

## Contingut

- 1 Introducción
- 2 Objetivos
- 3 Anatomía de un mapa: los tiles
- 4 Mapas personalizados
  - ◆ 4.1 Mapa base
  - ◆ 4.2 Mapa de superposición
- 5 Mapas basados en fuentes WMS
  - ◆ 5.1 El servicio OGC/WMS
  - ◆ 5.2 Petición de datos
  - ◆ 5.3 Las coordenadas de un tile
  - ◆ 5.4 Las coordenadas mundiales
  - ◆ 5.5 Transformación de coordenadas
  - ◆ 5.6 El primer mapa WMS
  - ◆ 5.7 Más mapas WMS
    - ◇ 5.7.1 Mapa WMS superpuesto a los mapas de Google Maps
    - ◇ 5.7.2 Dos mapas WMS procedentes de una única fuente combinados
    - ◇ 5.7.3 Dos mapas WMS procedentes de fuentes distintas combinados
- 6 Resumen
- 7 Notas

## Introducción

En los módulos anteriores, hemos aprendido a programar herramientas SIG utilizando siempre la cartografía que acompañaba a la aplicación: en el caso de Google Maps, los cuatro tipos de mapa que nos ofrece su API, y en el caso de gvSIG, los mapas que habíamos obtenido previamente de algun servidor de mapas.

Sin embargo, la proliferación de los servicios de cartografía en línea ha hecho posible una nueva forma de trabajar en la que el cliente no dispone ya de una copia propia de los datos, sino que accede continuamente a servicios en línea que mantienen una copia actualizada de esos datos.

En este módulo, aprenderemos a utilizar la cartografía disponible en línea, es decir, a interactuar con los servicios OGC desde nuestro código para obtener todo tipo de mapas e incorporarlos a nuestra aplicación SIG.

Concretamente, aprenderemos a utilizar un servicio OGC/WMS desde una aplicación web basada en Google Maps, de la misma manera que lo hacen aplicaciones como [Wikiloc](#) <sup>[1]</sup>. También aprenderemos a combinar las fuentes de datos de Google con los mapas obtenidos mediante el servicio WMS, e incluso prescindiremos de los mapas de Google para combinar dos o más mapas WMS externos.

Aunque este módulo trata, exclusivamente, sobre el servicio OGC/WMS, las técnicas aquí explicadas son fácilmente trasladables a los demás servicios OGC que proporcionan datos en forma de imágenes.

En este módulo estudiaremos la Versión 3 de la API de Google Maps, la versión oficial de la API cuando se actualizaron por última vez estos materiales.

## Objetivos

Al final de este módulo, deberíamos ser capaces de:

1. Crear tipos de mapa personalizados en Google Maps.
2. Agregar nuevos tipos de mapa a Google Maps.
3. Llamar a un servicio OGC/WMS desde una aplicación web.
4. Crear mapas personalizados en GoogleMaps que incorporen datos procedentes de servicios WMS.
5. Combinar los mapas de Google Maps con mapas WMS externos.
6. Combinar dos o más fuentes WMS prescindiendo de los mapas de Google Maps.

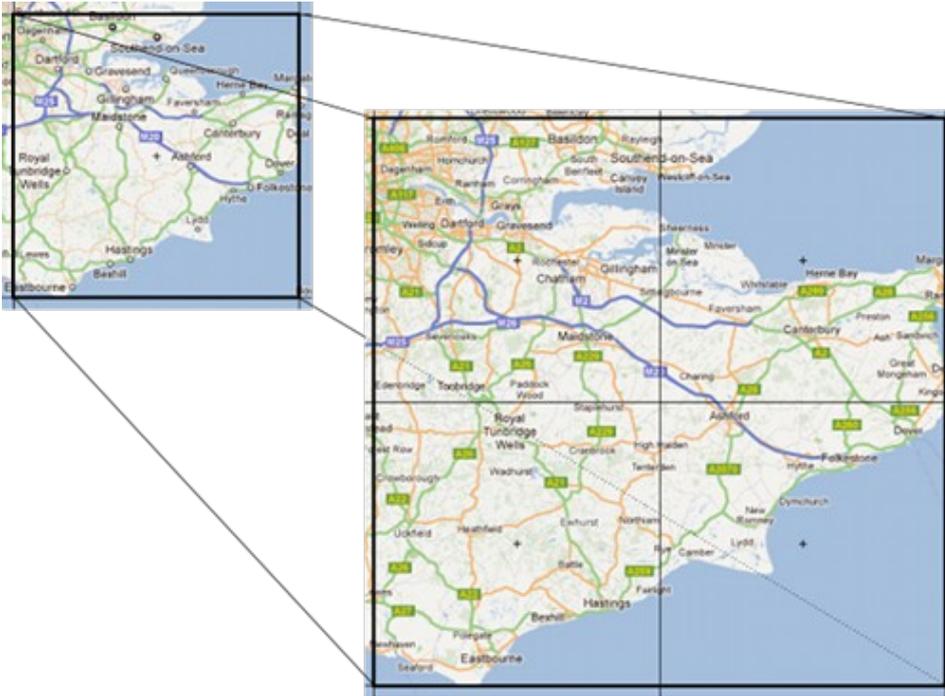
## Anatomía de un mapa: los tiles

Todos los mapas de Google Maps están divididos en pequeñas porciones rectangulares llamadas *tiles*. Cuando se solicita un mapa, Google Maps descarga del servidor las distintas porciones por separado y las junta, creando la ilusión de una única imagen. En el mapa siguiente, las líneas continuas muestran los límites de los *tiles*, y las cruces, los límites de los *tiles* en el siguiente nivel de zoom:

Observad que todas las porciones de un mapa tienen el mismo tamaño. Este tamaño puede variar de un mapa a otro, aunque lo más común es que los *tiles* sean cuadrados de 256 píxeles de lado — el valor por defecto.

El objetivo de reducir el mapa en porciones no es otro que minimizar el tamaño de los datos que el servidor tiene que enviar a cada usuario. Debemos pensar que si bien en niveles de zoom pequeños sería factible tener todo el mapa en una única imagen (cuyo tamaño sería de alrededor de unos pocos kilobytes), no sucede lo mismo con los niveles de zoom grandes, en los que sería prácticamente imposible enviar el mapa entero (cuyo tamaño sería de gigabytes).

