el icono *Iniciar servicio* del menú superior.





Si queremos que el servicio se inicie automáticamente cada vez que se arranque el ordenador deberemos abrir las propiedades del servicio pulsando el botón derecho del ratón. En la pestaña *General* verificaremos que el *Tipo de inicio* es *Automático* y que el *Estado del servicio* está *Iniciado*. Sino pulsaremos el botón *Iniciar*.

Por último comprobaremos que, efectivamente, el servidor Apache está instalado y activado correctamente. Para ello abriremos el navegador web e introduciremos la siguiente dirección URL: <http://localhost:8008>



**It works!**

Como vemos en la pantalla anterior, la instalación de Apache se produjo de forma adecuada, así pues hemos completado este apartado satisfactoriamente.

## 4. MapGuide Maestro

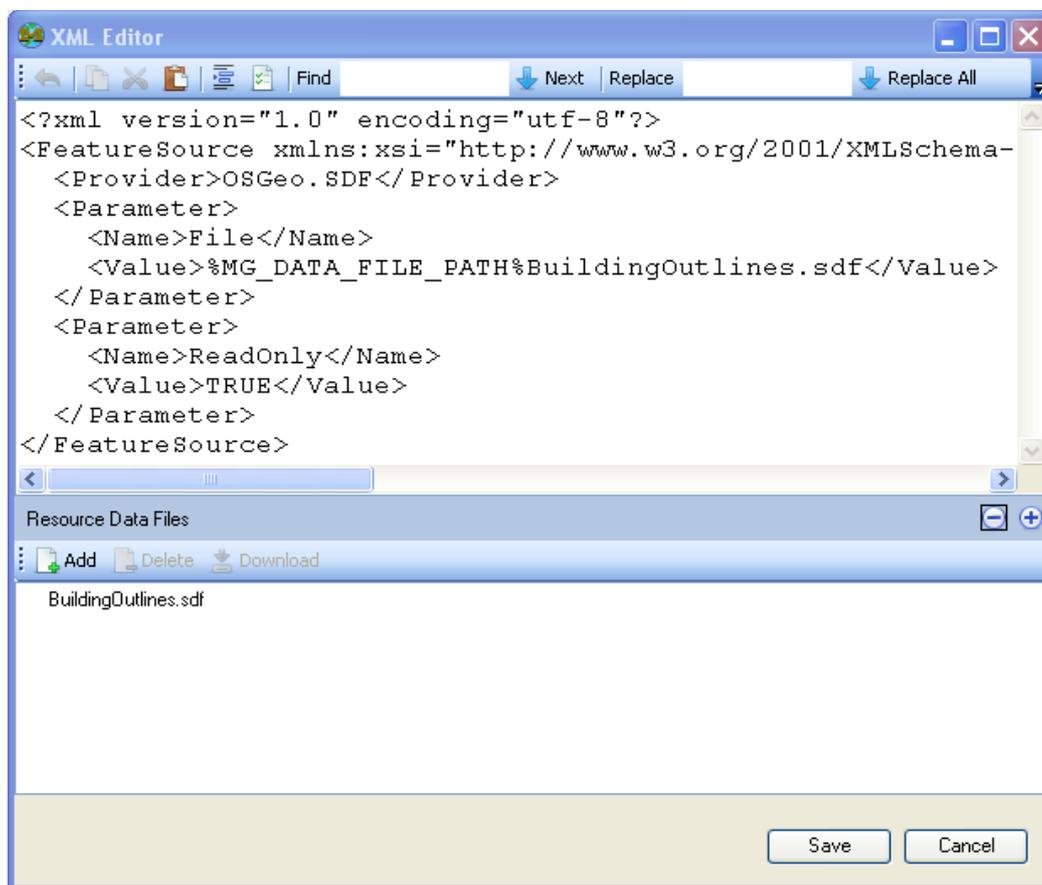
MapGuide Maestro es una herramienta de autor que facilita la gestión de datos espaciales MapGuide Open Source y está especialmente dirigido a la edición de componentes como las fuentes de datos, las definiciones de capas, las definiciones de mapas, diseño web del visualizador, plantillas y widgets<sup>4</sup> que conformarán el geoportal.

En realidad Maestro no es más que un entorno gráfico amigable para editar archivos XML<sup>5</sup> de configuración de cada recurso, donde se le dice a MapGuide todas las características que va a tener el proyecto que vamos a visualizar a través del servidor cartográfico: señalaremos las fuentes de datos de donde vamos a obtener la información espacial (shapefiles, .tab, PostGIS, etc.), el número de capas que se visualizarán, cuales de ellas serán consultables, la simbología con que se

4 Un widget es una pequeña aplicación ya preprogramada que permite acceder a funciones que son utilizadas con frecuencia. En nuestro caso estas pueden ser medir distancias, crear áreas de influencia, seleccionar y deseleccionar elementos, etc.

5 Estos archivos son almacenados en una base de datos Berkeley XML DB.

representarán, los niveles de zoom, etc. En definitiva, podríamos editar este archivo XML directamente desde, por ejemplo, el bloc de notas de Windows, pero resulta mucho más fácil y rápido hacerlo con Maestro.



Aunque la gran mayoría de la configuración de un proyecto de MapGuide se puede realizar con Maestro, en determinados casos muy concretos, como añadir símbolos personalizados al mapa, aún es necesario hacerlo "manualmente" editando el código XML.

MapGuide Maestro es la alternativa de código libre a Autodesk MapGuide Studio, a semejanza de MapGuide Open Source y Autodesk MapGuide Enterprise que ya comentamos con anterioridad. Existen versiones de Maestro para Windows, Linux y Mac OS.

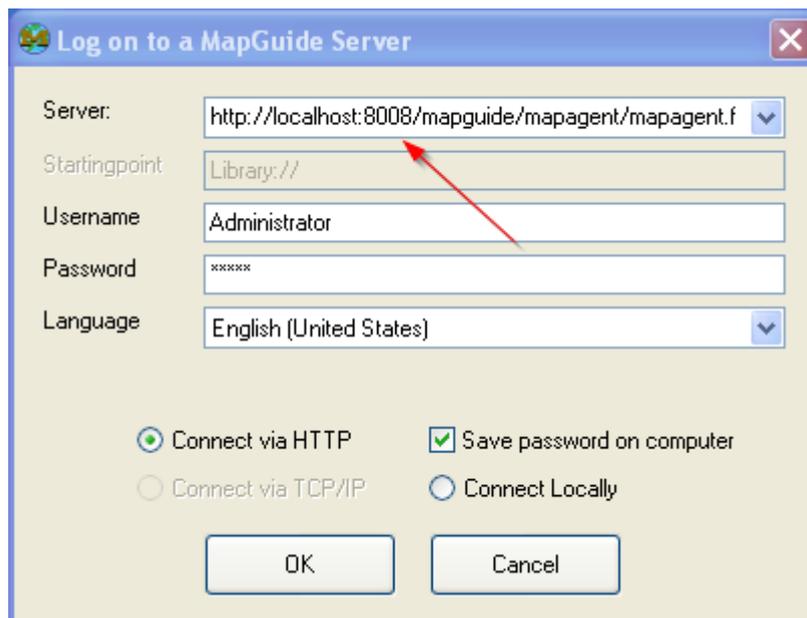


Descarga MapGuide Maestro desde el siguiente enlace:  
<http://trac.osgeo.org/MapGuide/wiki/maestro/Downloads>

Maestro requiere tener instalado previamente en el ordenador por lo menos Microsoft .NET Framework 2.0 Service Pack 1. Si no es tu caso deberás descargar el ejecutable desde la web de Microsoft e instalarlo.

## 4.1 El entorno de trabajo

Una vez hemos arrancado MapGuide Maestro vemos que al iniciar la primera vez el programa nos aparece la siguiente ventana:



En ella deberemos introducir la URL del servidor. Como hemos hecho la instalación por defecto de MapGuide bajo Apache la dirección será la siguiente:

<http://localhost:8008/mapguide>



Recuerda poner en la URL el puerto que hemos definido para servidor web de MapGuide en el asistente de instalación. En nuestro caso hemos utilizado el **puerto 8008** que viene por defecto.

En el caso en que hayamos instalado Maestro en un equipo diferente a donde esté corriendo el servidor MapGuide entonces sustuiremos *localhost* por el nombre o la IP de ese servidor (por ejemplo <http://192.168.0.11:8080/mapguide>).

A continuación pondremos en las siguientes cajas de texto el nombre de usuario que creamos al instalar MapGuide (por defecto **Administrator**) y la contraseña (por defecto **admin**).

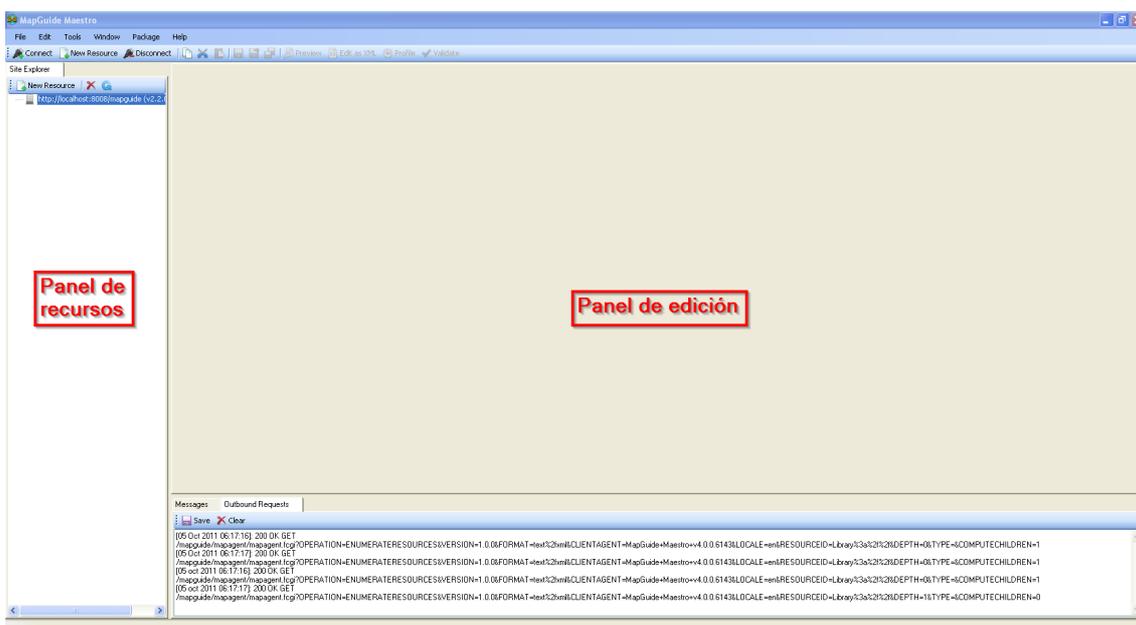
Dejamos seleccionada la conexión vía HTTP y marcamos si queremos que nos grabe la contraseña en el equipo para que nos la recuerde la próxima vez que entremos.

A continuación pulsaremos OK con lo que conectaremos Maestro con el servidor de mapas MapGuide y entraremos en el programa.



Si Maestro lanza un error señalando que no se puede conectar con el servidor cartográfico revisa que el servicio *ApacheMapGuide* está iniciado en el Panel de Control de Windows → Herramientas administrativas → Servicios. Si no es así arráncalo y vuélvelo a intentar.

En la parte superior, debajo la barra de menús, veremos una barra de herramientas con la mayoría de los iconos inicialmente desactivados. A la izquierda se situarán, a modo de árbol, un listado del que colgará todos los elementos -llamados *recursos*- de que consta el proyecto y en el centro todo un espacio, ahora vacío, destinado a la edición de estos recursos.

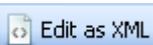
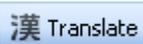


En la parte inferior se pueden visualizar la consola de IronPython<sup>6</sup>, el panel de solicitudes salientes y un panel con el registro de mensajes.

Los iconos son los siguientes:

 Connect	Conectar al servidor de mapas
---	-------------------------------

<sup>6</sup> IronPython es una implementación del lenguaje de programación Python para la plataforma .Net

 New Resource	Crear un nuevo recurso.
 Disconnect	Desconectar del servidor de mapas.
	Copiar un recurso o carpeta seleccionada.
	Cortar un recurso o carpeta seleccionada.
	Pegar un recurso o carpeta desde el portapapeles.
	Guardar el recurso actual.
	Guardar el recurso actual con otro nombre.
	Guardar todos los recursos abiertos.
 Preview	Vista preliminar del recurso.
 Edit as XML	Editar el recurso actual con un editor XML.
 Profile	Analizar el tiempo de renderizado del mapa.
 Validate	Validar el recurso actual contra errores.
 漢 Translate	Traducir las herramientas disponibles en el visor web.
	Eliminar un recurso o carpeta seleccionada.
	Actualizar el árbol de recursos.

Como la mejor manera de aprender es viendo un modelo ya hecho vamos a cargar unos datos de ejemplo para poder movernos por la interfaz gráfica de Maestro. Para ellos nos bajaremos el paquete de datos de ejemplo “*Sheboygan.mgp*”.



Descarga los datos de ejemplo de la ciudad estadounidense de Sheboygan desde el sitio web de MapGuide:  
<http://MapGuide.osgeo.org/download/releases/2.0.x-samples>

En MapGuide un paquete es similar a una copia de seguridad de todos los datos de nuestro proyecto.



Un problema frecuente con los paquetes de MapGuide es que Internet Explorer renombra los archivos .mgp como .zip si se descargan de Internet con este navegador. En este caso simplemente hay que volver a cambiar la extensión renombrándolos.

Una vez tengamos descargado el archivo “Sheboygan.mgp” en Maestro nos vamos al menú **Package** → **Load package...** y seleccionamos el archivo.

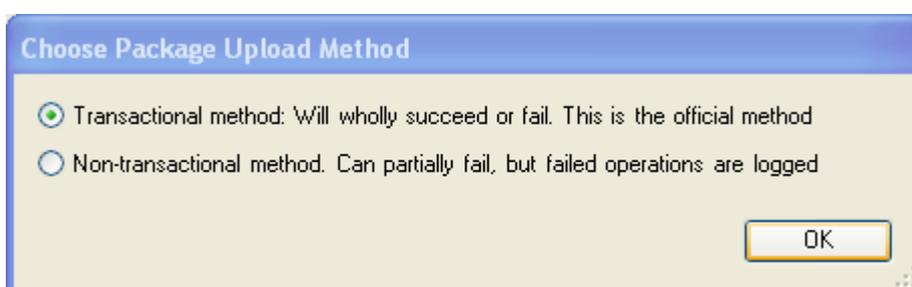
A continuación aparecerá una nueva ventana en la que se nos preguntará el método de carga del paquete: transaccional o no transaccional.

La diferencia radica en el caso de una carga **transaccional** en que esta se realiza como si fuese un conjunto de datos completo, por lo que al finalizar el proceso esta puede ser exitosa o no para el total. Este método suele ser más rápido.

Por contra, en el caso de una carga **no transaccional** se realiza mediante la subida individual de cada recurso almacenado en el paquete, permitiendo, a diferencia del método anterior, ver el progreso de toda las operaciones de carga de este. Si se produce un fallo en la subida este será parcial, restringido a los recursos que dan problemas, permitiendo ver en el archivo de registro cuales son.

En definitiva deberíamos utilizar el método no transaccional de carga cuando:

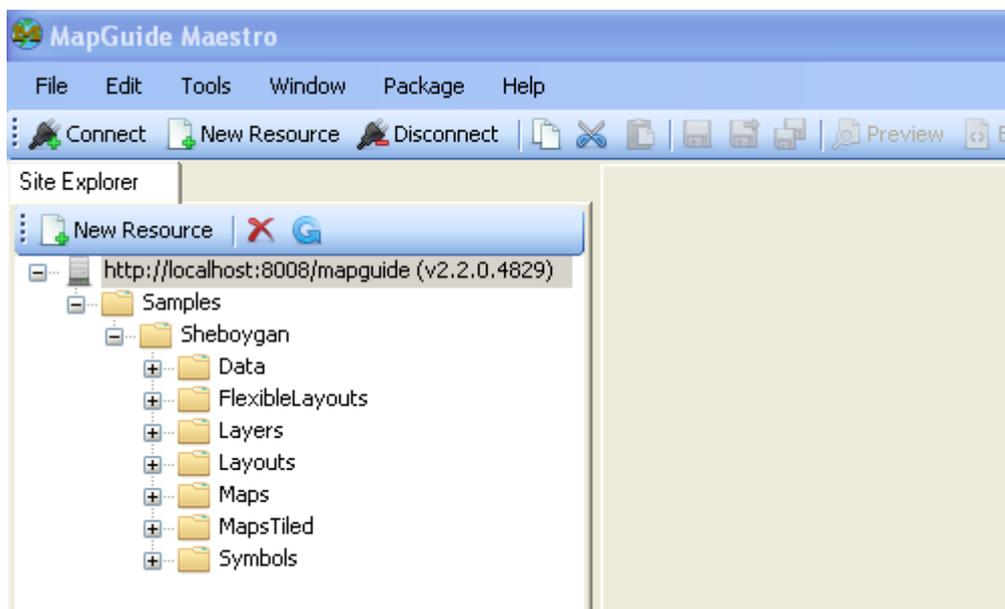
- El archivo del paquete es de varios cientos de megabytes de tamaño.
- El archivo de paquete contiene cientos de recursos para ser cargados.



Dado que el tamaño del paquete “Sheboygan.mgp” es de unos pocos megabytes, en nuestro caso utilizaremos el método transaccional que viene activado por defecto.

Tras cargar el paquete veremos como el panel de recursos se ha añadido una carpeta llamada *Sheboygan* de la cual cuelga toda un serie de recursos de que consta ese

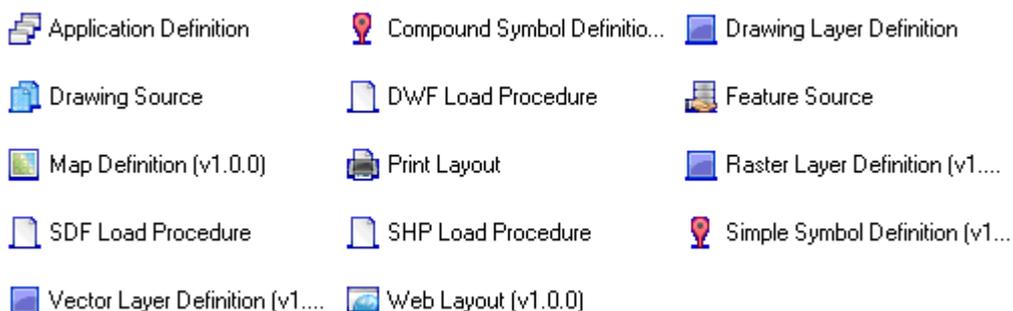
proyecto. Si vamos expandiendo el árbol veremos subcarpetas que recogen las fuentes de datos (*Data*), los *Flexible Layouts*, las capas (*Layers*), el diseño del visor cartográfico (*Layouts*), los mapas (*Maps*), mapas teselados (*MapsTiled*) y finalmente los símbolos (*Symbols*).



MapGuide Maestro permite crear de manera fácil y rápida la estructura de carpetas normalizada típicas de un proyecto MapGuide. Para ello seleccionamos con el botón derecho del ratón la carpeta raíz de la que queremos que cuelgue la estructura y seleccionando **Create Standard Folder Structure**.