En el nivel de zoom más pequeño (el nivel 0), el planeta entero está representado en una única porción. Cada nivel subsiguiente de zoom divide cada porción en cuatro subporciones, de manera que el número total de porciones en un nivel de zoom determinado responde a la fórmula  $4^N$ , donde  $N$  es el nivel de zoom. Así pues, en el nivel 1 habrá cuatro porciones (que formarán una rejilla de 2 x 2), en el nivel 2, dieciséis (que formarán una rejilla de 4 x 4), y así sucesivamente:



Dependiendo del tamaño del *viewport*, la coordenada central del mapa y el nivel de zoom, se mostrarán unas porciones u otras.

## Mapas personalizados

Aunque Google Maps dispone de una buena colección de mapas, su API nos permite definir mapas personalizados, es decir, mapas cuyo contenido (el contenido de cada una de las porciones) podemos elegir arbitrariamente.

Hay varias formas de crear un mapa personalizado pero probablemente la más cómoda sea utilizar la clase "google.maps.ImageMapType".

Un ejercicio simple consiste en crear un mapa con la misma imagen en todos los *tiles*. En el fragmento de código siguiente se instancia un tipo de mapa personalizado cuyo contenido es un mosaico de logotipos de la UOC:

```
function obtenerUrlTile(coordTile, zoom)
{
    return "logo-uoc.png";
}

var opcionesMapaPersonalizado = {
    getTileUrl: obtenerUrlTile,
    tileSize: new google.maps.Size(256, 256)
}

var mapaPersonalizado = new google.maps.ImageMapType(opcionesMapaPersonalizado);
```

Como podéis observar, las opciones que espera el constructor de la clase "ImageMapType" se especifican mediante el objeto "opcionesMapaPersonalizado", que contiene dos atributos:

- **getTileUrl**: Función que, dadas las coordenadas de un *tile* y el nivel de zoom del mapa, devuelve la dirección (URL) de la imagen que debe mostrarse en el *tile*. En el ejemplo, se especifica la función

"obtenerUrlTile()", que devuelve siempre la misma imagen: "logo-uoc.png".

- **tileSize**: Atributo que especifica el tamaño de las porciones del mapa. El tamaño se especifica mediante un objeto "google.maps.Size". En el ejemplo, se especifica un tamaño de 256 x 256 píxeles. El primer parámetro del constructor corresponde a la anchura de los *tiles*; El segundo, a la altura.

Con ello conseguiremos un resultado parecido al siguiente:



Hay que remarcar que el código precedente sólo crea un mapa personalizado pero no lo muestra. De hecho, dependiendo de como queramos mostrarlo, incluso deberemos especificar algunas opciones más. Un mapa personalizado puede mostrarse como:

- un de mapa base, o
- un mapa de superposición

Los **mapas base** son aquellos que podemos seleccionar mediante los controles situados en la parte superior derecha de un mapa: el mapa de carreteras, el mapa satélite, etc. Los **mapas de superposición** se muestran superpuestos a los mapas base. Normalmente se juega con transparencias para mostrar la información de los mapas base y de superposición a la vez. Podemos imaginarlos — los mapas de superposición — como una capa de información adicional que se muestra sobre los mapas base.

En los apartados siguientes aprenderemos a mostrar un mapa personalizado como un mapa base o como un mapa de superposición.

## Mapa base

Como ya hemos visto, un mapa base es un mapa independiente al que se puede acceder mediante los botones situados en la parte superior derecha del mapa.

Para mostrar un mapa personalizado como un mapa base hay que pasar algunas opciones más al constructor de la clase "ImageMapType":

- **maxZoom:** Especifica el nivel de zoom máximo del mapa.
- **minZoom:** Especifica el nivel de zoom mínimo del mapa. Este valor debe ser más grande que el valor de zoom máximo.
- **name:** Especifica el nombre del mapa base. Es el nombre que se mostrará en los controles del mapa.

Además, hay que indicarle a la API que queremos mostrar el nuevo mapa, y agregar manualmente los botones necesarios que nos permitirán cambiar de un mapa a otro.

El código siguiente muestra el mosaico de logotipos de la UOC como un mapa base:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Mosaico como mapa base</title>
<script type="text/javascript"
    src="http://maps.googleapis.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
function obtenerUrlTile(coordTile, zoom) {
    return "http://whereami.wikidot.com/local--files/uso-de-servicios-ogc/logo-uoc.png";
}

function InicializarMapa() {
    var opcionesControles;
    var opciones;
    var mapa;
    var opcionesMapaPersonalizado;
    var mapaPersonalizado;

    opcionesControles = {
        mapTypeIds: [ "MOSAICO_LOGO_UOC", google.maps.MapTypeId.ROADMAP, google.maps.MapTy
    };

    opciones = {
        center: new google.maps.LatLng(41.414829, 2.133003),
        zoom: 17,
        mapTypeId: "MOSAICO_LOGO_UOC",
        mapTypeControlOptions: opcionesControles
    };
    mapa = new google.maps.Map(document.getElementById("elemento_mapa"), opciones);

    // crea el mapa personalizado...
    opcionesMapaPersonalizado = {
        getTileUrl: obtenerUrlTile,
        tileSize: new google.maps.Size(256, 256),
        maxZoom: 20,
        minZoom: 0,
        name: "Mosaico"
    };
    mapaPersonalizado = new google.maps.ImageMapType(opcionesMapaPersonalizado);
```

## Uso\_de\_servicios\_OGC

```
// ... y lo registra como un mapa base
mapa.mapTypes.set("MOSAICO_LOGO_UOC", mapaPersonalizado);
}
</script>
</head>
<body onload="InicializarMapa()">
<div id="elemento_mapa" style="width: 500px; height: 300px"></div>
</body>
</html>
```

El resultado es el siguiente:

Como se puede observar, una vez creado el mapa personalizado, lo hemos "presentado" al objeto "Map" para que, cuando hiciésemos referencia a él, supiese de qué mapa estábamos hablando. Esto lo hemos hecho mediante una llamada a la función "set()" del atributo "mapTypes" del objeto "Map". Con esta llamada hemos asociado el identificador "MOSAICO\_LOGO\_UOC" al mapa personalizado:

```
mapa.mapTypes.set("MOSAICO_LOGO_UOC", mapaPersonalizado);
```

Una vez hecho esto, hemos utilizado el identificador para establecer — mediante el atributo "setMapTypeId" — el mapa personalizado como mapa predeterminado (aquél que se muestra de buen principio), y para enumerar — mediante el atributo "mapTypeControlOptions.mapTypeIds" — los mapas que se podrán seleccionar mediante los botones de cambio de mapa:

```
opcionesControles = {
    mapTypeIds: [ "MOSAICO_LOGO_UOC", google.maps.MapTypeId.ROADMAP, google.maps.MapTypeId.HYBRID ]
};

opciones = {
    /* (...) */
    mapTypeId: "MOSAICO_LOGO_UOC",
    mapTypeControlOptions: opcionesControles
};
```

Observad que los botones aparecen en el mismo orden que aparecen en el vector "mapTypeIds": primero el mapa personalizado, después el mapa de carreteras, y al final el mapa satélite.

## Mapa de superposición

Como ya hemos visto, los mapas de superposición se muestran superpuestos a los mapas base. De hecho, no es que se superpongan a un mapa base en particular, sino a todos ellos. Es decir, aunque cambiemos de mapa base se continuarán mostrando los mapas de superposición. Por esta razón, este tipo de mapa contiene información que complementa la ofrecida por los mapas base y que se puede sobreponer a ésta gracias al uso de transparencias.

Una vez creado el mapa personalizado, para superponerlo a los mapas base tan solo es necesario llamar a la función siguiente:

```
mapa.overlayMapTypes.insertAt(0, mapaPersonalizado);
```

El primer parámetro (en el ejemplo, "0") será la posición relativa del mapa respecto a los demás mapas de superposición. Si tenemos dos o más mapas de superposición podemos decidir en qué orden se tienen que dibujar. Cuanto menor sea este valor, más cerca estará el mapa del observador; Cuanto más grande, más cerca del mapa base.

## Uso\_de\_servicios\_OGC

El código siguiente muestra el logotipo de la UOC superpuesto al mapa a modo de marca de agua <sup>[2]</sup>:

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Mosaico como mapa base</title>
<script type="text/javascript"
        src="http://maps.googleapis.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
function obtenerUrlTile(coordTile, zoom) {
    return "http://whereami.wikidot.com/local--files/uso-de-servicios-ogc/logo-uoc.png";
}

function InicializarMapa() {
    var opciones;
    var mapa;
    var opcionesMapaPersonalizado;
    var mapaPersonalizado;

    opciones = {
        center: new google.maps.LatLng(-24.994167,134.86694),
        zoom: 3,
        mapTypeId: google.maps.MapTypeId.HYBRID
    };
    mapa = new google.maps.Map(document.getElementById("elemento_mapa"), opciones);

    // crea el mapa personalizado...
    opcionesMapaPersonalizado = {
        getTileUrl: obtenerUrlTile,
        tileSize: new google.maps.Size(256, 256),
        opacity: 0.5
    };
    mapaPersonalizado = new google.maps.ImageMapType(opcionesMapaPersonalizado);

    // ... y lo superpone a los mapas base
    mapa.overlayMapTypes.insertAt(0, mapaPersonalizado);
}
</script>
</head>
<body onload="InicializarMapa()">
<div id="elemento_mapa" style="width: 500px; height: 300px"></div>
</body>
</html>
```

El resultado es el siguiente:

Es importante observar que hemos pasado una nueva opción al constructor de la clase "ImageMapType": la opción "opacity":

```
opcionesMapaPersonalizado = {
    /* (...) */
    opacity: 0.5
};
```

Esta opción nos permite cambiar la opacidad del mapa personalizado, de manera que nos revele en mayor o menor medida el contenido de las capas inferiores (que pueden ser otros mapas de superposición o el mapa base). Un valor de 0,0 hace el mapa completamente transparente; 1,0, completamente opaco. En el ejemplo, hemos establecido la opacidad a 0,5.

## Mapas basados en fuentes WMS

Hasta ahora hemos aprendido a crear mapas personalizados y a mostrarlos como mapas base o mapas de superposición. Sin embargo, el contenido del mapa era muy poco interesante: se trataba tan sólo una imagen que se repetía una y otra vez.

Ahora utilizaremos todo lo que hemos aprendido para crear mapas personalizados que contengan cartografías obtenidas de un servicio OGC/WMS. La idea que reside en el fondo de esta técnica es relativamente simple: obtener la imagen de cada porción de un servicio OGC/WMS y proporcionarla a Google Maps.

### El servicio OGC/WMS

El servicio OGC/WMS, en adelante WMS (por *Web Map Service*, servicio de mapas de la Web), proporciona mapas georeferenciados a partir de datos geográficos. Los mapas se generan a petición de un cliente, que especifica, mediante un protocolo bien definido <sup>[3]</sup>, los datos que quiere obtener, la zona geográfica en la que se limita la petición, el formato del archivo de vuelta, etc. Al final, el cliente obtiene una imagen georeferenciada que contiene toda la información que esperaba obtener. Es importante destacar que el cliente no recibe nunca los datos en sí: tan sólo una imagen apta (hablando en términos de resolución) para ser mostrada en la pantalla de un ordenador.

El estándar define dos operaciones obligatorias que debe soportar el servicio WMS — *GetCapabilities* y *GetMap* — y otra optativa — *GetFeatureInfo*:

- **GetCapabilities:** Proporciona información acerca de los datos ofrecidos por el servidor y sus características, los sistemas de referencia soportados, el ámbito geográfico, etc.
- **GetMap:** Proporciona una imagen georeferenciada cuyo contenido viene determinado por los parámetros de la operación. Entre otras cosas, estos parámetros nos permiten definir el área del mapa, las capas mostradas, el formato de la imagen de vuelta, etc.
- **GetFeatureInfo:** Proporciona información acerca de características particulares mostradas en el mapa.

La información que se puede obtener por medio del servicio WMS gira en torno a las capas de información. Es decir, el proveedor proporciona una o más capas, que se pueden obtener aisladas o combinadas mediante la operación *GetMap*. Es importante remarcar que dicha operación siempre devuelve una única imagen como respuesta.