Para ver mejor todos los tipos de recursos disponibles pulsaremos sobre el icono "New Resource" y se nos abrirá una nueva ventana con todos los disponibles.

#### Default



Aunque ver todos estos tipos de recursos va más allá de lo que es este tutorial, si profundizaremos a continuación en los cuatro tipos que contribuyen de manera directa o indirectamente a la creación de una aplicación de mapas web en MapGuide.

## 4.2 Las fuentes de datos

Una **Fuente de Datos** (*Feature Source*) es un tipo de recurso que proporciona a MapGuide la información necesaria para configurar un acceso a datos SIG.

MapGuide permite conectarse a un gran número de fuentes de datos a través de FDO.<sup>7</sup> FDO es un conjunto de bibliotecas de acceso, manipulación y análisis de datos independientemente del tipo de formato en que se almacena. De esta manera permite al servidor de mapas acceder a una gran cantidad de datos espaciales en Shapefile, PostGIS, MySQL, Oracle Spatial, MS SQL Server Spatial, ArcSDE, WMS, WFS, SDF, SQLite, etc. Así mismo, se puede acceder a multitud de formatos vectorial y raster a través de OGR y GDAL. FDO también puede conectarse a otras fuentes de datos geográficos no explícitos a través de ODBC.



La instalación por defecto de MapGuide no instala todos los proveedores a fuentes de datos existentes. Puedes agregarlos ejecutando de nuevo el instalador y seleccionando *Change*, sin necesidad de volver a instalar MapGuide de nuevo.

7 FDO es utilizado por una gran cantidad de productos, teniendo una licencia libre LGPL.

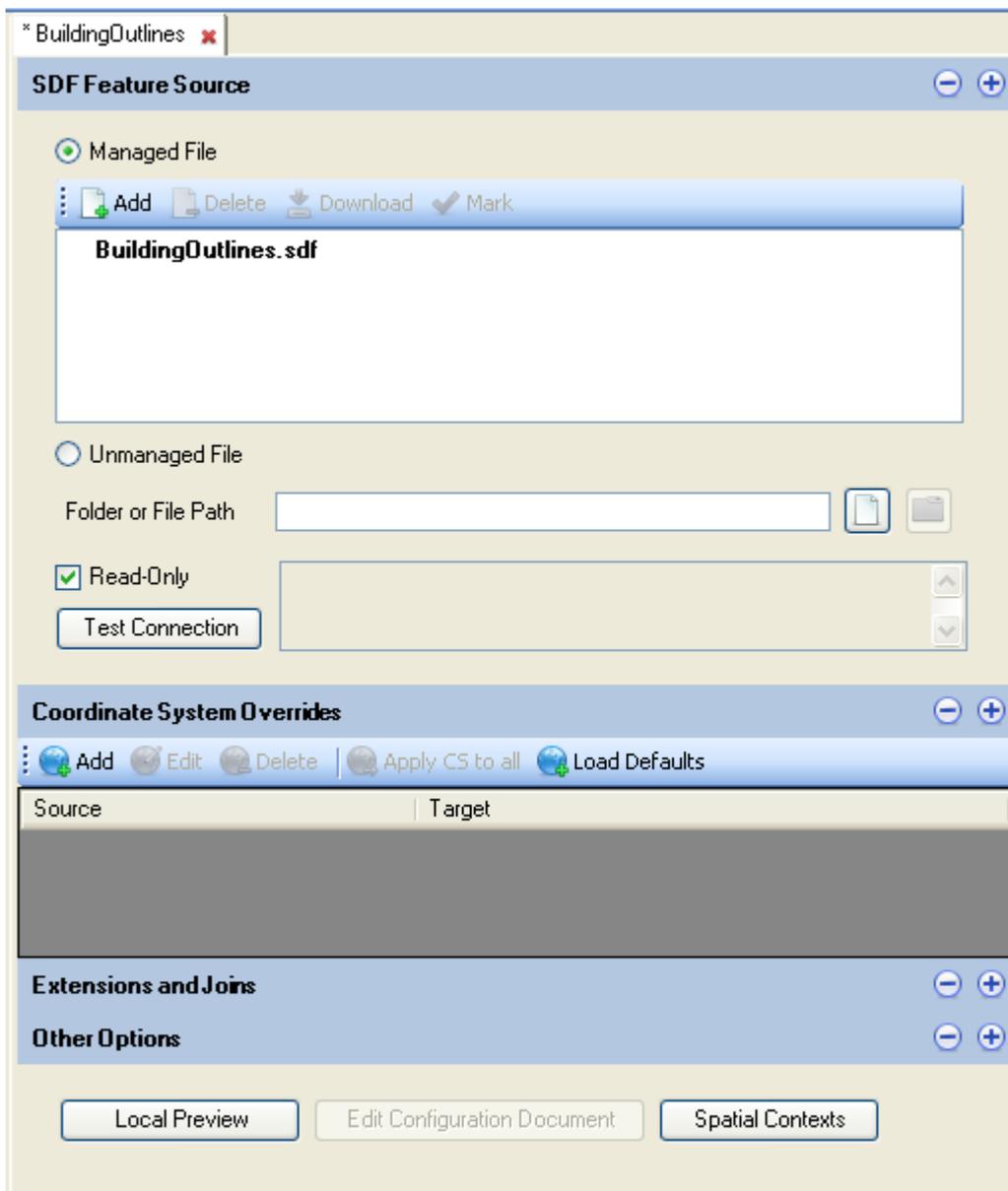


OGR admite de por sí un conjunto muy grande de fuentes de datos (incluso algunas no están basadas en archivo). Si seleccionas este tipo de proveedor MapGuide Maestro mostrará un editor personalizado para ese origen de datos concreto.

Si pulsamos sobre la carpeta *Data* presente en el árbol de elementos del proyecto Sheboygan veremos 12 fuentes de datos a las que se accede en el proyecto donde se almacena información geográfica relativa a usos del suelo, edificaciones, red hidrográfica, tipos de suelo, etc. Si hacemos doble clic sobre una de ellas, por ejemplo la primera llamada “BuildingOutlines”, veremos que se abre una pestaña de edición con cuatro secciones que se expanden y recogen por medio de los botones 

La primera sección señala el tipo de conector utilizado, en este caso para una archivo SDF.<sup>8</sup>

<sup>8</sup> SDF (Spatial Data File) es un formato de archivo de almacenamiento de datos espaciales de Autodesk que lo desarrolló inicialmente para su software Autodesk MapGuide.



Mediante sendos botones de opción podemos definir el tipo de conexión. En MapGuide se puede optar por utilizar los datos en su ubicación original o, por contra, copiar estos datos al servidor cartográfico. Si utilizas los datos en una carpeta fuera del servidor de mapas se denomina **Unmanaged file**, y si lo guardas en el interior del servidor de MapGuide se llama **Managed file**.

La razón por la que existen dos maneras de colocar los datos es que ambas tienen ventajas y desventajas.

Ventajas del **administrado** (*Managed file*) de archivos:

- Los datos no pueden ser modificados por otro programa.
- Están separados de los datos originales.

- Están contenidos en una copia de seguridad (paquete .mgp).

Ventajas del **no administrado** (*Unmanaged file*) de archivos:

- Puede ser modificado o leído por otro programa.
- Al no formar parte de una copia de seguridad los datos son los mismo que los originales.
- No están contenidos en una copia de seguridad (paquete .mgp), pero pueden ser respaldados por programas tradicionales que realizan *backups*.

Como te habrás dado cuenta las ventajas son exactamente las opuestas. Los recursos administrados se pueden transferir a otra persona simplemente poniéndolos en un paquete. Con los recursos no administrados no es necesario crear ese paquete, pero al enviar los datos por separado hay que asegurarse que el receptor copia los datos en los directorios adecuados. Cuál es la mejor manera dependerá del tipo de datos espaciales y el flujo de trabajo que tengamos.

En principio no debe existir una diferencia de rendimiento apreciable entre ambos tipos, aunque debido a que el uso de archivos no administrados permite más opciones de configuración, puede que si se realizan cambios indebidos afecten al rendimiento (por ejemplo, referenciar a un directorio de archivos SIG ubicado en otro equipo a través de una red).

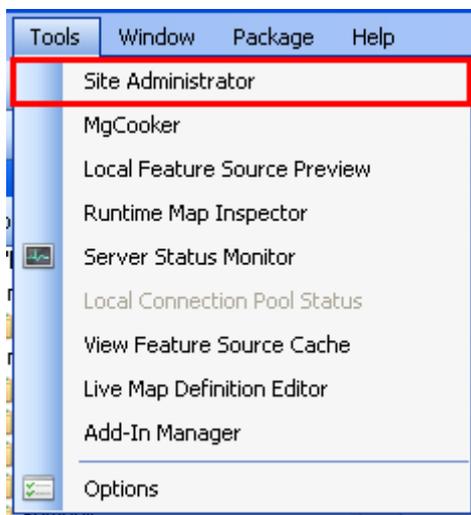
Si se opta por los datos administrados las opciones que permite MapGuide Maestro son las siguientes:

	Añadir un archivo al servidor.
	Eliminar un archivo del servidor.
	Descargar a nuestro equipo una copia del archivo desde servidor.
	Seleccionar el directorio para utilizarle como repositorio de datos (para varios archivos).

Si los datos se compone de varios archivos y se utiliza toda una carpeta, el proveedor OGR intentará averiguar el formato de los datos que existen en ella.

Si se prefieren datos no administrados basta con que se introduzca la ruta de acceso a los datos en **Folder or File Path**. Del mismo modo que el caso anterior se puede especificar una carpeta o un archivo. Si se especifica una carpeta la ruta no debe contener una barra o barra invertida al final (/). El botón  permitirá elegir el archivo o carpeta. Para utilizar esta última función se debe antes configurar una o más carpetas de alias a través del *MapGuide Site Administrator* (se puede acceder a las

páginas de administración del sitio MapGuide desde el menú **Tools** → **Site Administrator** de *MapGuide Maestro*).<sup>9</sup>



Ten en cuenta que la ruta a los archivos es la del equipo donde está instalado MapGuide, no la del ordenador donde está instalado MapGuide Maestro, ya que puede que ambos nos estén ejecutándose en el mismo equipo.

Podremos señalar también si la fuente de datos estará protegida contra escritura marcando **Read-Only**.

A continuación para verificar si es correcta la conexión a los datos pulsaremos sobre el botón **Test Connection**. Si es así aparecerá la palabra *True*.

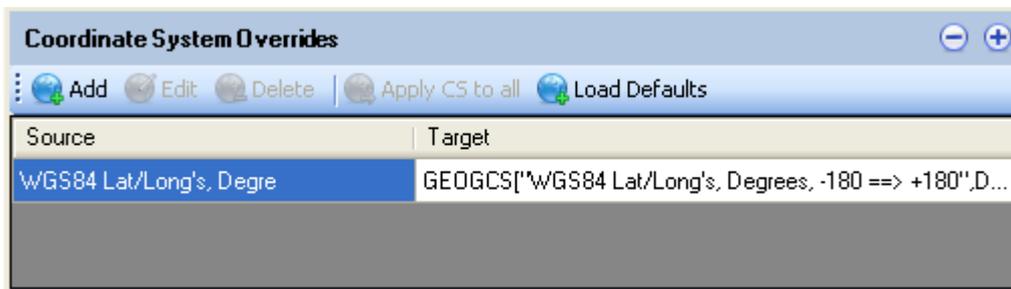


En el apartado **Coordinate System Overrides** Maestro nos da la posibilidad de definir un sistema de coordenadas para los datos independientemente si ya le tienen asignado de origen o no. Esto es conveniente si una fuente cartográfica tiene definido un sistema incorrecto o simplemente no está definido.<sup>10</sup> La opción permitirá asignarle un sistema de coordenadas sin tener que modificar los datos fuentes. Lógicamente el

<sup>9</sup> Las opciones de configuración del sitio MapGuide a través de las páginas de administración las veremos más adelante.

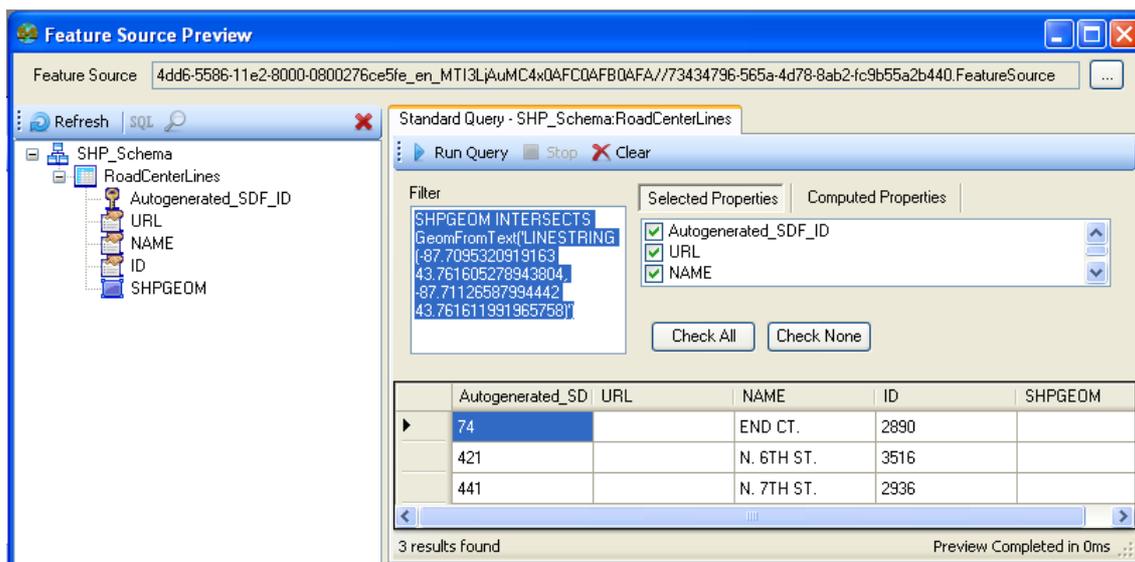
<sup>10</sup> MapGuide solo puede trabajar con datos espaciales con sistemas de coordenadas definidos.

sistema de coordenadas seleccionado en MapGuide Maestro prevalecerá sobre el intrínseco de los datos.



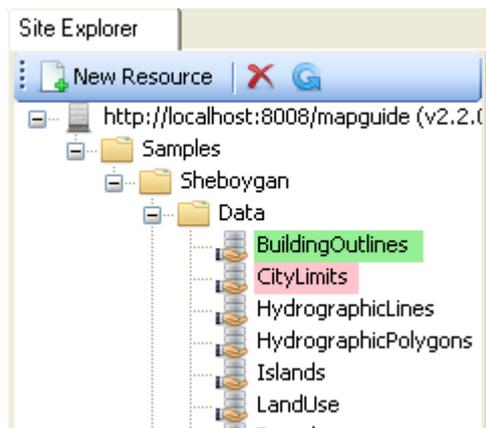
Para un mejor rendimiento del servidor se suele utilizar el mismo sistema de coordenadas tanto para los datos como para el mapa que mostraremos.

Por último señalar que en el apartado **Other Options**, pulsando sobre el botón **Local Preview** podemos acceder a un editor de expresiones donde realizar consultas SQL espaciales o no a la fuente de datos.



Cerraremos las pestañas de edición saliendo sin guardar pulsando el icono con una "X" roja.

Si hemos realizado alguna edición veremos que en el árbol de elementos de nuestro proyecto la fuente de datos se resaltarán en rojo y una ventana nos preguntará si queremos guardar los cambios. Pulsamos "No".

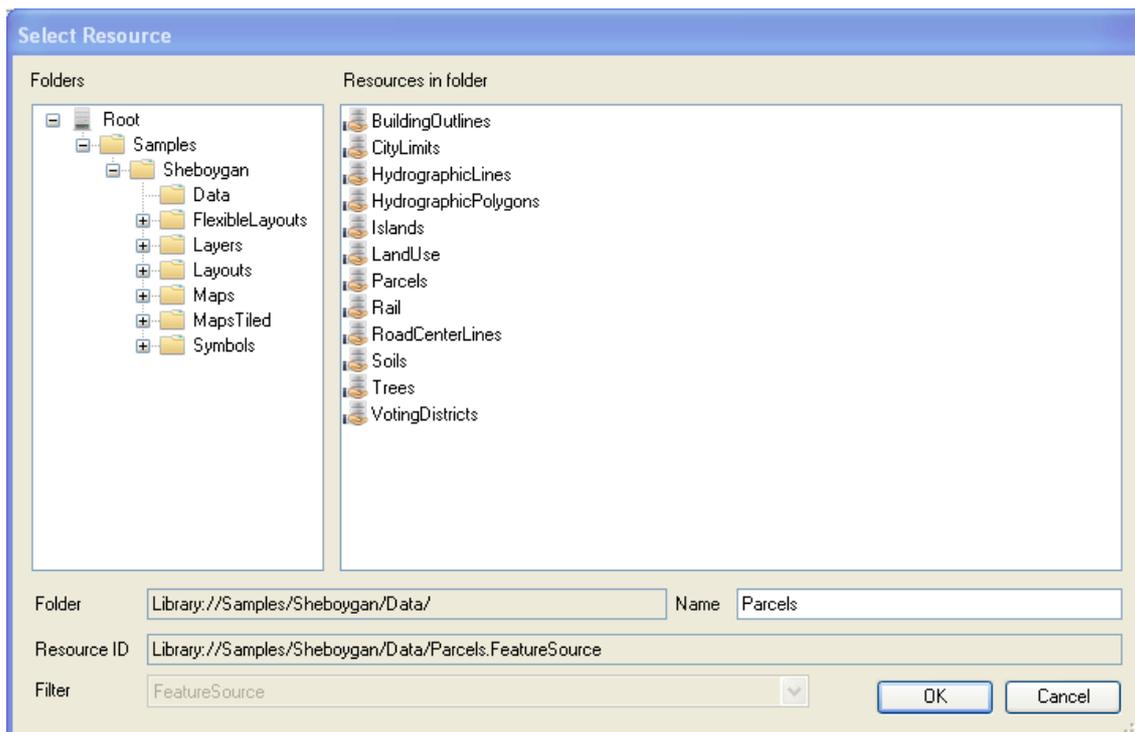


### 4.3 Las capas de datos

Una **Definición de Capa** (*Layer Definition*) es un recurso que proporciona a MapGuide las características que tendrán la representación cartográfica de una fuente de datos.