A continuación, se detallan los parámetros obligatorios que espera la operación *GetMap* y el significado de cada uno de ellos:

- **SERVICE=WMS (servicio):** Especifica el servicio al que se quiere acceder. En nuestro caso queremos acceder al servicio WMS.
- **VERSION=1.1.1:** Especifica el número de versión del protocolo. En otras palabras, especifica la versión del servicio a la que se espera acceder. Dependiendo de la versión, el proveedor aceptará unas operaciones u otras. Estos materiales se basan exclusivamente en la versión 1.1.1 del servicio.
- **REQUEST=GetMap (petición):** Especifica la operación que se quiere ejecutar. Para la operación *GetMap*, el valor de REQUEST siempre será "GetMap".
- **LAYERS (capas):** Especifica las capas de información que se desean obtener, separadas por comas. Las capas se deben listar desde la más profunda hasta la más superficial.
- **STYLES (estilos):** Especifica el estilo con el que se dibujará cada capa. Igual que en el parámetro anterior, los nombres de los estilos deben ir separados por comas.
- **SRS (de Source Reference System, "sistema de referencia de la fuente"):** Especifica el sistema de

referencia de la petición. Todas las coordenadas especificadas en los demás parámetros se suponen escritas en este sistema de referencia.

- **BBOX (de *Bounding BOX*, "cuadro delimitador")**: Especifica los límites del mapa que se desea obtener. Se trata de una lista separada por comas de las coordenadas suroeste (primero longitud, después latitud) y noreste del mapa. Las coordenadas deben especificarse de acuerdo al sistema de referencia indicado en el parámetro SRS.
- **WIDTH (anchura)**: Especifica la anchura, medida en píxeles, que debe tener la imagen resultante.
- **HEIGHT (altura)**: Especifica la altura, medida en píxeles, que debe tener la imagen resultante.
- **FORMAT (formato)**: Especifica el formato de archivo (PNG, JPEF, GIF, etc.) de la imagen resultante.

En resumen: mediante la operación *GetMap* podemos obtener una porción de cualquier mapa que nos ofrezca el proveedor, con el formato y las dimensiones que deseemos.

## Petición de datos

Toda la potencia y la flexibilidad que nos ofrece el servicio WMS queda ensalzada por otra característica no menos importante del servicio: su accesibilidad desde la Web. Como su nombre nos permite intuir (*Web Map Service*), cualquier cliente de la Web puede conectarse a un servicio WMS y obtener mapas.

En este sentido, el estándar WMS ha dispuesto que uno de los métodos de acceso a las operaciones WMS sea uno de los mecanismos fundamentales de la Web: el método HTTP GET. En otras palabras, se puede acceder a las operaciones mediante una URL cualquiera, de la misma forma que solicitamos una página web.

Así pues, si introducimos la URL siguiente en nuestro navegador:

<http://shagrati.icc.es/lizardtech/iserv/ows?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&LAYERS=mtc50m>

obtendremos un mapa topográfico a escala 1:50.000 de Benifallet y sus alrededores.

Si observamos detenidamente la URL, veremos los parámetros que hemos descrito en el apartado anterior separados por el carácter *et* (*&*, *ampersand* en inglés): REQUEST, SERVICE, VERSION, SRS, BBOX, WIDTH, HEIGHT, LAYERS, STYLES y FORMAT. La tabla siguiente muestra los valores asignados a cada uno de ellos y su significado:

Parámetro	Valor	Significado
SERVICE	WMS	Especifica que se desea acceder al servicio WMS.
VERSION	1.1.1	Especifica que la comunicación se basará en lo estipulado en la versión 1.1.1 del protocolo WMS.
REQUEST	GetMap	Especifica que se desea ejecutar la operación GetMap.
LAYERS	mtc50m	Especifica que se desea acceder a los datos de la capa ?mtc50m?. Las capas serán exclusivas de cada servidor, y se podrá obtener una lista de ellas mediante la operación "GetCapabilities". En este caso, se trata de un mapa topográfico a escala 1:50.000 ofrecido por el ICC <sup>[4]</sup> .
STYLES	<i>ninguno</i>	Especifica que no se desea ningún estilo en particular. De hecho, la capa seleccionada no tiene ningún estilo asociado. De nuevo, los estilos son propios de cada servidor y de cada capa. Se puede obtener una lista de estilos mediante la operación "GetCapabilities".
SRS	EPSG:23031	Especifica que las coordenadas que se utilicen en los demás parámetros estarán escritas según el sistema de referencia European Datum 1950

## Uso\_de\_servicios\_OGC

		proyección UTM Huso 31 Norte (EPSG:23031).
BBOX	290368.84, 4538236.42, 292203.28, 4540070.86	Especifica los límites del área que se desea obtener mediante las coordenadas de la esquinas suroeste (290368.84, 4538236.42) y noreste (292203.28, 4540070.86) del mapa. Recordemos que estas coordenadas están escritas en EPSG:23031.
WIDTH	500	Especifica que se desea obtener una imagen cuya anchura sea de 500 píxeles.
HEIGHT	500	Especifica que se desea obtener una imagen cuya altura sea de 500 píxeles.
FORMAT	JPEG	Especifica que se desea obtener una imagen en formato JPEG.

### Las coordenadas de un tile

Ahora que ya sabemos cómo obtener una porción de un mapa mediante un servicio WMS, veamos cómo podemos trasladar estas porciones a un mapa personalizado de Google Maps.

Como ya hemos visto, un mapa de Google Maps se divide en porciones rectangulares de igual tamaño. Además, el número de porciones varía en función del zoom, obedeciendo a la fórmula  $4^N$ , donde  $N$  es el nivel de zoom. Así, recordemos, en el nivel de zoom 0 (menor ampliación), tenemos una única porción; en el nivel 1, cuatro (formando una rejilla de 2 x 2); en el nivel 2, dieciséis (formando una rejilla de 4 x 4), y así sucesivamente.

Google Maps identifica las porciones según un par de coordenadas ( $x,y$ ), donde  $x$  es el número de columna dónde se encuentra la porción, e  $y$ , el número de fila. Las filas y las columnas se cuentan de derecha a izquierda y de arriba a abajo, empezando desde 0. Por ejemplo, el mapa siguiente muestra coordenadas de las porciones de un mapa en el nivel de zoom 2:

Cuando nos hemos introducido en la creación de mapas personalizados, hemos mencionado que la función "getTileUrl()" de la clase `?ImageMapType?` recibe por parámetro las coordenadas de la porción y el nivel de zoom. Aunque entonces no ha sido necesario consultar ninguno de sus parámetros porque siempre se mostraba la misma imagen en todas las porciones, ahora sí será necesario, puesto que el contenido de una porción u otra variará en función de la parte del mapa en la que se encuentre.

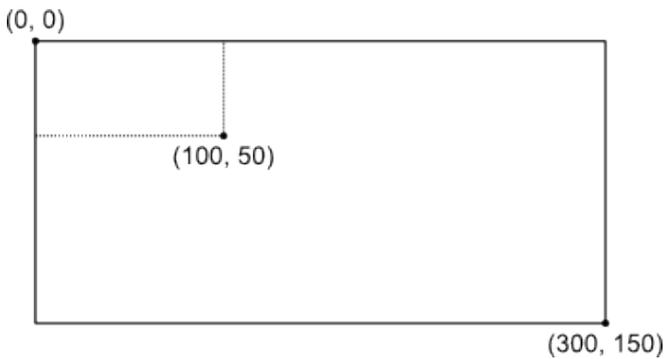
Para representar correctamente la parte del mapa correspondiente a una porción, debemos determinar el área que cubre dicha porción. Aquí ya empezamos a tener un punto de contacto con la función "GetMap" del servicio WMS, cuyo parámetro "BBOX" especificaba el área del mapa que se quería obtener mediante las coordenadas de las esquinas noreste y suroeste. Si somos capaces de obtener dichas coordenadas de una porción, podremos obtener la imagen del mapa por medio del servicio WMS.

Pero, ¿cómo obtenemos las coordenadas geográficas de las esquinas noreste y suroeste de una porción? La respuesta está en lo que Google Maps llama *coordenadas mundiales* y en un par de funciones que permiten convertir las *coordenadas mundiales* en coordenadas geográficas y viceversa.

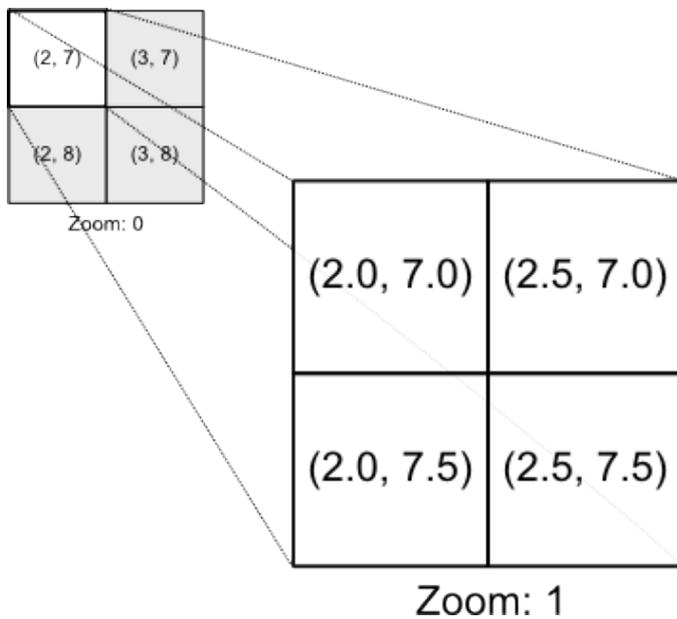
### Las coordenadas mundiales

Las *coodenadas mundiales* son un sistema de referencia propio de Google Maps que expresa una posición del mapa en función de un par de coordenadas ( $x,y$ ), dónde la  $x$  expresa la posición respecto la horizontal, e  $y$  respecto la vertical. La coordenada  $x$  puede tomar cualquier valor entre 0,0 y la anchura de las porciones del mapa; La  $y$ , entre 0,0 y la altura. Las coordenadas tienen su origen en el punto más al noroeste del mapa y crecen hacia el este y el sur.

Suponiendo un tamaño de las porciones de 300 x 150 píxeles, la figura siguiente muestra las *coordenadas mundiales* de diferentes puntos de un mapa en el nivel de zoom 0:



Pero, ¿qué sucede cuando aumentamos el nivel de zoom? Cada píxel del mapa — al igual que las porciones — se divide en cuatro sub-píxeles cada vez que aumenta el nivel de zoom. En consecuencia, las coordenadas de los nuevos píxeles estarán entre las coordenadas del píxel original y las de los píxeles adyacentes. La figura siguiente muestra las coordenadas que toman los nuevos píxeles cuando aumenta el nivel de zoom:



Podemos decir que las *coordenadas mundiales* expresan las coordenadas de un punto del mapa respecto al nivel de zoom 0, donde hay una sola porción y cuya anchura y la altura corresponden con la anchura y la altura del mapa.

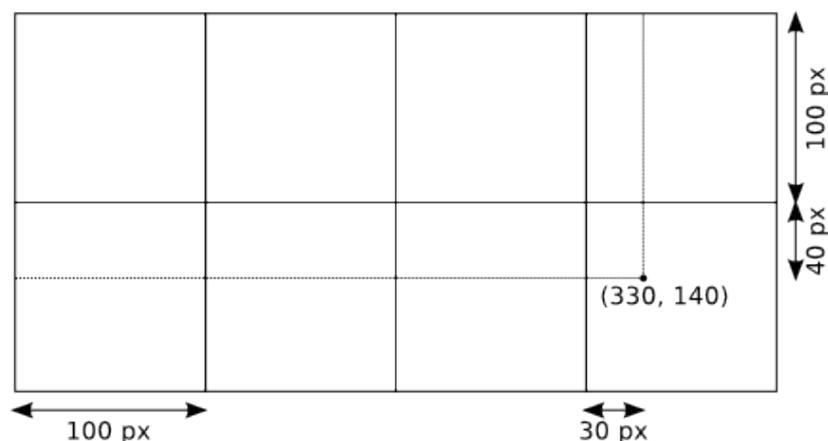
Una forma sencilla de calcular las *coordenadas mundiales* de un punto del mapa es determinar sus coordenadas en píxeles y dividir las entre  $2^{\text{zoom}}$ :

$$x_{\text{mundial}} = \frac{x_{\text{píxeles}}}{2^{\text{zoom}}}$$

$$y_{\text{mundial}} = \frac{y_{\text{píxeles}}}{2^{\text{zoom}}}$$

Sabemos que todos los *tiles* tienen las mismas dimensiones, y conocemos su altura y su anchura medidas en píxeles. Conocer las coordenadas en píxeles de un punto se reduce a determinar cuántos *tiles* le preceden, multiplicar esos *tiles* por su altura o su anchura (según sea el caso) para obtener el equivalente en píxeles, y

finalmente sumar los píxeles que le separan en vertical u horizontal el punto del origen de su *tile*. Por ejemplo, la figura siguiente muestra las coordenadas en píxeles de un punto de un mapa cuyos *tiles* miden 100 x 100 píxeles:



Aplicando esto a las fórmulas precedentes, tenemos que:

$$x_{mundial} = \frac{x_{tile} \cdot anchura_{tile} + x_{pixel}}{2^{zoom}}$$

$$y_{mundial} = \frac{y_{tile} \cdot altura_{tile} + y_{pixel}}{2^{zoom}}$$

Teniendo en cuenta que dentro de un *tile* la esquina suroeste se encuentra en las coordenadas  $(0, altura_{tile})$  tenemos que las *coordenadas mundiales* de la esquina suroeste serán las siguientes:

$$x_{mundial} = \frac{x_{tile} \cdot anchura_{tile}}{2^{zoom}}$$

$$y_{mundial} = \frac{y_{tile} \cdot altura_{tile} + altura_{tile}}{2^{zoom}}$$

Lo mismo es aplicable a la esquina noreste, que se encuentra en las coordenadas  $(anchura_{tile}, 0)$ . Sus *coordenadas mundiales* serán las siguientes:

$$x_{mundial} = \frac{x_{tile} \cdot anchura_{tile} + anchura_{tile}}{2^{zoom}}$$

$$y_{mundial} = \frac{y_{tile} \cdot altura_{tile}}{2^{zoom}}$$

## Transformación de coordenadas

Una vez obtenidas las *coordenadas mundiales* de las esquinas noroeste y suroeste, podemos obtener sus coordenadas geográficas través de la función "fromPointToLatLng()" de la clase "Projection". El objeto "Projection" se obtiene del mapa, mediante una llamada a la función "getProjection()".

La función "fromPointToLatLng()" espera un único parámetro, de tipo "google.maps.Point", que representa las *coordenadas mundiales* que se quieren transformar, y devuelve un objeto de tipo "google.maps.LatLng" que contiene las coordenadas geográficas equivalentes.

### El primer mapa WMS

Con todos los ingredientes en la mano, ya podemos "cocinar" nuestro primer mapa WMS.

El código siguiente crea un mapa personalizado que contiene un único mapa topográfico a escala 1:250.000 procedente del servicio OGC/WMS del ICC:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Google Maps y WMS</title>
<script type="text/javascript"
        src="http://maps.googleapis.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
// Parámetros:
//     x: número
//     y: número
//     zoom: número
//     mapa: google.maps.Map
function xyzALatLng(x, y, zoom, mapa) {
    var xyMax;
    var coordMundial;
    var latLng;

    // coordenada_mundial = coordenada_en_píxeles / 2^zoom
    xyMax = Math.pow(2, zoom);
    coordMundial = new google.maps.Point(x / xyMax, y / xyMax);

    // llamamos a la función que realiza la transformación de coordenadas
    // mundiales a latitud/longitud
    latLng = mapa.getProjection().fromPointToLatLng(coordMundial);

    return latLng;
}

// Parámetros:
//     coordTile: google.maps.Point
//     zoom: número
function obtenerUrlTile(coordTile, zoom) {
    var urlTile;
    var x0, y0, so, ne;

    // a partir de las coordenadas del "tile", calculamos las coordenadas
    // en píxeles de la esquina superior izquierda del "tile"
    x0 = coordTile.x * this.tileSize.width;
    y0 = coordTile.y * this.tileSize.height;

    // calculamos latitud y longitud de las coordenadas SO y NE del "tile"
    // estas coordenadas indican al servicio WMS/GetMap el área del mapa
    // que queremos obtener
    so = xyzALatLng(x0, y0 + this.tileSize.height, zoom, this.opcionesWms.mapa);
    ne = xyzALatLng(x0 + this.tileSize.width, y0, zoom, this.opcionesWms.mapa);

    // genera la URL que contiene la imagen del "tile"
```

## Uso\_de\_servicios\_OGC

```
// la imagen se obtiene llamando al servicio WMS/GetMap con los
// parámetros apropiados
urlTile = this.opcionesWms.url +
    "?SERVICE=WMS" +
    "&VERSION=1.1.1" +
    "&REQUEST=GetMap" +
    "&LAYERS=" + this.opcionesWms.capas +
    "&STYLES=" +
    "&SRS=EPSG:4326" +
    "&BBOX=" + so.lng() + "," + so.lat() + "," + ne.lng() + "," + ne.lat() +
    "&WIDTH=" + this.tileSize.width +
    "&HEIGHT=" + this.tileSize.height +
    "&FORMAT=" + this.opcionesWms.formatoImagen;

return urlTile;
}

function InicializarMapa() {
    var opciones, mapa;
    var opcionesMapaWms, mapaWms;
    var opcionesServicio;

    // establece las opciones y crea un nuevo mapa
    opciones = {
        center: new google.maps.LatLng(41.074167, 1.138889),
        zoom: 12,
        mapTypeId: google.maps.MapTypeId.ROADMAP,
        mapTypeControlOptions: {
            mapTypeIds: [google.maps.MapTypeId.ROADMAP, "WMS"]
        },
        mapTypeId: "WMS"
    }
    mapa = new google.maps.Map(document.getElementById("elemento_mapa"), opciones);

    // especifica las opciones del servicio WMS
    opcionesServicio = {
        url: "http://shagratt.icc.es/lizardtech/iserv/ows",
        capas: "mtc250m",
        formatoImagen: "jpeg",
        mapa: mapa
    }

    // crea el mapa WMS...
    opcionesMapaWms = {
        getTileUrl: obtenerUrlTile,
        tileSize: new google.maps.Size(256, 256),
        maxZoom: 15,
        minZoom: 6,
        name: "MTC250M",
        opcionesWms: opcionesServicio
    }
    mapaWms = new google.maps.ImageMapType(opcionesMapaWms);