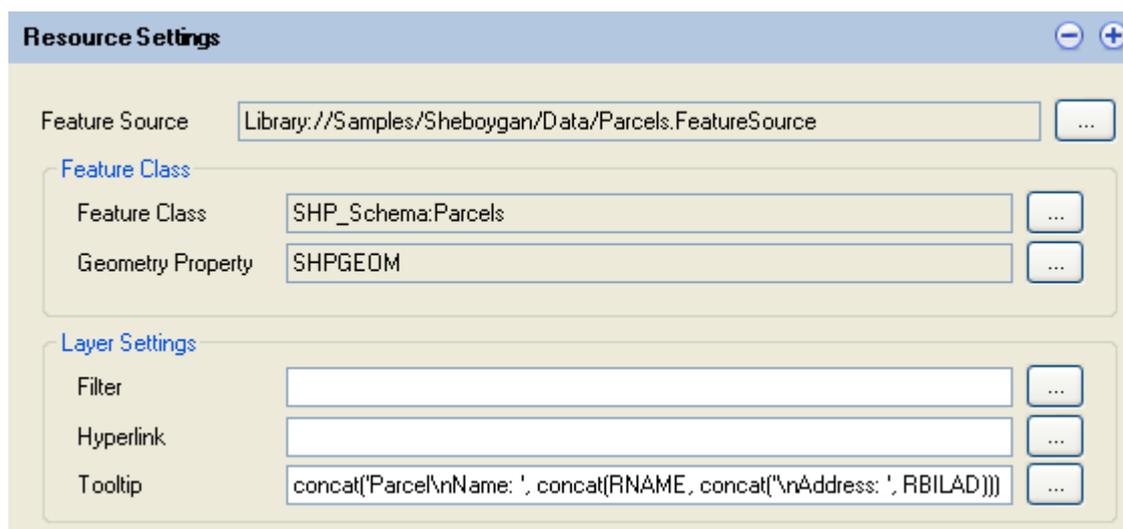
Ahora en el panel de vista de árbol de la izquierda expandimos la carpeta *Layers* y hacemos doble clic sobre la capa "Parcels". Se abrirá de nuevo una pestaña de edición. En el apartado "**Resource Settings**" (*Configuraciones del recurso*) configuraremos las propiedades básicas de la capa.

En "**Feature Source**" pulsamos sobre el botón con tres puntos suspensivos  y se abrirá una ventana con todos los recursos existentes. Seleccionamos la carpeta "Data" en el árbol de recursos y en el panel de la derecha veremos todas las fuentes de datos de que disponemos. En nuestro proyecto de ejemplo llamado Sheboygan se ha seleccionado la fuente de datos *Parcels*, la cual almacena las parcelas catastrales de la ciudad. Pulsamos "Cancel" para cerrar la ventana sin realizar cambios.



El “**Feature Class**” muestra la lista de tablas o archivos disponibles en la fuente de datos. Los datos basados en archivos, como por ejemplo los shapefiles, tendrán aquí una única opción.

En “**Geometry Property**” se define el campo donde se almacenan los datos geométricos. Como en el caso anterior para la mayoría de las fuentes de datos solo dispondremos de una única opción por lo que no será necesario modificarlo.



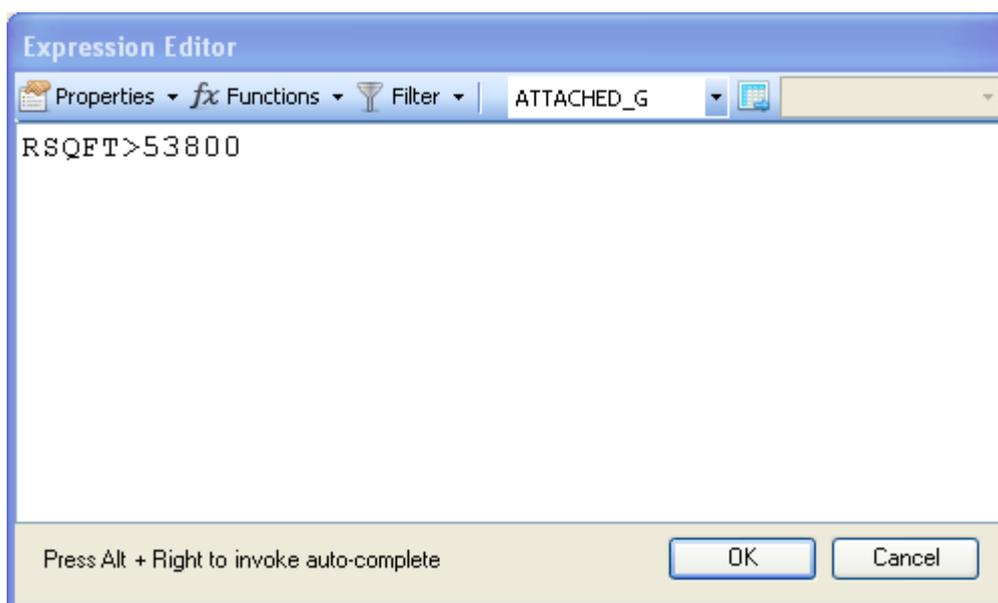
El apartado “**Layer settings**” (*Configuración de la capa*) podremos ajustar algunos comportamientos de la capa.

Si solo deseamos mostrar determinados registros de la capa podemos realizar una selección por filtro en la opción “**Filter**”, por ejemplo visualizar únicamente los lotes de parcelas que sea mayores de 5000 m<sup>2</sup> (53800 Sq ft). Un filtro es una expresión SQL<sup>11</sup> en formato FDO, por lo que para utilizar este campo debes tener algún conocimiento de este lenguaje de consulta.



El aprendizaje del lenguaje SQL cae fuera del ámbito de esta materia, pero en Internet encontrarás buenos tutoriales de introducción a SQL que te podrán ser de ayuda. Para un nivel básico solo necesitarás aprender la sintaxis fundamental de SQL para poder realizar selecciones de registros de bases de datos.

Si pulsamos el botón  de la derecha de la caja de texto se abrirá una editor de expresiones que nos ayudará a escribir las condiciones expresadas tras la clausula WHERE de la declaración SQL. Siguiendo el ejemplo anterior la condición sería la siguiente:



En nuestro caso RSQFT es el campo donde se almacena las superficies de las parcelas en pies cuadrados. Con esto se ocultará del mapa todas las parcelas inferiores a 53800 Sq ft.

Hacemos clic sobre el botón “**Cancel**” para cerrar el editor de expresiones sin guardar.

La opción “**Hyperlink**” permite poner un enlace a cada entidad. Así, si un usuario detiene el ratón sobre el elemento del mapa recibirá un mensaje indicando que puede

<sup>11</sup> SQL (Structured Query Language) es el lenguaje estandarizado más popular utilizado para realizar consultas a bases de datos.

abrir un hipervínculo. Para ello se debe introducir en la caja de texto la dirección URL entre comillas simples.

Hyperlink `'http://mapguide.osgeo.org'`

En este caso todos los elementos del mapa tendrán un enlace que apuntarán a un mismo sitio. En la mayoría de los casos quedremos utilizar el valor de una columna como parámetro del enlace, creando vínculos más complejos. Para concatenar los valores utilizaremos el operador +.

Hyperlink `'http://www.catastro.es?parcela=' + "id"`

En este caso a la URL le pasamos el parámetro “id” que corresponderá con el campo ID de nuestra capa. Observa que en este caso el nombre del campo va entre comillas dobles.



Una forma más robusta de hacer lo mismo, porque da menos problemas, pero menos evidente, es usando la función SQL “CONCAT”. En nuestro caso el ejemplo anterior quedaría así:

```
CONCAT('http://www.catastro.es/?parcela=', "id")
```

Al igual que en casos anteriores si pulsamos el botón de la derecha de la caja de textos se abrirá una editor de expresiones que nos facilitará introducir nombres de columnas y valores.

Finalmente la opción “**Tooltip**” permite introducir un mensaje emergente, a modo de descripción, que el usuario del visor cartográfico podrá ver si mantiene unos segundos el puntero del ratón sobre el elemento del mapa. Del mismo modo que en la opción “**Hyperlink**”, el campo está sujeto a la mismas reglas que este. En nuestro ejemplo de *Sheboygan* se ha concatenado diferentes campos de la capa.

Tooltip `concat('Parcel\nName: ', concat(RNAME, concat('\nAddress: ', RBILAD)))`

Los caracteres \n representan un retorno de carro. El resultado en el mapa será una descripción emergente como esta:





En los *tooltips* se permite introducir código HTML como el ejemplo siguiente, que mostraría una imagen JPG cuyo nombre coincide con el identificador (campo ID) de cada registro:

```
''
```

En el siguiente gran apartado llamado “**Layer Properties Visible in Viewer**” (*Propiedades de la capa visible en el visor*) es posible seleccionar aquellos campos de información asociados al elemento geográfico de la capa que deseamos se muestren en el visor cartográfico. Para ello debemos seleccionarlos marcando la columna “**Visible**”. Además, podemos cambiar el título del campo por otro más explícito en la columna “**Display Name**”, así como el orden de presentación de los campos seleccionando uno de ellos y pulsando los iconos con una flecha hacia arriba o hacia abajo:

Layer Properties Visible in Viewer		
<input checked="" type="checkbox"/> Check All <input type="checkbox"/> Uncheck All <input type="button" value="Invert"/> <input checked="" type="button" value="↑"/> <input checked="" type="button" value="↓"/>		
Visible	Name	Display Name
<input checked="" type="checkbox"/>	RTYPE	Zone
<input type="checkbox"/>	DETACHED_G	DETACHED_G
<input checked="" type="checkbox"/>	RACRE	Ácreage
<input checked="" type="checkbox"/>	RLOT	Lot Dimensions
<input checked="" type="checkbox"/>	RNAME	Owner
<input checked="" type="checkbox"/>	RLDESCR2	Description2
<input type="checkbox"/>	NO_UNITS	NO_UNITS

Finalmente tenemos el último gran apartado llamado “**Layer styles**”. Es en esta sección donde definiremos realmente los estilos de la capa, cómo se deberán mostrar los datos en el visor cartográfico desde el punto de vista semiológico.

Debemos tener en cuenta que un mapa puede mostrar una gran cantidad de información al mismo tiempo. Como en cualquier Sistema de Información Geográfica, para comunicar esta información hay que elaborar previamente un esquema visual adecuado que nos permita transmitir sin ambigüedades los atributos del territorio que queremos mostrar así como sus relaciones espaciales. Para ellos es necesario que definamos un esquema que organice la información desde la más vaga a la más compleja según el grado de zoom.

Un mapa típico empezará con información general de la zona que queremos mostrar para luego, en función de la escala de visualización que decida el usuario, volverla más y más detallada. Si uno trata de mostrar, por ejemplo, una visión general de Cantabria con todos los edificios existentes en la región lo más probable es que consiga un mapa poco útil y manejable. Por esta razón cada capa debe tener unos

rangos de escala en la que visualizarse. Con esto evitaremos que el mapa pueda parecer una compleja maraña de símbolos desordenados de manera caótica.



«No existe la sobrecarga de información, únicamente un mal diseño.»

*Edward Tufte*

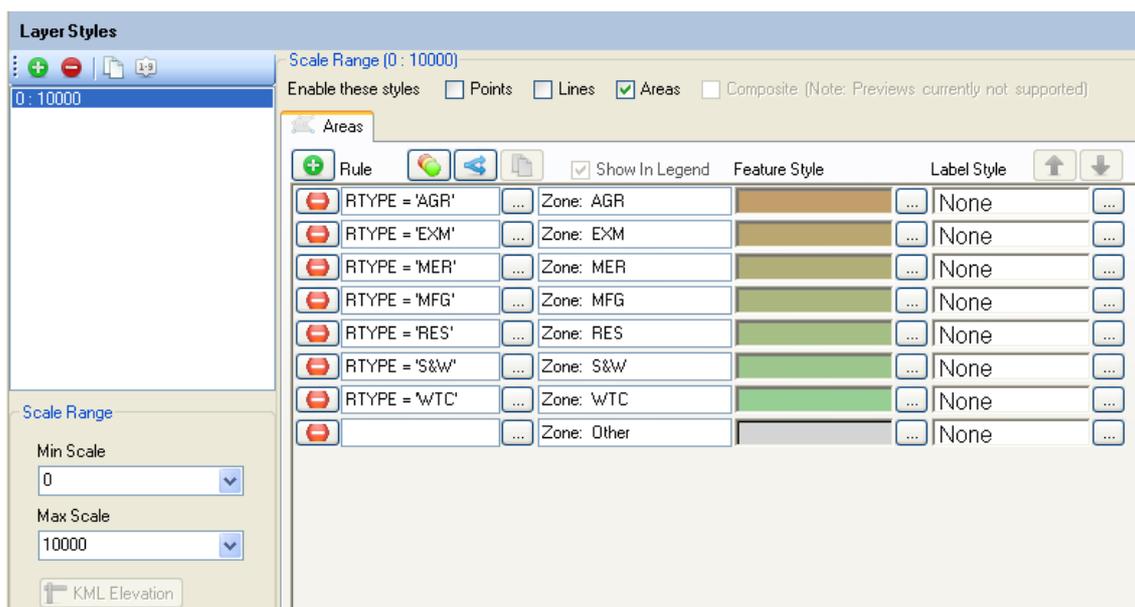
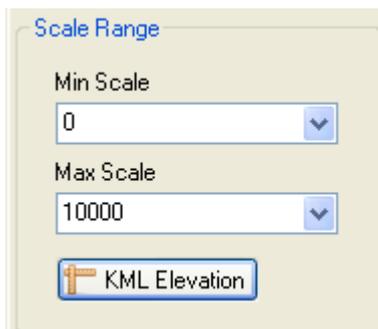
Un rango de escala define lo que la capa mostrará en un determinado nivel de zoom. Un uso común de los rangos de escala es el de ocultar elementos que por su pequeño tamaño no se activarán a menos que el usuario del visor no haya ampliado suficientemente el zoom del mapa. Otro uso sería, por ejemplo, el de ajustar el grosor de las líneas de una capa con el fin de que sean visualmente más o menos intrusivas a determinados niveles de escala.

En este apartado de MapGuide Maestro vemos en la parte superior cinco iconos. Los tres primeros botones se utilizan respectivamente para:

	Añadir un nuevo rango de escalas vacío.
	Eliminar un rango de escalas ya existente.
	Hacer una copia de un rango de escala y presente.
	Ordenar los rangos de escalas según su valor.

Si pulsamos en el icono para añadir un nuevo rango de escalas aparecerá en el panel inferior un grado de escala genérico *0:Infinity*. En el apartado “**Scale Range**” seleccionamos la horquilla de valores superior e inferior que definirán la escala mínima y máxima en la que queremos que se active el estilo que vamos a definir.<sup>12</sup>

<sup>12</sup> Si nos damos cuenta en estos dos cuadros de diálogos el concepto de escala es a la inversa del real, en el cual una escala es mayor cuanto más grande está ampliado el mapa y menor cuanto más nos alejamos de la superficie. Aquí se utiliza la idea extendida de relacionar el grado de una escala con el denominador (1000) y no con el cociente de la división (1/1000).



En el rango ya existente el valor mínimo de escala está señalado como ilimitado (*Infinity*) y el máximo está fijado en 10.000. Esto significaría que la capa "Parcels" en la que estamos ahora sólo se verá en niveles de zoom iguales o inferiores a 1/10.000. Si la capa debe ser visible siempre, a cualquier nivel de zoom, pondríamos Min Scale=0 y Max Scale=Infinity.

Si nos fijamos, en el apartado "Scale Range" existe un botón denominado "**KML Elevation**". Permite especificar los ajustes de elevación/extrusión de la capa cuando señalamos que el formato de salida de esta sea en KML para su visualización mediante Google Earth.

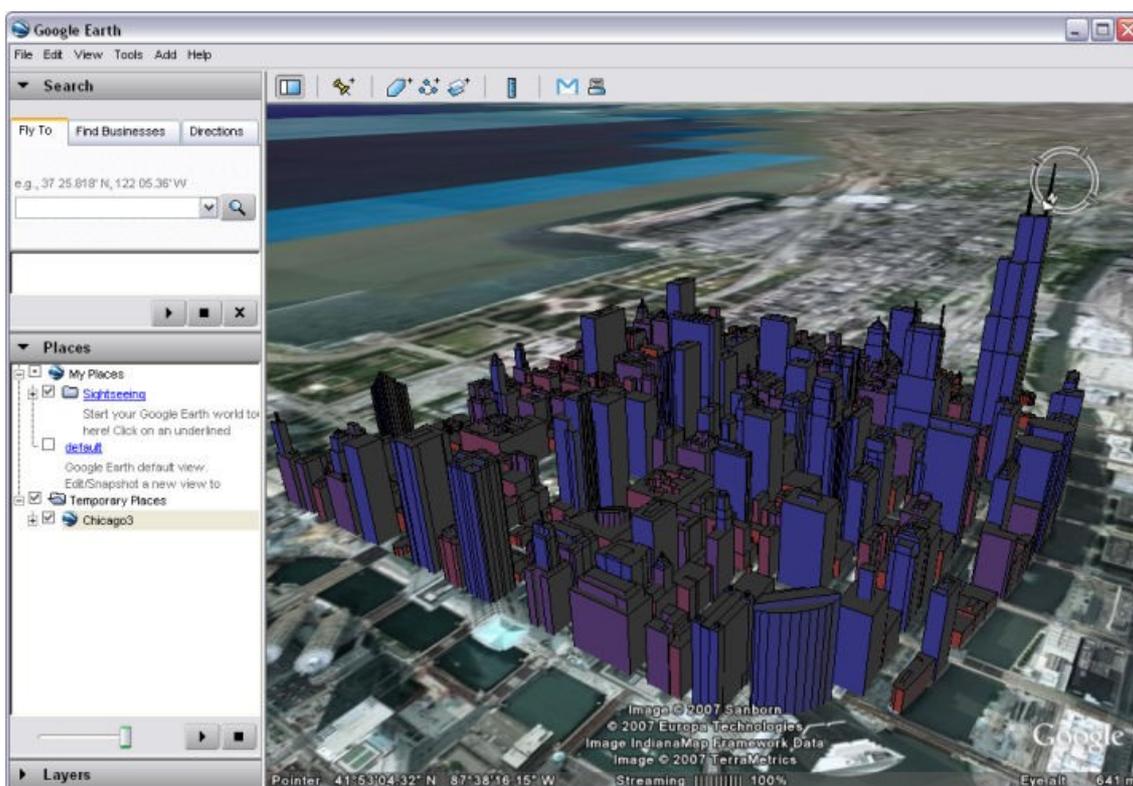


Esta característica solo aparecerá si el recurso de nuestro proyecto está actualizado a la última versión del esquema. Si no es así MapGuide Maestro nos ofrece la posibilidad actualizarlo desde el botón "**Upgrade**"

This resource can be upgraded to a newer version to take advantage of newer feature

Upgrade

MapGuide es compatible actualmente con representaciones en 2D de los datos espaciales. Sin embargo, muchos conjuntos de estos datos pueden contener información sobre su altura, dando la posibilidad de una representación 2.5D.<sup>13</sup> El último esquema<sup>14</sup> de definición de capas de MapGuide permite introducir estos valores mostrando objetos sólidos en 2.5D en Google Earth (por el momento solo en este programa, con idea de extenderlo en el futuro a otro tipo de visores).



Fuente: <http://trac.osgeo.org/mapguide/wiki/MapGuideRfc16>

Para ello debemos definir:

- **Z-Offset.** Valor de desplazamiento de altura de arranque de la capa. Esto es útil cuando existe un “hundimiento” apreciable de los elementos de la capa o, al

13 Actualmente la mayoría de los Sistemas de Información Geográfica que existen en el mercado gestionan y analizan la información espacial en dos dimensiones. Dentro de estos existen *softwares* híbridos a medio camino entre el 2D y el 3D, denominados de 2.5D o falso 3D, utilizados fundamentalmente para la visualización de los datos. El desarrollo de SIG completamente 3D, con capacidades de gestionar y analizar este tipo de geometrías, es uno de los grandes retos a los que se enfrenta la evolución de este tipo de sistemas.

14 Un esquema es una colección de características para una fuente de datos. Los esquemas evolucionan a medida que lo hace MapGuide, añadiendo nuevas características a los datos visualizados en el servidor de mapas.

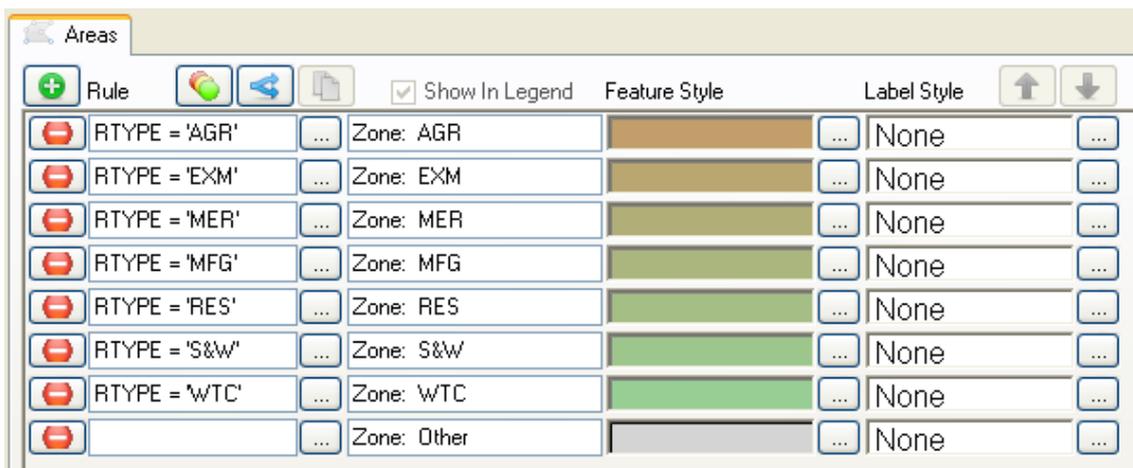
contrario, estos se quedan “en el aire”. Viene motivado porque la base de los elementos está en cota negativa o positiva (esto es muy por encima o por debajo) respecto al MDT de Google Earth. Con este valor lo corregimos variando la altura de arranque si es el caso.



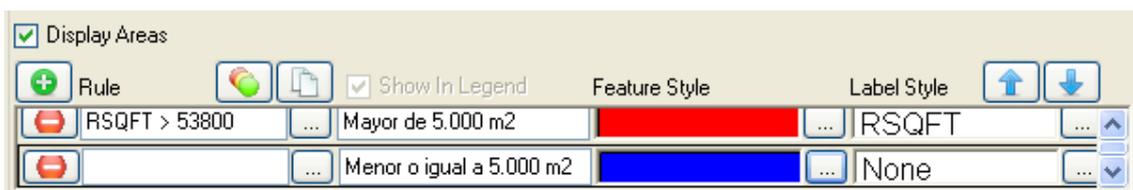
- **Z-Extrusion.** Valor de la extrusión vertical de los elementos de la capa.
- **Z-Offset Type.** Si el valor Z-Offset se va a contabilizar desde la superficie del MDT (*RelativetoGround*) o no (*Absolute*).
- **Units.** Selección de unidades de medición métricas o imperiales en que está definida la altura.

Dentro de cada rango de la escala es posible seleccionar la información que aparecerá en cada uno de los tres tipos básicos de geometría: puntos, líneas y polígonos. La casilla de verificación nos permite ocultar la pestaña un tipo de geometría específica, lo cual es útil si la fuente de datos de la capa admite diferentes tipos de geometrías. Si ninguna de las casillas están marcada ninguna de las geometrías será visible en el rango de escala propuesto.

Para cada geometría deberemos crear una regla o filtro. A cada regla se le puede asignar una expresión que filtrará aquellos elementos de la capa a los que se les asignará un estilo concreto. Al igual que en los casos anteriores que hemos visto, las expresiones de las reglas se basan en una sintaxis SQL. En la mayoría de los casos utilizaremos expresiones sencillas. Si observamos el ejemplo de *Sheboygan* que estamos siguiendo vemos que se ha filtrado cada tipo de parcela según su uso de suelo recogido en el campo “RTYPE”.



Ahora imaginemos que queremos mostrar con un estilo únicamente aquellos lotes de parcelas que tengan una superficie mayor de 53.800 pies cuadrados y el resto con otro estilo. Crearíamos un par de reglas pulsando el botón + y las configuraríamos como se muestra a continuación.



Observa que he añadido una etiqueta para cada regla con una breve descripción del intervalo que se mostrará en el gestor de capas del visor a modo de leyenda, para ello debe activarse la casilla **"Show In Legend"** (*Mostrar en la leyenda*).



Esta característica solo aparecerá si el recurso de nuestro proyecto está actualizado a la última versión del esquema. Si no es así MapGuide Maestro nos ofrece la posibilidad de actualizarlo desde el botón **"Upgrade"**

This resource can be upgraded to a newer version to take advantage of newer features

Upgrade

Así mismo, se ha configurado el primer estilo, el de color rojo, para que en los lotes de los polígonos se muestre en su centroide la superficie (dato este recogido en el campo RSQFT). Fíjate que la segunda regla no tiene ninguna expresión.<sup>15</sup> Todos los objetos que tengan en el campo RSQFT un valor superior a 53800 se verán con el estilo que dicta la primera regla, y todos los demás se mostrarán como señala la segunda regla.

<sup>15</sup> Podríamos haber escrito RSQFT <= 53800 y el resultado sería el mismo.

Si pulsamos en la barra superior de herramientas sobre el icono de previsualización

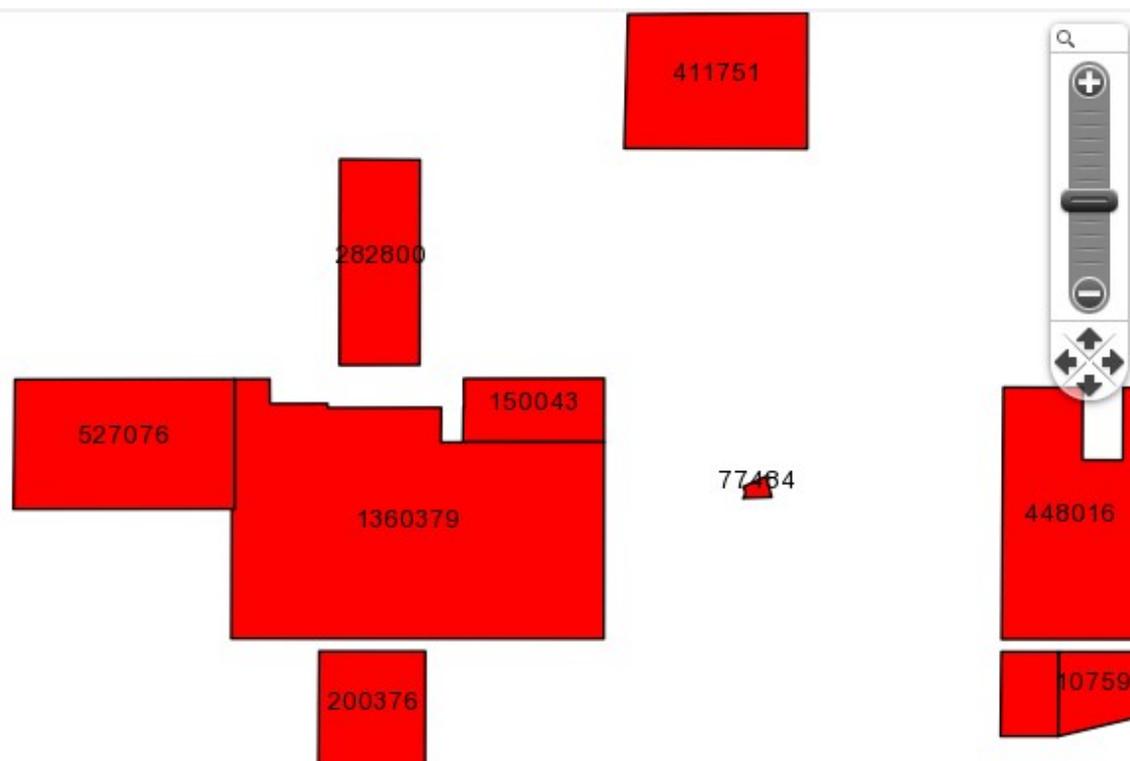


el resultado en el navegador será el siguiente:



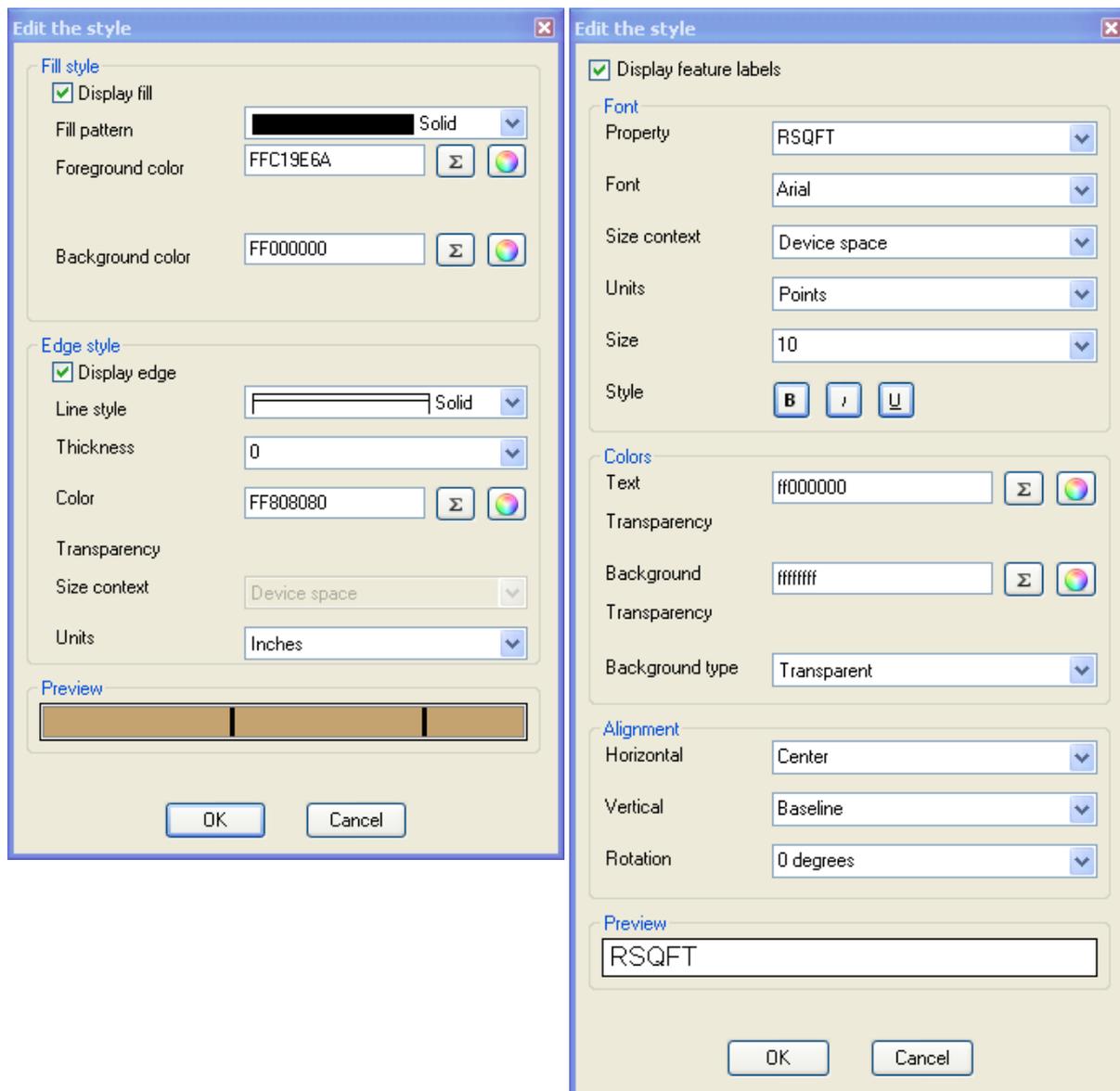
Podemos ver como los polígonos de lotes de parcela mayores de 53800 Sq ft son mostrados de color rojo con su superficie en su interior, mientras que los inferiores se representan de color azul sin etiquetar.

Si eliminamos la segunda regla haciendo clic sobre el botón  que hay a su izquierda solo se mostrarán los elementos filtrados por la primera regla. Si un atributo de los elementos coincide con varias reglas se aplicará la primera ellas.



Si pulsamos el botón  en la columna relativa al estilo de los elementos se abrirá una ventana para editar los estilos de color de relleno y borde de cada regla. Si por contra pulsamos sobre el mismo botón pero en la columna que se refiere al estilo de etiqueta podremos definir el campo por el que se etiquetará, el tipo de fuente y tamaño, su color y su alineamiento. En este último caso el programa nos permite introducir una expresión para colocar la etiqueta en función de un campo (que, por ejemplo, recoja el ángulo de rotación).





Si pulsamos el botón  podemos hacer que Maestro nos cree las reglas automáticamente en función del número de intervalos que le señalemos y su tipo, si el campo es numérico: iguales, desviación estándar, cuantil o de valores únicos. Estos intervalos podrán sobrescribir las reglas ya existentes al agregarse a ellas.

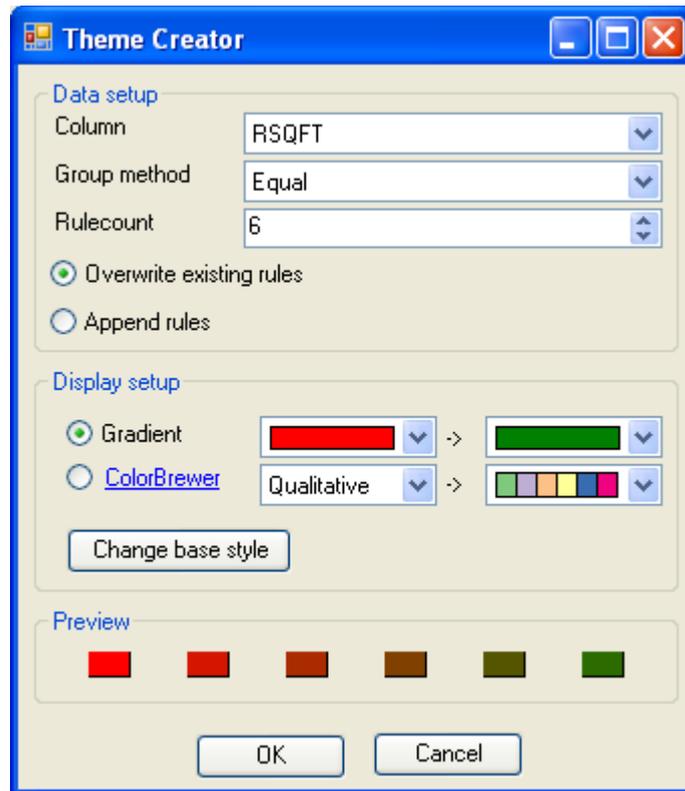
Así mismo, permite definir las rampas de colores que vamos a utilizar, bien indicando un gradiente de color basándonos en los colores extremos que señalemos, o bien utilizando rampas ya predefinidas dependiendo de si los datos son cualitativos, secuenciales o divergentes.



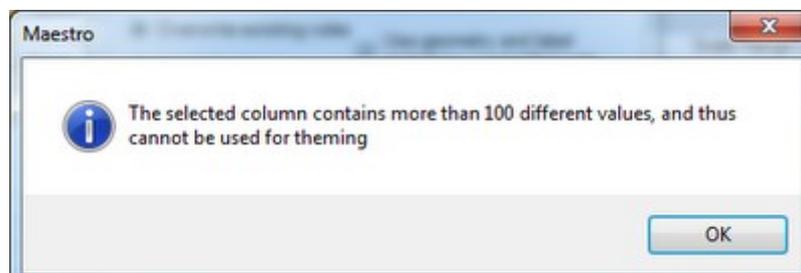
Para crear tablas de colores personalizadas puedes utilizar *ColorBrewer*, una web que te permite realizar esquemas de

colores destinados al uso en mapas. La herramienta fue originalmente creada por Cynthia A. Brewer, del Departamento de Geografía de la Universidad de Pensilvania, como aplicación de la teoría del color a la cartografía.

<http://colorbrewer2.org>



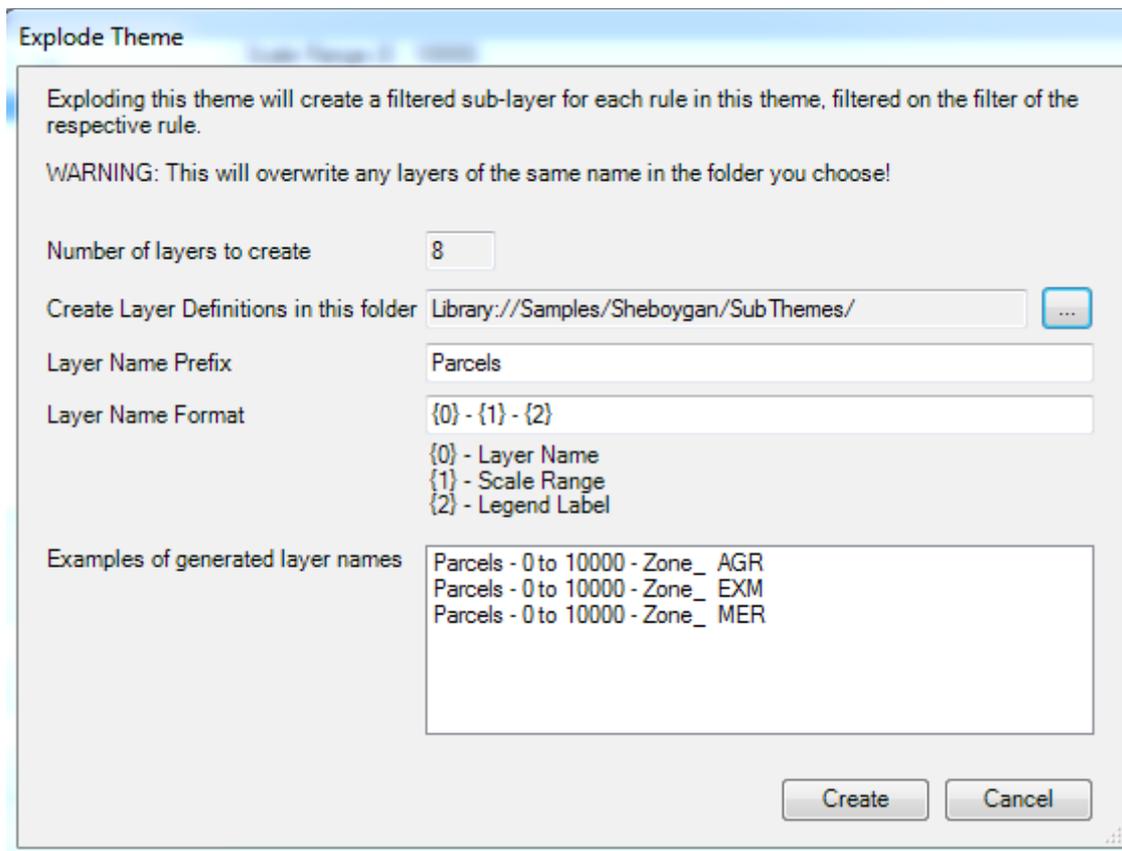
A veces puede ocurrir que al añadir una capa al tema aparezca un mensaje como este:



Esta ventana de advertencia se muestra por una sencilla razón, tener un tema con más de 100 clases es por lo general muy poco práctico:

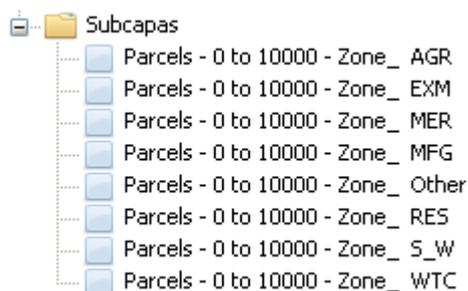
- En el visor web el panel de la pantalla no tiene suficiente espacio vertical para mostrar un tema tan grande.
- Repercute en gran manera en el rendimiento del servidor que podía llegar a quedarse colgado, por lo que hasta ahora automáticamente si ocurría estos casos se truncaba en la leyenda.