    // ... y lo agrega como un tipo de mapa base
    mapa.mapTypes.set("WMS", mapaWms);
}
</script>
</head>
<body onload="InicializarMapa()">
<div id="elemento_mapa" style="width: 500px; height: 300px"></div>
</body>
</html>
```

El resultado es el siguiente:

Es importante observar que entre las opciones que se pasan al constructor de la clase "ImageMapType" hay una nueva opción llamada "opcionesWms" que nos permite especificar algunos parámetros relacionados con el servicio WMS:

```
opcionesServicio = {
    url: "http://shagrati.icc.es/lizardtech/iserv/ows",
    capas: "mtc250m",
    formatoImagen: "jpeg",
    mapa: mapa
}

opcionesMapaWms = {
    /* (...) */
    opcionesWms: opcionesServicio
}
```

El significado de cada una de las opciones se detalla a continuación:

- **url**: La URL base del servicio WMS al que se quiere acceder.
- **capas**: Lista separada por comas de las capas que se quieren obtener. Corresponde al parámetro "LAYERS" de la operación "GetMap" del servicio WMS.
- **formatoImagen**: Formato de la imagen que queremos que devuelva el servidor. Corresponde al parámetro "FORMAT" de la operación "GetMap" del servicio WMS.
- **mapa**: El objeto mapa en el que se muestra el mapa personalizado. A partir de él se obtiene el objeto "Projection" que nos permite transformar las *coordenadas mundiales* en coordenadas geográficas.

Todos estos parámetros los utiliza la función "obtenerUrlTile()" para construir la llamada al servicio WMS o, lo que es lo mismo, la URL de la imagen que se mostrará en el *tile*. Observad como esta función calcula las coordenadas en píxeles de las esquinas noreste y suroeste, y las pasa a la función "xyzALatLng()". Ésta última las transforma a *coordenadas mundiales* y calcula las coordenadas geográficas equivalentes.

La función "xyzALatLng()" utiliza la función "Math.pow()" (propia de JavaScript) para calcular el valor de  $2^{\text{zoom}}$ .

## Más mapas WMS

Partiendo del código desarrollado en el apartado anterior, a continuación se muestran algunos ejemplos que combinan los mapas de Google Maps con mapas WMS, o que incluso prescinden de los mapas de Google para mostrar un único mapa WMS o un par de ellos combinados. Para ilustrar estos ejemplos, hemos utilizado los servicios WMS que ofrecen el [ICC](#) <sup>[5]</sup>, el [IGC](#) <sup>[6]</sup> y el [CREAF](#) <sup>[7]</sup>.

### Mapa WMS superpuesto a los mapas de Google Maps

En este ejemplo, se ha creado un mapa personalizado que combina los mapas de Google Maps con un mapa geológico a escala 1:50.000 procedente del servicio WMS del IGC:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Google Maps y WMS</title>
```

## Uso\_de\_servicios\_OGC

```
<script type="text/javascript"
    src="http://maps.googleapis.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
// Parámetros:
//     x: número
//     y: número
//     zoom: número
//     mapa: google.maps.Map
function xyzALatLng(x, y, zoom, mapa) {
    var xyMax;
    var coordMundial;
    var latLng;

    // coordenada_mundial = coordenada_en_píxeles / 2^zoom
    xyMax = Math.pow(2, zoom);
    coordMundial = new google.maps.Point(x / xyMax, y / xyMax);

    // llamamos a la función que realiza la transformación de coordenadas
    // mundiales a latitud/longitud
    latLng = mapa.getProjection().fromPointToLatLng(coordMundial);

    return latLng;
}

// Parámetros:
//     coordTile: google.maps.Point
//     zoom: número
function obtenerUrlTile(coordTile, zoom) {
    var urlTile;
    var x0, y0, so, ne;

    // a partir de las coordenadas del "tile", calculamos las coordenadas
    // en píxeles de la esquina superior izquierda del "tile"
    x0 = coordTile.x * this.tileSize.width;
    y0 = coordTile.y * this.tileSize.height;

    // calculamos latitud y longitud de las coordenadas SO y NE del "tile"
    // estas coordenadas indican al servicio WMS/GetMap el área del mapa
    // que queremos obtener
    so = xyzALatLng(x0, y0 + this.tileSize.height, zoom, this.opcionesWms.mapa);
    ne = xyzALatLng(x0 + this.tileSize.width, y0, zoom, this.opcionesWms.mapa);

    // genera la URL que contiene la imagen del "tile"
    // la imagen se obtiene llamando al servicio WMS/GetMap con los
    // parámetros apropiados
    urlTile = this.opcionesWms.url +
        "?SERVICE=WMS" +
        "&VERSION=1.1.1" +
        "&REQUEST=GetMap" +
        "&LAYERS=" + this.opcionesWms.capas +
        "&STYLES=" +
        "&SRS=EPSG:4326" +
        "&BBOX=" + so.lng() + "," + so.lat() + "," + ne.lng() + "," + ne.lat() +
        "&WIDTH=" + this.tileSize.width +
        "&HEIGHT=" + this.tileSize.height +
        "&FORMAT=" + this.opcionesWms.formatoImagen;

    return urlTile;
}

function InicializarMapa() {
    var opciones, mapa;
    var opcionesMapaWms, mapaWms;
}
```

## Uso\_de\_servicios\_OGC

```
var opcionesServicio;

// establece las opciones y crea un nuevo mapa
opciones = {
    center: new google.maps.LatLng(41.2975, 1.028333),
    zoom: 13,
    mapTypeId: google.maps.MapTypeId.ROADMAP
}
mapa = new google.maps.Map(document.getElementById("elemento_mapa"), opciones);

// especifica las opciones del servicio WMS
opcionesServicio = {
    url: "http://siurana.igc.cat/arcgis/services/geologia/BG50mv1r02/MapServer/WMSServ",
    capas: "0,1,2,3",
    formatoImagen: "image/jpeg",
    mapa: mapa
}

// crea el mapa WMS...
opcionesMapaWms = {
    getTileUrl: obtenerUrlTile,
    tileSize: new google.maps.Size(256, 256),
    opacity: 0.2,
    opcionesWms: opcionesServicio
}
mapaWms = new google.maps.ImageMapType(opcionesMapaWms);

// ... y lo superpone a los mapas base
mapa.overlayMapTypes.insertAt(0, mapaWms);

}
</script>
</head>
<body onload="InicializarMapa()">
<div id="elemento_mapa" style="width: 500px; height: 300px"></div>
</body>
</html>
```