Para estos casos se utiliza el botón  que lo que hace es expandir un tema de gran tamaño en subcapas más manejables asignándoles un estilo de representación específico para cada una de ellas. Una vez apretado el botón aparecerá la siguiente ventana:



Aquí definiremos la carpeta de nuestro proyecto donde se guardarán los subtemas, así como el prefijo que queremos que identifique a cada uno de ellos. En la caja de texto inferior se nos muestra el ejemplo de como quedarán nombradas las capas.

Al pulsar sobre el botón “Create” se generarán loas subcapas en la carpeta designada.

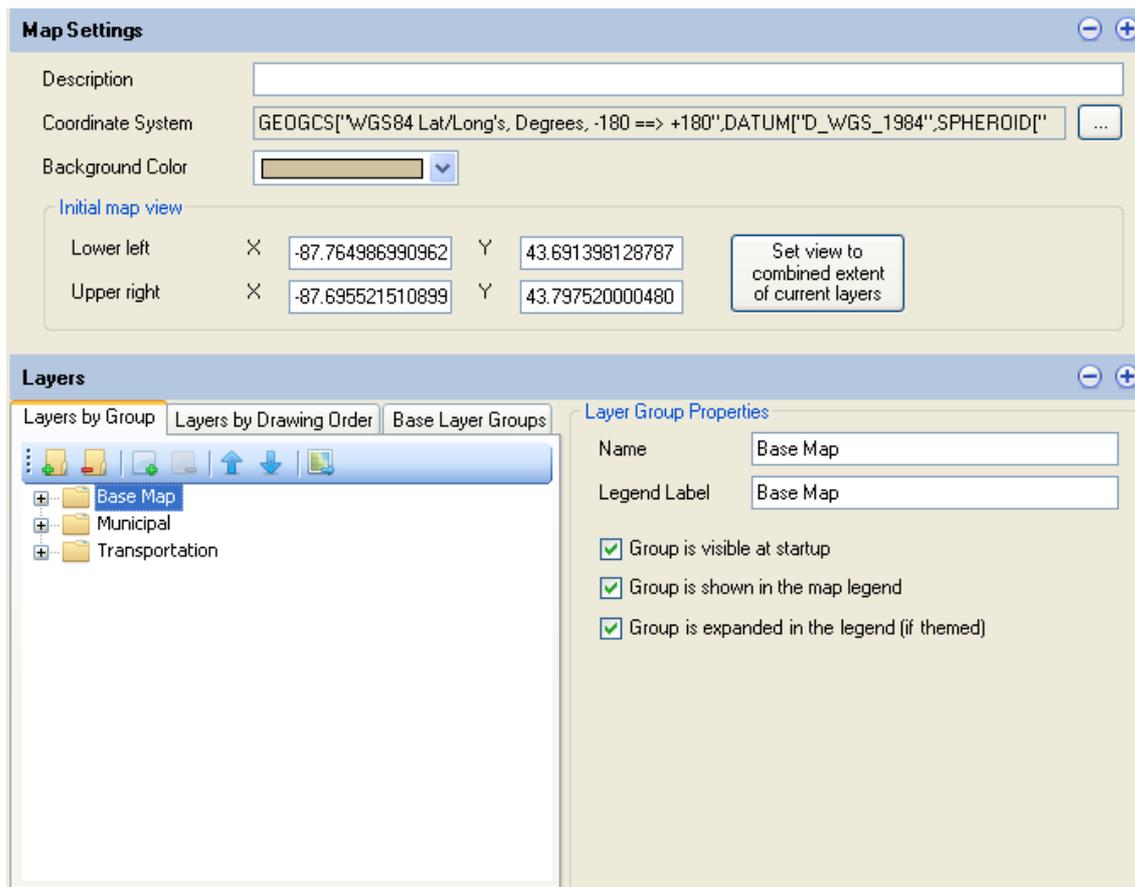


Si abrimos cualquiera de estas subcapas veremos que el tipo de elemento y la fuente de datos es la misma que la capa original, y que además hereda el filtro, el estilo y la escala de la regla. Desde aquí podremos generar temas que solo son aplicables a este tipo de subcapas.

## 4.4 El Mapa

Una **Definición de Mapa** (Map Definition) es un recurso que detalla la apariencia que tendrá un mapa en el visor cartográfico de nuestra página web.

Vamos a seleccionar la carpeta *Maps* existente en el panel de elementos de MapGuide y pulsamos doble clic sobre *Sheboygan*. Se abrirá el panel de edición donde se define cómo es el mapa en el que se mostrará la información sobre “Sheboygan”.



En primero lugar vemos un apartado llamado “**Map Settings**” (*Ajustes del Mapa*) con un cuadro de texto denominado “**Description**”. Aunque en nuestro ejemplo aparece vacío aquí podríamos escribir una breve descripción de lo que muestra el mapa.

A continuación aparece definido el sistema de coordenadas que se va a utilizar en el mapa. Este no tiene porque ser el mismo que el de la fuente de datos ya que MapGuide puede reproyectar los datos al vuelo (aunque es aconsejable). Si pulsamos sobre el botón  de la derecha se abrirá una nueva ventana donde se cargarán todos los sistemas de coordenadas disponibles, el cual se puede seleccionar bien por país, código WKT<sup>16</sup>, código en el sistema de coordenadas o bien por el código EPSG<sup>17</sup>. Una vez seleccionado el sistema pulsamos “**Validate**” y finalmente “**OK**”.

<sup>16</sup>WKT es el acrónimo inglés de *Well-Known Text* (literalmente *Texto Bien Conocido*). En el contexto de los sistemas de coordenadas y en el caso que nos atañe se refiere a un tipo de codificación que define los parámetros y métodos de transformación que se utilizarán para convertir coordenadas entre diferentes sistemas de referencia espacial.

<sup>17</sup>EPSG es el acrónimo inglés de *European Petroleum Survey Group*. Fué una organización -actualmente recibe otro nombre- vinculada al sector petrolero que recopiló en una base de datos un conjunto de parámetros geodésicos (elipsoides, datums, sistemas de coordenadas etc.) la cual ha sido ampliamente utilizado desde entonces en el ámbito de la geomática. Por extensión esa base de datos se la suele llamar EPSG y su codificación se ha convertido ya en un estándar.

**Select Coordinate System**

Select by list

Coordinate Category: Spain

Coordinate System:

Type WKT code

Well-Known-Text (WKT): PROJCS["ETRS89.UTM-30N",GEOGCS["LL-ETRF89",D Validate

Type coordinate system code

Coordinate system code: ETRS89.UTM-30N Validate

Type EPSG code

Coordinate system code: 25830 Validate

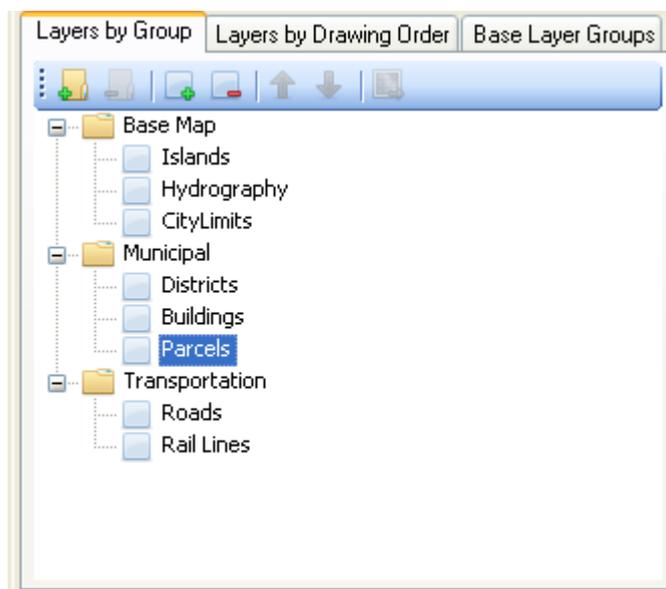
OK Cancel

A continuación en un menú desplegable "**Background color**" se puede seleccionar el color de fondo que tendrá el mapa y finalmente podremos ver las coordenadas de la vista inicial del mapa que coincidirán con la extensión máxima de las capas (esto se puede modificar al crear el *layout*).

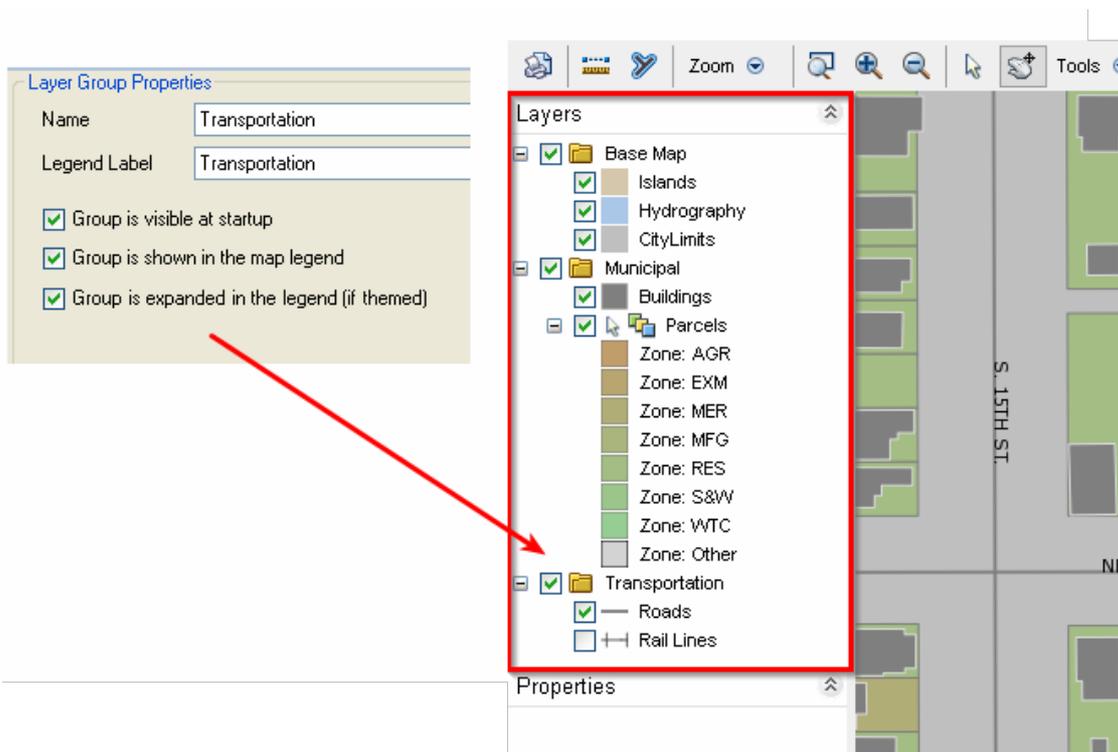
En la sección de capas observamos tres pestañas. En la primera, denominada "**Layers by Group**" (*Capas por grupo*) aparecen un conjunto de iconos que son para:

	Añadir un nuevo grupo de capas.
	Eliminar un nuevo grupo de capas.
	Añadir capa.
	Eliminar capa.
	Desplazar grupo/capa a un nivel superior.
	Desplazar grupo/capa a un nivel inferior.
	Crearlas un grupo de capas a partir de las capas seleccionadas.

	Mover la capa a la parte superior.
	Mover la capa a la parte inferior.

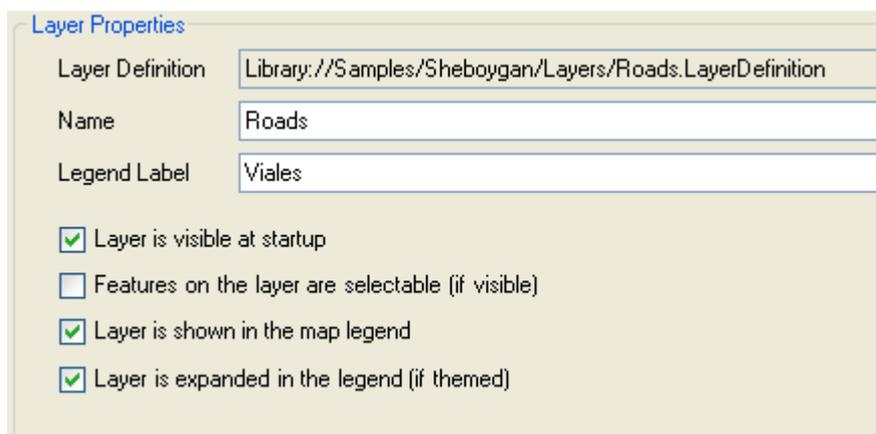


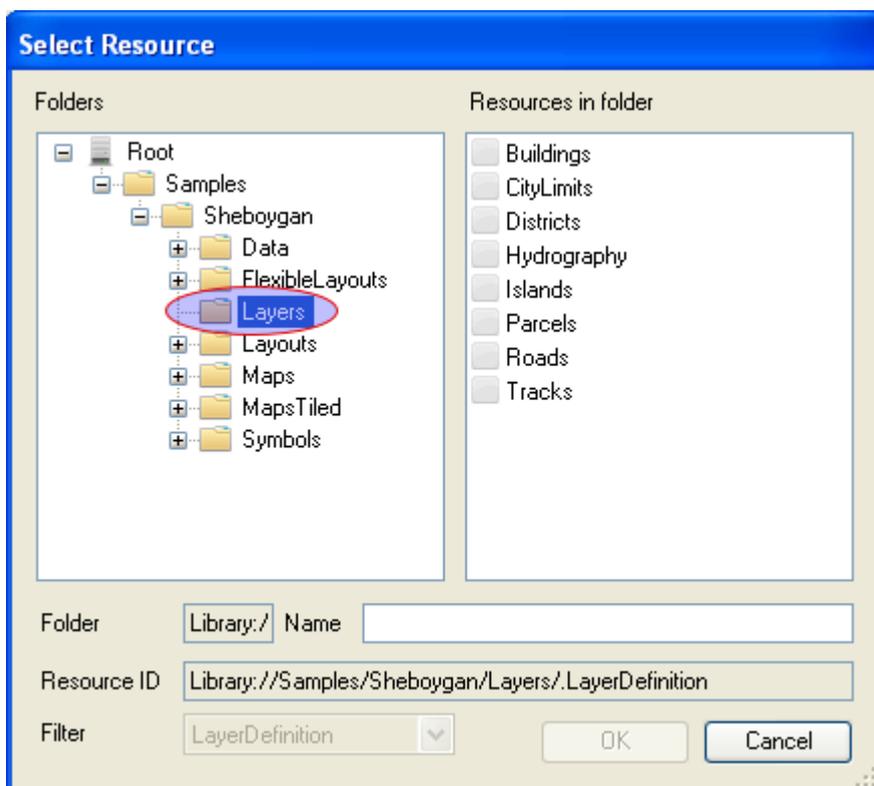
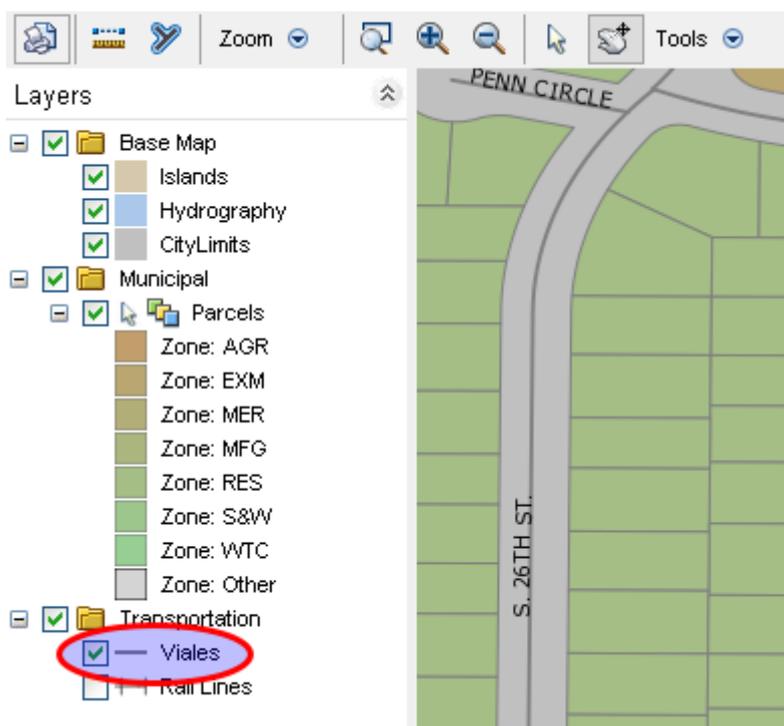
Un *Grupo* en MapGuide es una colección de capas. Si en el árbol de capas pulsamos sobre una de las carpetas en las que se agrupan veremos como en el apartado “**Layer properties**” (*Propiedades de la capa*) aparece el nombre que tiene este grupo de capas y la denominación que recibirán en la leyenda del visor cartográfico. Así mismo, podremos definir si el grupo será visible nada más arrancar la aplicación en el navegador, si el grupo será mostrado en la leyenda y si este se mostrará o no expandido.



Si pulsamos sobre el icono de añadir capa  se abrirá una nueva ventana en la que podremos seleccionar la capa que deseamos añadir al mapa de entre las que previamente hemos creado.

Como en el caso anterior podremos definir su nombre, el alias con que aparecerá en la leyenda, si será visible o no al inicio (esto irá en función del rango de escala que la asignemos), si los elementos serán seleccionables al pulsar sobre ellos con el cursor, si la capa se visualizará en la leyenda y finalmente si también será expansible en la leyenda (por ejemplo si posee diferentes tipo de simbología por categorías).

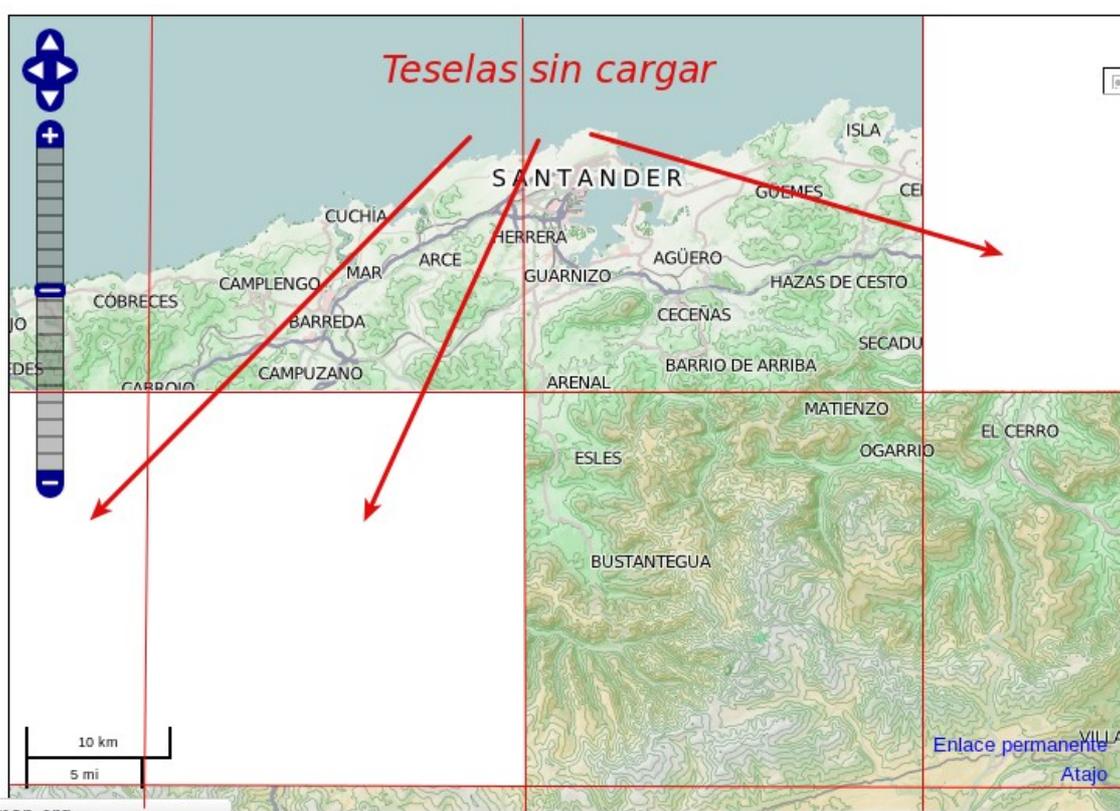




La pestaña “**Layers by Drawing Orders**” (*Capas por orden de dibujado*) permite definir el orden de visualización de las capas en función de su posición, con el fin de evitar que el solape impida su correcta visión.

Por último tenemos la pestaña “**Base Layer Groups**” (*Grupos de capas base*). Aquí podemos definir una colección de capas base para implementar un *TileCache*, o caché de teselas, y mejorar así el rendimiento del servidor. Vamos a explicar esta técnica más detalladamente.

Por lo general definiremos como capas base a aquellas cuyo contenido es constante, esto es, no cambian con frecuencia. Estas capas base conforman una única imagen la cual se puede cortar a modo de cuadrícula en teselas (*tiles*). Esto crea un conjunto de imágenes en mosaico que se cargan bajo demanda cuando son solicitadas por los usuarios al servidor, sirviendo aquellas dependiendo la zona en que se esté viendo el mapa. Los “trozos” de imágenes se almacenan en la *memoria cache*<sup>18</sup> del servidor por lo que si un nuevo usuario vuelve a visualizar esa misma zona del mapa el servidor no tiene que estar generando una nueva imagen cada vez que se hace una petición, sino que se envía la que ya existe.

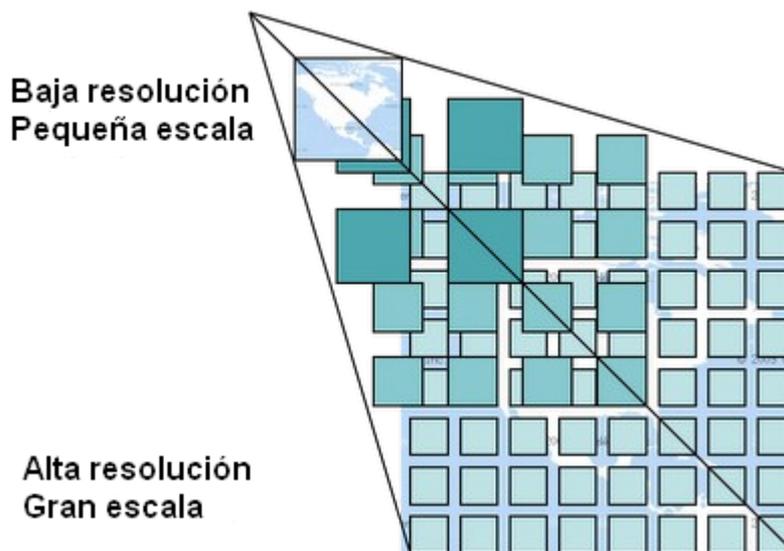


nap.org...

Fuente: Elaboración propia a partir de OpenStreetMap y contribuidores.

En el *tilecache* se emplea las denominadas *pirámides*, en las que se renderiza previamente el mapa a diferentes escalas de manera que cuando un usuario desde su ordenador solicite una porción de mapa al servidor esta ya esté lista para ser enviada, independientemente de la escala.

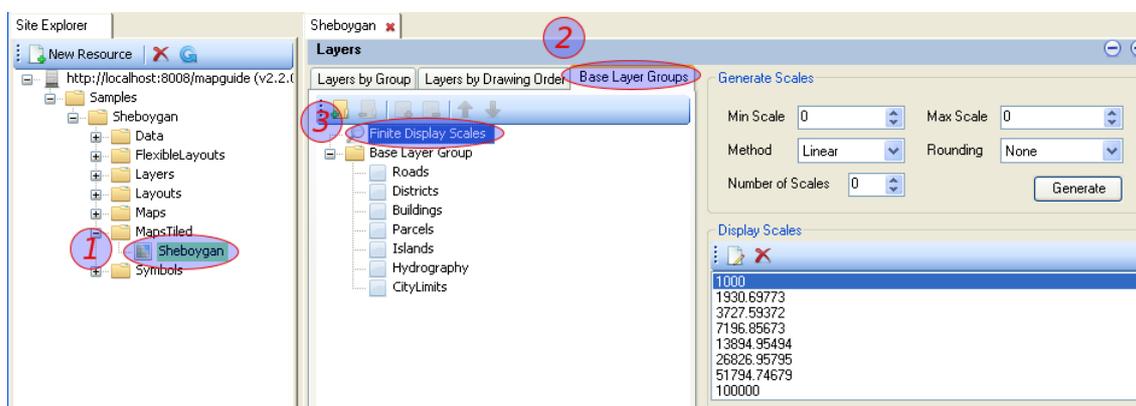
<sup>18</sup> Se define como *memoria cache* a aquella memoria en la que se almacena una serie de datos para su acceso rápido por parte del servidor.



Con esta técnica se consigue una menor sobrecarga del servidor y mayor rapidez en funciones de zoom y desplazamiento por el mapa. Ni que decir tiene que, con independencia de la infraestructura de que se disponga, cuanto más poblada de estos bloques de imágenes esté la memoria cache del servidor más fluida será la navegación por el mapa.

Por contra si los datos que sirve el servidor son muy dinámicos –se están actualizando casi de continuo– la técnica del teselado (*tiling*) y cacheado (*cache*) del mapa no es la más apropiada, siendo más conveniente el *single-tile*, o una única imagen por bloque de capa de mucha mayor dimensión.

Para ver un ejemplo de como se genera un caché de teselas con MapGuide Maestro vamos a abrir uno de los mapas que ya existe en el proyecto *Sheboygan* y que ya está configurado para ser teselado. Para ello expandimos la carpeta *MapsTiled* existente en el panel de recursos de Maestro y abrimos el único mapa que hay. A continuación nos vamos de nuevo a la pestaña “**Base layer groups**” (*Grupos de capa base*).



Vemos que se ha definido un único grupo de capas llamado “*Base Layer Group*” que recoge todas las capas del proyecto a las cuales se las ha dado diferentes propiedades de la manera que vimos con anterioridad. Se podría haber optado por

cachear únicamente aquellas capas cuyo contenido es más estático, y obviar aquellas otras cuya actualización es mucho más frecuente.



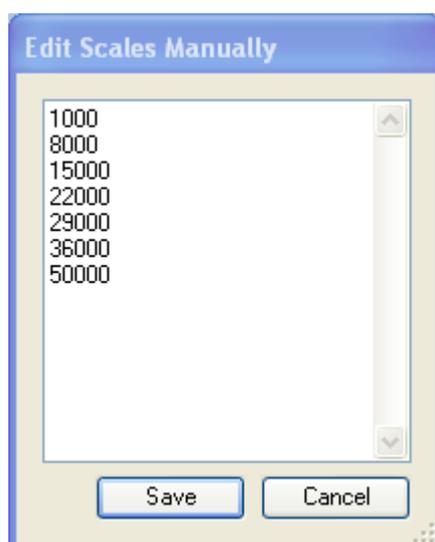
Si tienes una capa de datos que se actualiza frecuentemente (como por ejemplo las posiciones de un vehículo) o quieres que se pueda activar o desactivar en el visualizador, esta no deberá estar en esta pestaña "Base Layer Groups". MapGuide permite conjugar datos dinámicos y estáticos en un mismo mapa.

Si pulsamos sobre "**Finite display scales**" (*Escalas finitas de visualización*) se nos mostrará a la derecha una serie de opciones que podremos definir para crear nuestro cache de teselas.

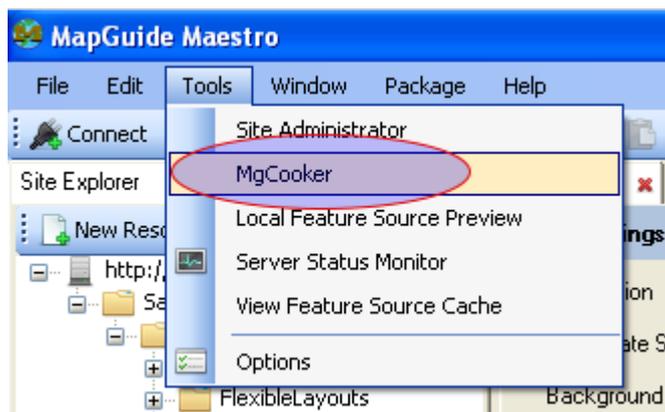
Dentro del apartado "**Generate scales**" (*Generar escalas*) introduciremos el denominador de la escala mínimo y máximo, el método para definir las diferentes escalas intermedias (lineal o exponencial), si a las escalas que nos genera Maestro hay que aplicarlas un redondeo para evitar decimales y ajustarlas a números enteros o si queremos que nos la aproxime a un número "bonito" terminado en cero. Finalmente especificaremos el número de escalas intermedias que existirán entre la escala máxima y la mínima.

Una vez tenemos definida la configuración que queramos pasaremos a crear las escalas pulsando sobre el botón "**Generate**". En nuestro caso nos advertirá si queremos sobrescribir las escalas ya existentes en el proyecto de Sheybogan. Decimos que sí y veremos como en la caja de textos "**Display scales**" no ha definido un nuevo conjunto de escalas.

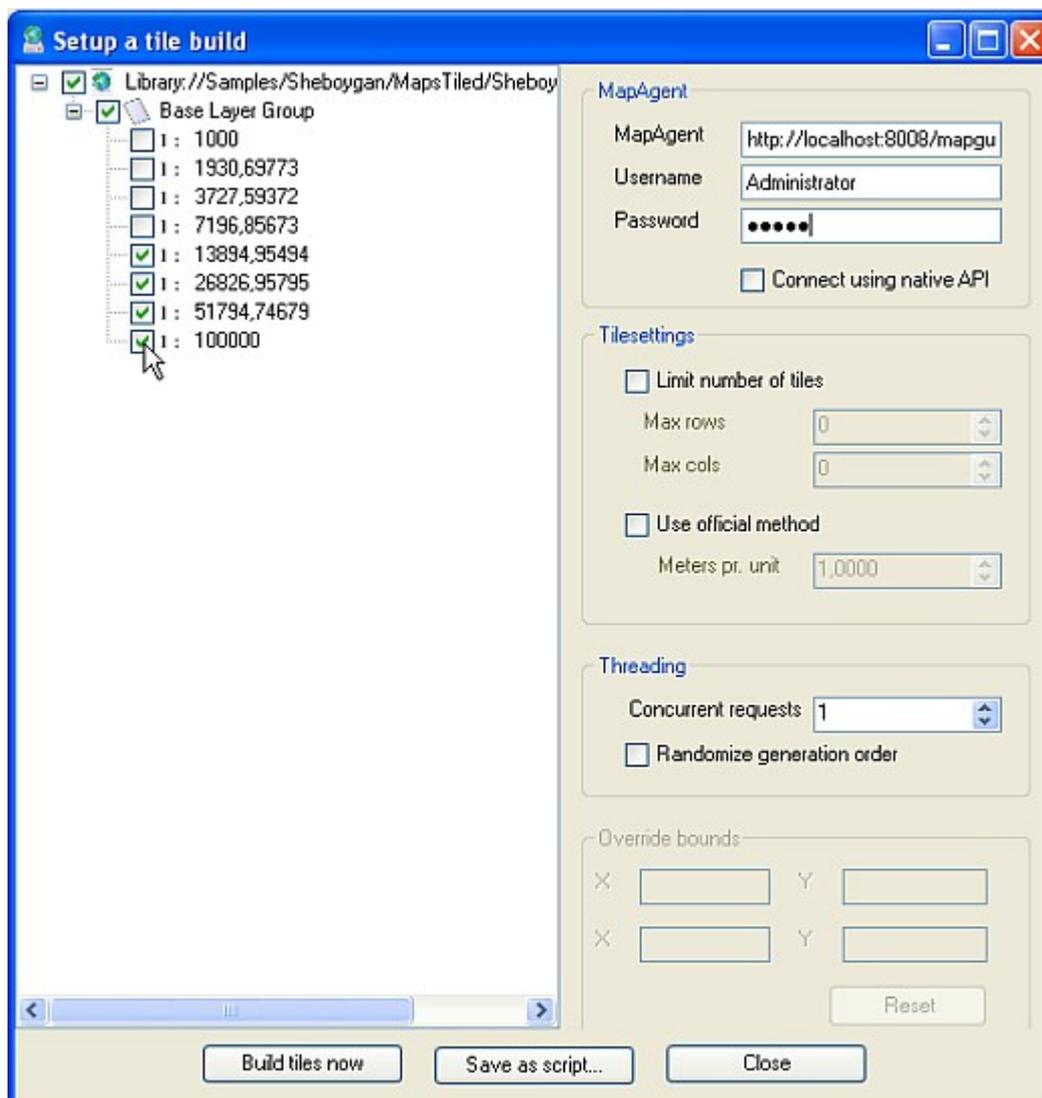
Si no estamos contentos con alguna de las escalas podemos editarla manualmente con el icono  o directamente eliminarla haciendo clic sobre el icono .



El último paso es prerenderizar las teselas del mapa. Para ello vamos al menú “**Tools**” y seleccionamos “**MgCooker**”, una herramienta que nos generará y copiará todas las teselas en las que se dividirá el mapa directamente en nuestro servidor cartográfico.



MgCooker puede ejecutarse también mediante consola. Tampoco es necesario abrir siempre MapGuide Maestro para usar MgCooker, ya que también se puede acceder mediante el menú de Windows **Inicio** → **Todos los programas** → **MapGuide Maestro**.

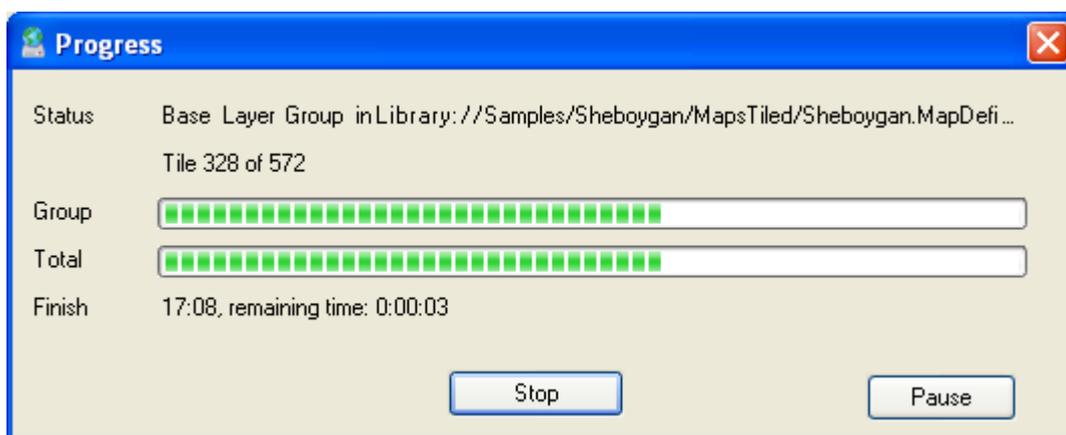


En el siguiente cuadro de diálogo definiremos a que niveles determinados de zoom o escala crearemos inicialmente las teselas. Lo más lógico es hacer principalmente aquellas que corresponden a un escala más pequeña porque serán las más utilizadas por lo usuarios, aunque podemos pedirle que las cree para todo el conjunto de escalas.

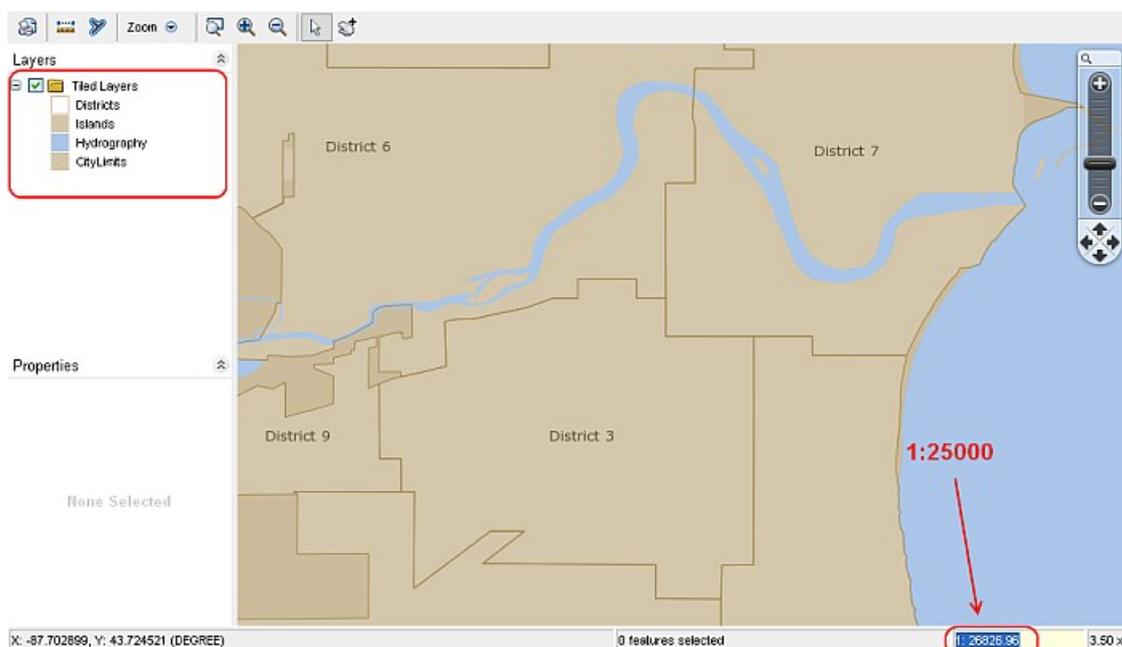
A continuación también podemos definir el número límite máximo de imágenes por filas y columnas de nuestra matriz de teselas o bien deducirlo en función del número de metros que queremos que abarque cada tesela del mapa. También podemos especificar el número de *tiles* que se crearan simultáneamente y si el orden en la creación de las teselas será aleatorio o no. Por defecto nosotros lo dejaremos tal y como está.

Finalmente pulsaremos el botón "**Build tiles now**". Se abrirá una ventana con dos barras de progreso. Cuando finalice ya tendremos nuestro cache de *tiles* listo en nuestro servidor MapGuide.

Es interesante el botón **“Save as script...”** ya que sirve para generar un *script* en un archivo .bat de proceso por lotes con la configuración que hemos elegido para teselar y cachear el mapa. Así podemos, por ejemplo, crear en Windows una tarea programada que a determinada hora del día ejecute este archivo .bat y nos vuelva a crear una nueva cache de teselas del mapa. Un procedimiento interesante cuando debemos actualizar periódicamente las capas del mapa pero no queremos que esta repercuta en el rendimiento del servicio cartográfico que ofrecemos.



Si realizamos un previsualización del mapa veremos que todas las capas están agrupadas en una misma capa base, en nuestro caso llamada *“Tiled layer”*, la cual está cacheada. Si modificamos la escala manualmente en el visor, introduciendo por ejemplo 1:25000 y pulsando *Enter* en el teclado, veremos que el mapa se ajusta a la escala de visualización más aproximada que se haya definido al crear las teselas, en nuestro caso 1:26826,95795.



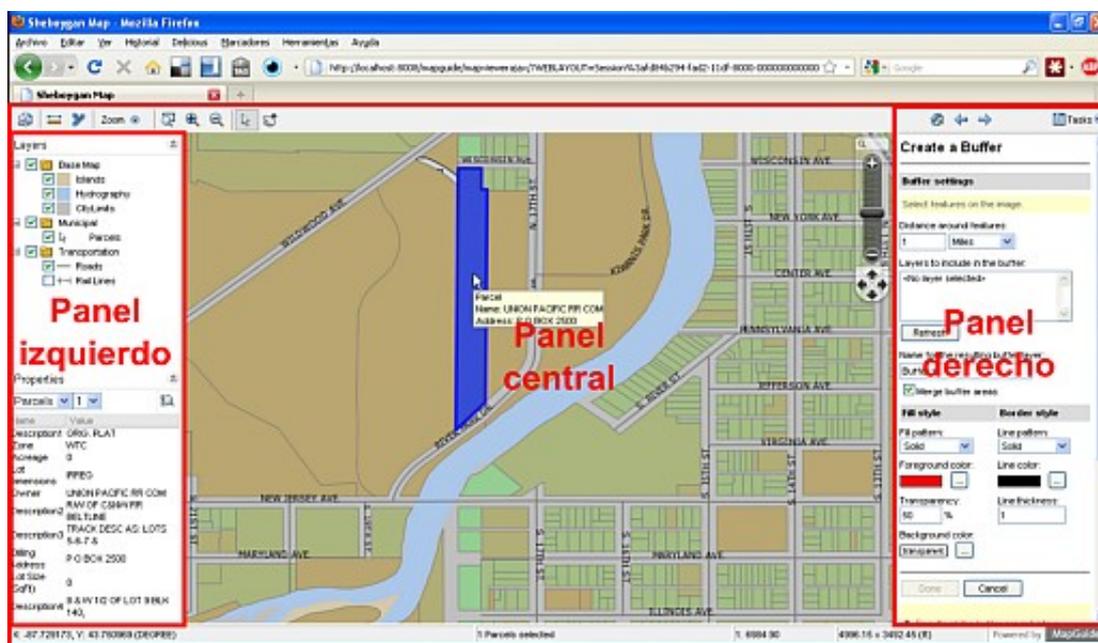
## 4.5 El diseño del visor cartográfico

Una vez ya configurado el mapa nuestro último paso es diseñar el visor cartográfico que los usuarios verán en su navegador. Para ello se crea un **Web Layout**, que no es más que una composición de paneles o marcos que contienen el mapa que hemos definido en el paso anterior. Describe la interfaz de usuario y las funcionalidades del visor AJAX básico.

En el ejemplo de *Sheboygan* que estamos siguiendo vamos a ver uno de los *layouts* ya creados. Para ello desde el panel de recursos desplegamos el árbol y en la carpeta **Layouts** abrimos el diseño *SheboyganPhp* con un doble clic.

A la hora de diseñar un *Web Layout* podemos definir lo que se va a ver en tres secciones del navegador:

- El **panel de la izquierda**, el cual recoge el gestor de capas y las propiedades de los elementos seleccionados.
- El **panel central**, que contiene básicamente la barra de herramientas, la barra inferior de estado y el mapa.
- El **panel de la derecha**, que reúne una barra de tareas y un marco destinado a tareas como consultas, geoprocursos, etc.



En la pestaña de edición vemos en primer lugar una caja de textos para introducir el título que tendrá la ventana del navegador.

Seguidamente en “**Map Resource ID**” se selecciona el mapa que previamente hemos definido en el apartado anterior. Para ello pulsamos el botón  de la derecha y seleccionamos el mapa, en nuestro caso el mapa no teselado de Sheybogán.

A veces puede ocurrir que el usuario que está viendo el mapa le caduque la sesión en el servidor MapGuide debido a que ha estado inactivo durante demasiado tiempo (no ha realizado ninguna operación en el visor). En tal caso le aparecerá en el navegador un mensaje “*resource not found*” (*recurso no encontrado*). Si queremos asegurarnos que la sesión del usuario se mantengan activa y no caduque se puede marcar la opción “**Keep Connectio Alive**” (*Mantener conexión viva*).



Esta característica solo aparecerá si el recurso de nuestro proyecto está actualizado a la última versión del esquema. Si no es así MapGuide Maestro nos ofrece la posibilidad de actualizarlo desde el botón “**Upgrade**”

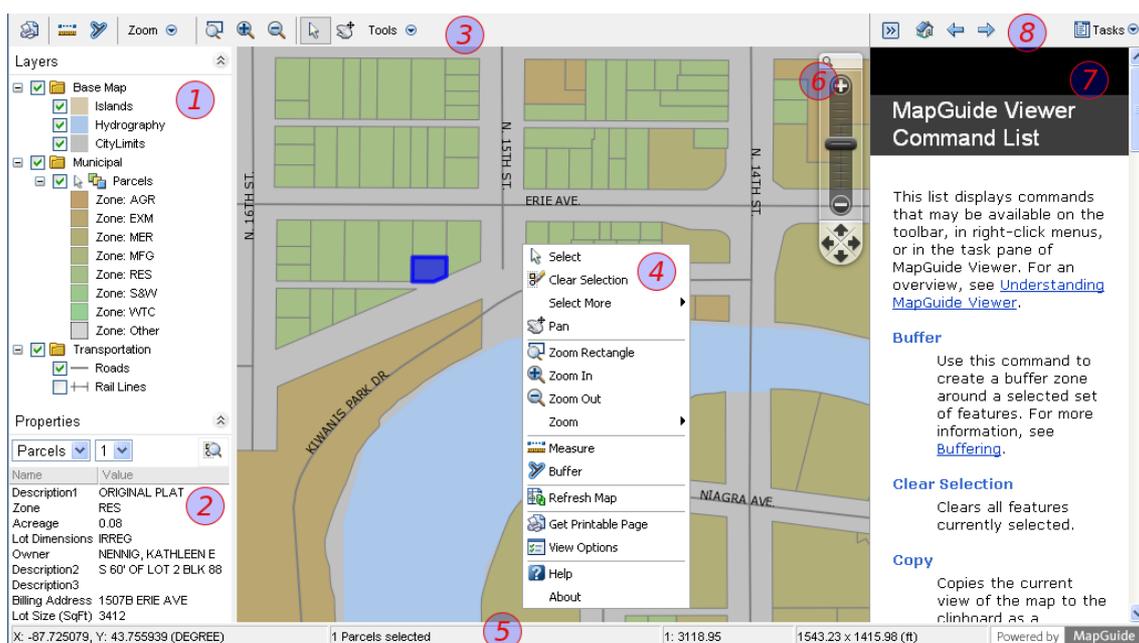
This resource can be upgraded to a newer version to take advantage of newer feature

 Upgrade

La opción de “**Override the Map's initial position and scale**” (*Sobrescribir la escala y extensión inicial del mapa*) nos permitiría definir las coordenadas del centro del mapa así como su escala inicial, lo que permite calcular a MapGuide el llamado *bounding box* o recuadro que circunscribe la vista máxima del mapa que podremos ver en el visor. Si lo activamos esta prevalecería sobre la vista inicial del mapa que especificamos con anterioridad.

A continuación en el apartado “**Item Visibility**” (*Objetos visibles*) se establece los elementos que tendrá cada panel del visor, así como su ancho en píxeles. Los elementos pueden ser:

1. Un panel para gestionar el control de capas (*Layer Control*).
2. Un panel que muestre las propiedades de los elementos seleccionados (*Properties Panel*).
3. Una barra de herramientas superior (*Toolbar*).
4. Un menú contextual al hacer clic con el botón derecho del ratón (*Context Menu*).
5. La barra inferior de estado (*Status Bar*).
6. El control de zoom (*Zoom Control*).
7. El panel de tareas (*Task Panel*).
8. Con su barra de tareas (*Task Bar*).



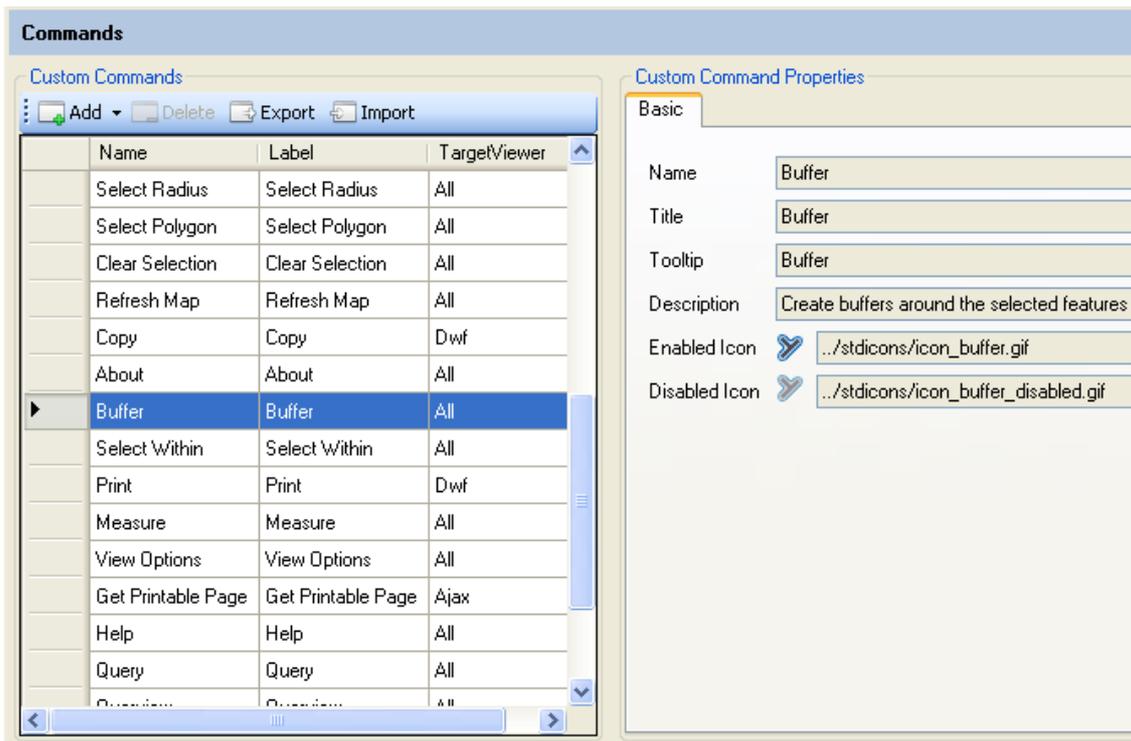
En la caja de textos llamada “**Initial Task Pane URL**” (*URL inicial del panel de tareas*) se puede especificar la ruta relativa al archivo que deseamos se muestre inicialmente al arrancar el visualizador en el panel derecho, como por ejemplo un archivo en HTML con la presentación al usuario del servicio cartográfico que va a utilizar. Si no se especifica ningún archivo se mostrará por defecto la ayuda de MapGuide.

Si se ha definido en los ajustes de la capa que determinados elementos del mapa posean un enlace, podemos señalar en la lista desplegable “**Feature Hyperlink Target**” (*Destino donde mostrar el hiperenlace*) donde se abrirán estos hipervínculos: bien en el panel derecho de tareas (*TaskPane*); en una nueva ventana (*NewWindow*); o bien, si existe, en un *frame* o marco de la página que definamos nosotros (*SpecifiedFrame*).

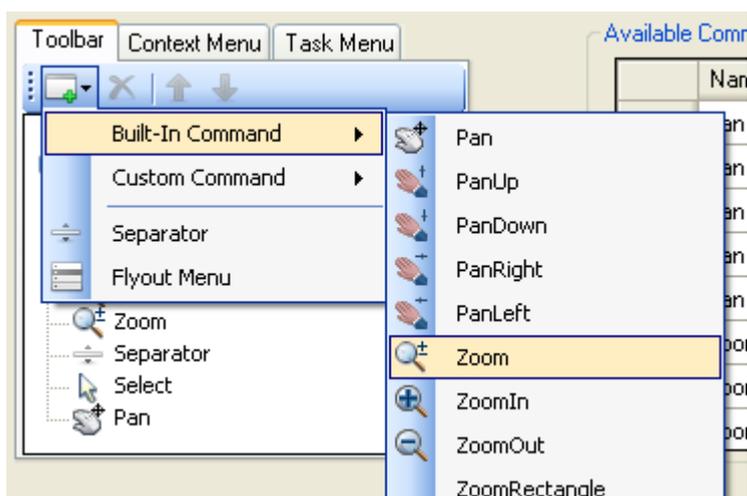
Con MapGuide Maestro podemos definir cuales son las herramientas y menús **-Menus and Toolbars-** que se utilizarán en el *layout*. Existen numerosas funciones ya preprogramadas que no hacen falta desarrollar -como el *zoom*, el *pan*, la selección de elementos, etc.- o bien podríamos crear nuestro propio botón o elemento del menú que llamase a un *script* o código HTML. Se podría personalizar el interfaz del visualizador para, por ejemplo, lanzar funciones de geoprocésamiento almacenadas en un servidor o para crear un informe personalizado de un elemento seleccionado del mapa.

De esta forma vemos que, para el ejemplo de *Sheboygan*, se ha definido una serie de herramientas para los menús de la barra de herramientas, el menú contextual y el menú del panel de tareas.

A cada herramienta elegida se le da en el apartado “**Commands**” (*Comandos*) unas propiedades básicas como un nombre identificativo a nivel interno, un título o nombre que aparecerá en el visor, un texto para el *tooltip*, una descripción de lo que hace, la ruta relativa a un archivo de imagen que se mostrará como icono para cuando es posible el uso la herramienta y otra para cuando está desactivada.



Para añadir una nueva herramienta seleccionamos la pestaña correspondiente el menú donde queremos insertarla, pulsamos sobre el icono  y seleccionamos **“Built-In Command”** (*Comando ya incorporado*). Nos aparecerá toda una serie de herramientas ya preprogramadas con la funciones más comunes.

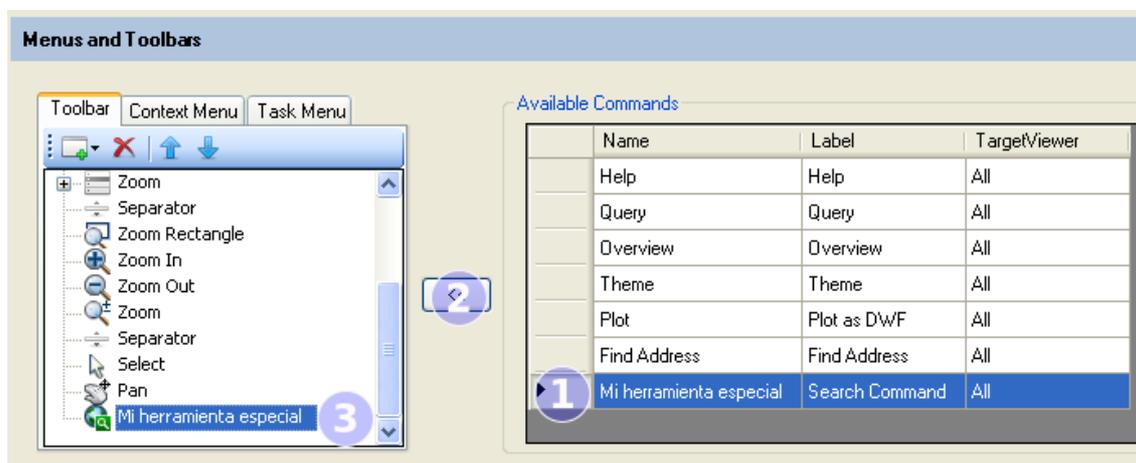


Pero como hemos comentado con anterioridad, también podemos agregar nuevas funcionalidades personalizadas al visor. Para adaptar alguno de estos comandos (por ejemplo que la herramienta de búsqueda actúe sobre una capa determinada u otra

que ejecute un *script* o llame a una página web) en el apartado “**Custom Commands**” (*Comandos personalizados*) pulsaremos el icono  y seleccionaremos:

- **Invoke URL.** Invoca una dirección web especificando en que marco o ventana se abra la página. Permite pasar variables por URL (GET).
- **Invoke Script.** Invoca a un programa simple escrito en algún lenguaje de programación web (*Javascript, PHP, ASP, etc.*) para que se ejecute.
- **Search.** Permite definir una búsqueda sobre los atributos de los elementos de una capa.

Una vez seleccionado ya está disponible en el apartado “**Available Commands**” (*Comandos disponibles*) para poder asignarle a los diferentes menús o barras de herramientas.



## 4.5 La aplicación Fusión



Aunque Fusión cae fuera de este tutorial -ya que se necesita estar familiarizado con HTML, Javascript, CSS- explicaremos brevemente qué es y cuál es su relación con MapGuide.

**Application Definition** (*Definición de aplicación*) es un recurso que proporciona instrucciones a un *Flexible Layout* sobre cómo construir una aplicación de mapas web. Se determina qué componentes funcionales (llamados *widgets*) estarán presentes en la aplicación, determinados aspectos de cómo se presentarán estos *widgets* y la configuración opcional de la funcionalidad de cada *widgets*.

Vamos a aclarar un poco más todo esto.

Dentro de MapGuide Maestro se denomina *Flexible Layouts* a *Fusion*. *Fusion* es un marco de desarrollo (framework) de aplicaciones *web-mapping* creadas en el lenguaje *javascript* para los servidores de mapas MapGuide OS y Mapserver. Básicamente permite a desarrolladores de webs no-espaciales crear aplicaciones web enriquecidas de manera rápida y sencilla.

Se fundamenta en el API de *javascript* de OpenLayers, hoy por hoy un estándar *de facto* en el desarrollo de visores de mapas interactivos en los navegadores web. *Fusion* permite a los desarrolladores crear *widgets* para añadir, eliminar o modificar funcionalidades del visor cartográfico. Todo ello siendo compatible con los estándares HTML y CSS, por lo que es funcional en la mayoría de navegadores web de Windows, Mac o Linux.

Inicialmente Fusión se desarrolló para el servidor de mapas MapGuide, pero debido a su versatilidad ha sido portado también a MapServer.

Para crear un *Flexible Layouts* deberemos seleccionar *New Resource -> Application Definition* y a continuación elegir la definición de mapa que queremos asignar a la plantilla de *Fusion*.

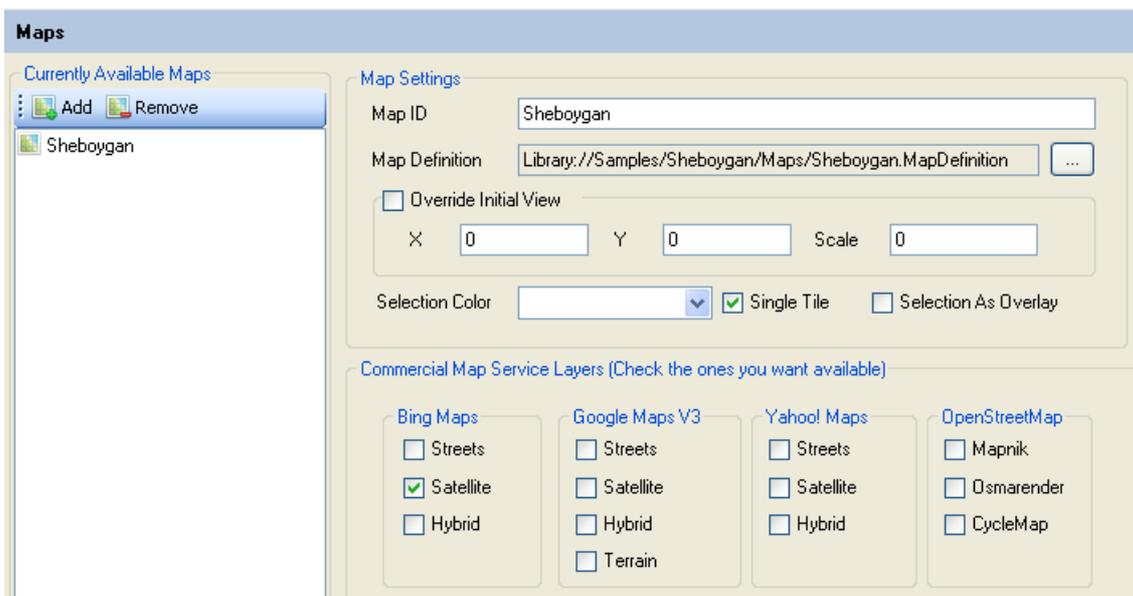


Una de las ventajas más visibles del uso de los *Flexible Layouts* quizá sea el poder integrar en nuestro visor cartografía de servicios de mapas comerciales externos ampliamente utilizados, como Google Maps.

A diferencia de los visores basados en plantillas de *Fusion*, el visor AJAX que usa por defecto MapGuide no utiliza la biblioteca de *OpenLayers*, siendo esta última probablemente la única forma legítima de usar las capas de Google Maps.<sup>19</sup>

Debes tener en cuenta que si utilizas alguno de estos servicios de cartografía (Google Mapg, Bing Maps u OpenStreetMap) tu *Definición de Mapa* (Map Definition) deberá estar en el, ampliamente utilizado por estos servicios, sistema de coordenadas WGS84/Pseudo Mercator (EPSG:3857) y es posible que debas reconfigurar la vista inicial del mapa.

<sup>19</sup>El API de Google Maps tiene restricciones severas sobre cómo se debe utilizar su servicio de mapas por terceros. Por otro lado, desde 2012 Google ha cambiado la política de uso de su producto Google Maps, pasando a cobrar por la utilización del API a sitios web que generen mucho tráfico: aquellos que cargan diariamente más de 25.000 mapas básicos o 2.500 mapas avanzados.



El servidor MapGuide reproyectará al vuelo las capas vectoriales. Ten en cuenta que la reproyección penaliza el rendimiento del servidor, por lo que suele ser más aconsejable crear el proyecto de MapGuide con todas las fuentes de datos en el sistema de coordenadas en que serviremos el mapa.

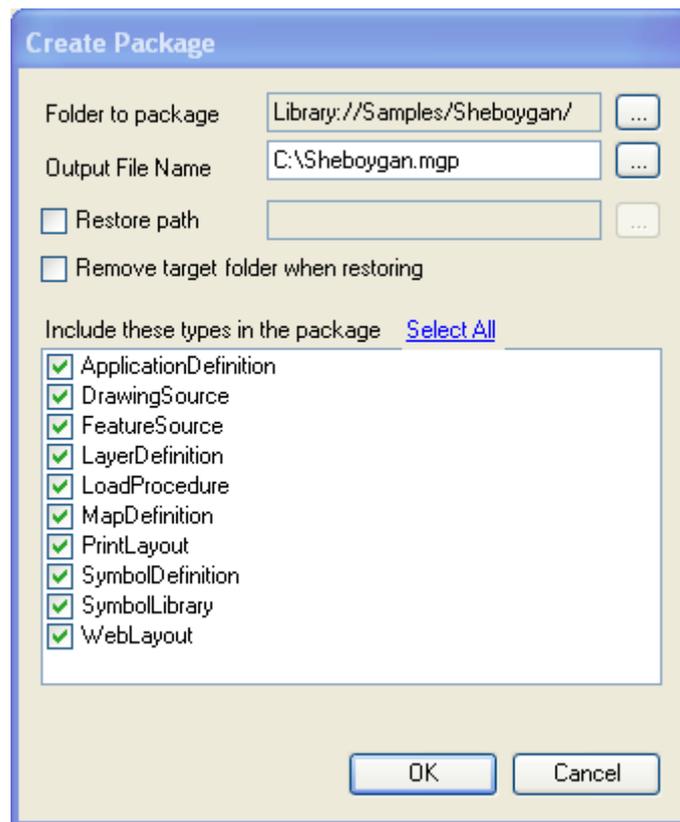
Te en cuenta que si imprimes o guardas el mapa a través del visor de Fusión no podrás no podrás ver la capa de Google Maps o de Bing Maps porque esto muy probablemente supondría otra violación de los términos del servicio.

## 5. Generar el paquete de datos y cargarlo en el servidor

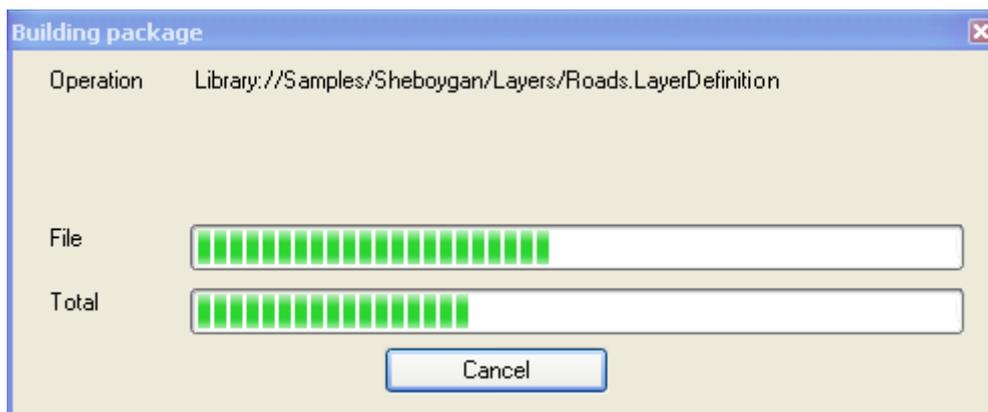
Una vez tenemos completado nuestro proyecto en MapGuide Maestro el siguiente paso es crear un paquete de datos que recoja todos los recursos de nuestro proyecto y nos lo guarde en un único archivo comprimido con extensión \*.mpg. Para ello en Maestro seleccionamos en el árbol de recursos la carpeta de donde cuelgan **todos nuestros recursos** (en nuestro caso Samples/Sheboygan/) y a continuación hacemos clic en el menú **Package** → **Package folder...** y se no abrirá un cuadro de diálogo.



Antes de crear el paquete podemos verificar el proyecto haciendo clic sobre la carpeta *Sheboygan* con el botón derecho del ratón y seleccionando **Validate**.



En el apartado “**Output File Name**” (*Nombre del archivo de salida*) seleccionaremos la ruta donde queremos guardar el paquete y le daremos un nombre a este (*Sheboygan.mgp*). Finalmente pulsaremos **OK**. Al poco rato tiempo Maestro nos habrá empaquetado todos nuestros recursos en un único archivo comprimido en la carpeta seleccionada.



Ya podemos cerrar MapGuide Maestro.

## 6 Cargar un paquete en el servidor de mapas

Hasta este momento no hemos tenido necesidad de instalar directamente un paquete en el servidor de mapa MapGuide ya que hemos desarrollado un proyecto directamente en el servidor de mapas a través de Maestro.

Supongamos ahora que otra persona nos pasa un proyecto y nos pide que lo subamos al servidor MapGuide y no disponemos de Maestro.

Para ello deberemos abrir las páginas de administración del servidor. El sitio administrador de MapGuide es el *back-end* del servidor. Para que nos hagamos un idea el *back-end* es como la trastienda o parte privada de nuestro servidor, a la que solo podrán acceder los administradores. Es el contrapunto al *front-end*, o la parte visible que ve cualquier usuario cuando se conecta (en nuestro caso el visor cartográfico).

El primer paso es copiar el archivo *Sheboygan.mgp* que hemos generado en el directorio donde se almacenan los paquetes en el servidor de mapas, dentro de la carpeta de instalación de MapGuide:

### C:\Archivos de programa\OSGeo\MapGuide\Server\Packages

Una vez copiado el siguiente paso es acceder al sitio administrador de MapGuide. Para ingresar a las páginas de administración del servidor cartográfico se puede hacer, bien a través de Maestro, pulsando el menú **Tools** → **Site Administrator**, o bien abriendo el navegador y escribiendo la URL:

**<http://localhost:8008/mapguide/mapadmin>**

Esta última manera es la más recomendable porque permite acceder al *Administrador* desde cualquier ordenador, independientemente si MapGuide Maestro está instalado o no.



Recuerda que en el caso de que estemos intentando acceder al *Administrador* desde un equipo diferente a donde esté ejecutándose el servidor MapGuide deberás sustituir *localhost* por la IP del equipo remoto.

Una vez se ha accedido al sitio de administración se nos mostrará primeramente un ventana de validación de usuario. Por defecto los datos de usuario y contraseña son los siguiente:

- Administrator ID: **Administrator**
- Password: **admin**

Es muy importante cambiarlos desde las páginas de administración una vez tengamos el servidor cartográfico en fase de producción.

A continuación vamos a cargar el paquete. Para ello en el menú de la izquierda del Administrador seleccionamos "**Manage Packages**" y en la sección "**Load Package**"

verás el paquete Sheboygan.mgp. Si no es así pulsa el botón **“Refresh”** para actualizar la lista.

## Load Package

Packaged resources or data in the package folder can be loaded to this site server.  
[Configure Packages directory.](#)

	Package Name	Size (bytes)	Status
 Load Package			
 View Log			
 Delete			
 Refresh			
	Sheboygan.mgp	13,912,231	Unknown

Si aún así sigues sin poder verlo vete al menú **“Configure Services”** y asegurate de que la ruta en **“Packages folder”** es la misma que donde copiaste el paquete.

## Resource Service

Data file trash folder:	Trash
Library data file folder:	C:/Archivos de programa/OSGeo/MapGuide/Server/Repositories/Library/DataFiles/
Library repository folder:	C:/Archivos de programa/OSGeo/MapGuide/Server/Repositories/Library/
Session data file folder:	C:/Archivos de programa/OSGeo/MapGuide/Server/Repositories/Session/DataFiles/
Session repository folder:	C:/Archivos de programa/OSGeo/MapGuide/Server/Repositories/Session/
Site repository folder:	C:/Archivos de programa/OSGeo/MapGuide/Server/Repositories/Site/
Resource schema folder:	C:/Archivos de programa/OSGeo/MapGuide/Server/Schema/
Repository checkpoints timer interval (seconds):	600
Packages folder:	C:/Archivos de programa/OSGeo/MapGuide/Server/Packages/

Seleccionamos en el botón de opción del paquete y pulsamos el icono **“Load Package”**. Una vez que paquete se haya cargado correctamente la columna de estado (*Status*) pondrá *Load Succeeded*.

Si se ha producido algún problema podemos revisar el archivo de incidencias .log pulsando el icono **“View Log”** o bien abriéndolo en el bloc de notas. El archivo se encuentra en:

**C:\Archivos de programa\OSGeo\MapGuide\Server\Packages\Sheboygan.mgp.log**

A partir de ahora si volvemos a abrir MapGuide Maestro nos deberá aparecer en el árbol de recursos nuestro proyecto de Sheboygan. Es posible que debamos refrescar el árbol pulsando sobre el botón 

## 7. Visualizar el proyecto en el navegador

El último paso para poder ver el mapa es bajarnos de la web de MapGuide el visor de ejemplo programado en PHP descargando el archivo **phpviewersample.zip**



Descárgate el visor de ejemplo desde el sitio web de MapGuide:

<http://MapGuide.osgeo.org/download/releases/2.0.x-samples>

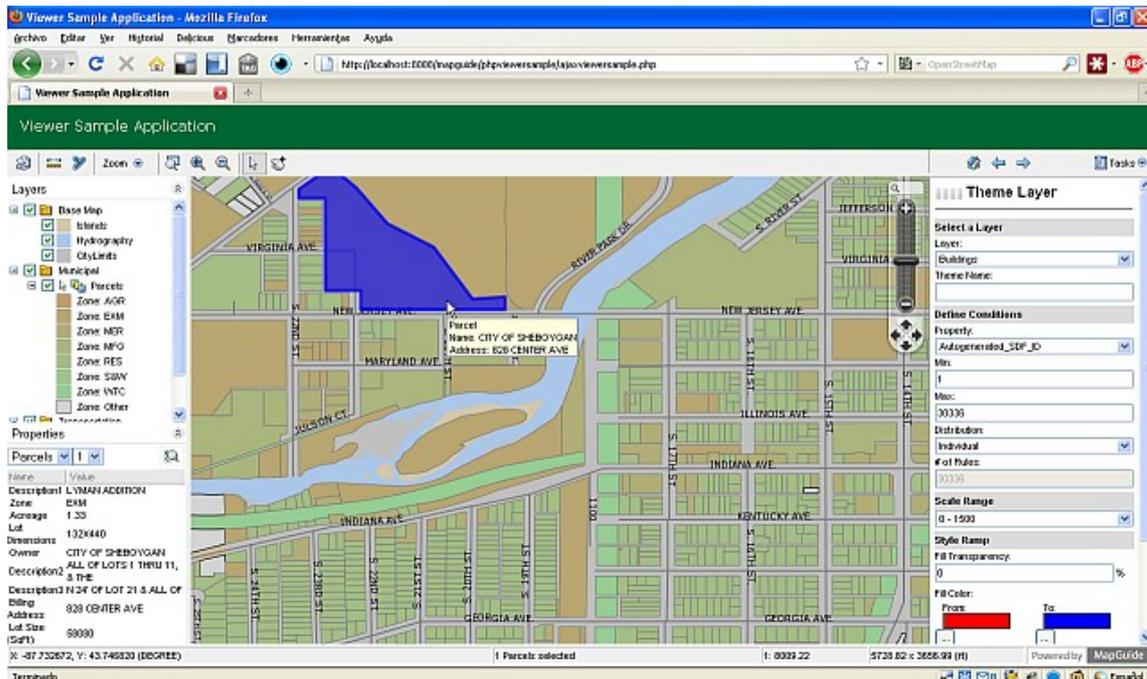
Una vez descargado descomprímelo y copia la carpeta **phpviewersample** al directorio público del servidor MapGuide:

**C:\Archivos de programa\OSGeo\MapGuide\Web\www\**

Finalmente abre el navegador y escribe la siguiente URL:

**<http://localhost:8008/mapguide/phpviewersample/ajaxviewersample.php>**

Et voilà! Ya tenemos listo nuestro servicio de publicación de mapas en un entorno web.



Es importante señalar que el archivo "ajaxviewersample.php" está configurado para mostrar el layout del ejemplo de Sheboygan. Si editamos este archivo con el bloc de notas veremos la línea de código PHP:

```
$webLayout = "Library://Samples/Sheboygan/Layouts/SheboyganPhp.WebLayout";
```

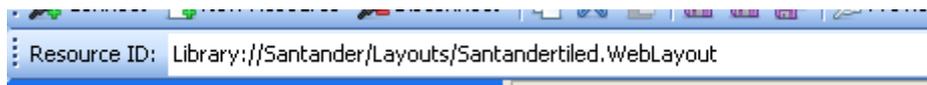
Esta hace referencia al proyecto Sheboygan y al *layout* con el nombre "SheboyganPhp". Si creamos un nuevo proyecto con un nombre diferente deberemos modificar esta línea sustituyendo el contenido que está entrecomillado por el del nuevo proyecto.



Nunca edites un archivo con código en un procesador de textos. Utiliza un editor de texto plano como el bloc de notas, ya que muestra todos los caracteres que hay en el archivo y, a diferencia de un procesador de texto, escribe solo texto, sin formato ni diagramación.

Para ello nos vamos a MapGuide Maestro y seleccionamos el *layout* que queremos mostrar en el visor cartográfico y pulsamos el botón derecho del ratón.

A continuación copiamos el identificador del recurso **Resource ID**.



Pegamos el contenido en la línea de código sustituyendo a la anterior:

```
$webLayout = "Library://Santander/Layouts/Santander.WebLayout";
```

Guardamos el archivo "ajaxviewersample.php" y si volvemos a cargar en el navegador la URL <http://localhost:8008/mapguide/phpviewersample/ajaxviewersample.php> veremos que el *layout* ha cambiado al del nuevo proyecto.



Si tienes varios proyectos cargados en el servidor MapGuide renombra cada carpeta que contiene su visor con un nombre que le identifique. Por ejemplo:  
<http://localhost:8008/mapguide/sheboygan/visor.php>  
<http://localhost:8008/mapguide/santander/visor.php>

## 8. Enlaces de interés

- Sitio oficial de MapGuide Open Source  
<http://bit.ly/mapguideos>
- Sitio oficial de MapGuide Maestro  
<http://bit.ly/mapguidemaestro>
- Listas de correo oficiales de MapGuide Open Source en Nabble  
<http://bit.ly/listasmaguide>
- Buenas prácticas con MapGuide OS  
<http://bit.ly/buenaspracticasmaguide>
- Arquitectura de MapGuide  
<http://bit.ly/arquitecturamaguide>
- Documentación de la API de MapGuide 2.0  
<http://bit.ly/apimaguide>