El resultado es el siguiente:

### Dos mapas WMS procedentes de una única fuente combinados

El ejemplo siguiente combina dos mapas procedentes del servicio WMS del ICC: la ortofoto del Vuelo americano (1956-57) serie B 1:5.000 (mapa base) y el topográfico 1:50.000 (mapa superpuesto):

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Google Maps y WMS</title>
<script type="text/javascript"
    src="http://maps.googleapis.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
// Parámetros:
//     x: número
//     y: número
//     zoom: número
//     mapa: google.maps.Map
function xyzALatLng(x, y, zoom, mapa) {
    var xyMax;
```

## Uso\_de\_servicios\_OGC

```
var coordMundial;
var latLng;

// coordenada_mundial = coordenada_en_píxeles / 2^zoom
xyMax = Math.pow(2, zoom);
coordMundial = new google.maps.Point(x / xyMax, y / xyMax);

// llamamos a la función que realiza la transformación de coordenadas
// mundiales a latitud/longitud
latLng = mapa.getProjection().fromPointToLatLng(coordMundial);

return latLng;
}

// Parámetros:
// coordTile: google.maps.Point
// zoom: número
function obtenerUrlTile(coordTile, zoom) {
    var urlTile;
    var x0, y0, so, ne;

    // a partir de las coordenadas del "tile", calculamos las coordenadas
    // en píxeles de la esquina superior izquierda del "tile"
    x0 = coordTile.x * this.tileSize.width;
    y0 = coordTile.y * this.tileSize.height;

    // calculamos latitud y longitud de las coordenadas SO y NE del "tile"
    // estas coordenadas indican al servicio WMS/GetMap el área del mapa
    // que queremos obtener
    so = xyzALatLng(x0, y0 + this.tileSize.height, zoom, this.opcionesWms.mapa);
    ne = xyzALatLng(x0 + this.tileSize.width, y0, zoom, this.opcionesWms.mapa);

    // genera la URL que contiene la imagen del "tile"
    // la imagen se obtiene llamando al servicio WMS/GetMap con los
    // parámetros apropiados
    urlTile = this.opcionesWms.url +
        "?SERVICE=WMS" +
        "&VERSION=1.1.1" +
        "&REQUEST=GetMap" +
        "&LAYERS=" + this.opcionesWms.capas +
        "&STYLES=" +
        "&SRS=EPSG:4326" +
        "&BBOX=" + so.lng() + "," + so.lat() + "," + ne.lng() + "," + ne.lat() +
        "&WIDTH=" + this.tileSize.width +
        "&HEIGHT=" + this.tileSize.height +
        "&FORMAT=" + this.opcionesWms.formatoImagen;

    return urlTile;
}

function InicializarMapa() {
    var opciones, mapa;
    var opcionesMapaTopo, mapaTopo;
    var opcionesServicioTopo;
    var opcionesMapaGeo, mapaGeo;
    var opcionesServicioGeo;

    // establece las opciones y crea un nuevo mapa
    opciones = {
        center: new google.maps.LatLng(40.718639, 0.865859),
        zoom: 13,
        mapTypeId: google.maps.MapTypeId.ROADMAP,
        mapTypeControlOptions: {
            mapTypeIds: [google.maps.MapTypeId.SATELLITE, "WMS"]
        }
    }
}
```

## Uso\_de\_servicios\_OGC

```
    },
    mapTypeId: "WMS"
}
mapa = new google.maps.Map(document.getElementById("elemento_mapa"), opciones);

// especifica las opciones del servicio WMS
opcionesServicioTopo = {
    url: "http://shagrat.icc.es/lizardtech/iserv/ows",
    capas: "ovab5m",
    formatoImagen: "jpeg",
    mapa: mapa
}

// crea el mapa WMS...
opcionesMapaTopo = {
    getTileUrl: obtenerUrlTile,
    tileSize: new google.maps.Size(256, 256),
    maxZoom: 15,
    minZoom: 6,
    name: "OVAB5M",
    opcionesWms: opcionesServicioTopo
}
mapaTopo = new google.maps.ImageMapType(opcionesMapaTopo);

// ... y lo agrega como un tipo de mapa base
mapa.mapTypes.set("WMS", mapaTopo);

// especifica las opciones del servicio WMS
opcionesServicioGeo = {
    url: "http://shagrat.icc.es/lizardtech/iserv/ows",
    capas: "mtc50m",
    formatoImagen: "jpeg",
    mapa: mapa
}

// crea el mapa WMS...
opcionesMapaGeo = {
    getTileUrl: obtenerUrlTile,
    tileSize: new google.maps.Size(256, 256),
    opacity: 0.5,
    opcionesWms: opcionesServicioGeo
}
mapaGeo = new google.maps.ImageMapType(opcionesMapaGeo);

// ... y lo superpone a los mapas base
mapa.overlayMapTypes.insertAt(0, mapaGeo);
}
</script>
</head>
<body onload="InicializarMapa()">
<div id="elemento_mapa" style="width: 500px; height: 300px"></div>
</body>
</html>
```

El resultado es el siguiente:

### Dos mapas WMS procedentes de fuentes distintas combinados

El ejemplo siguiente combina el mapa topográfico de Cataluña a escala 1:250.000 procedente del servicio WMS del ICC (mapa base) con el atlas climático de Cataluña (clima anual, temperatura media) procedente del servicio WMS del CREAM.

## Uso\_de\_servicios\_OGC

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Google Maps y WMS</title>
<script type="text/javascript"
  src="http://maps.googleapis.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
// Parámetros:
//   x: número
//   y: número
//   zoom: número
//   mapa: google.maps.Map
function xyzALatLng(x, y, zoom, mapa) {
  var xyMax;
  var coordMundial;
  var latLng;

  // coordenada_mundial = coordenada_en_píxeles / 2^zoom
  xyMax = Math.pow(2, zoom);
  coordMundial = new google.maps.Point(x / xyMax, y / xyMax);

  // llamamos a la función que realiza la transformación de coordenadas
  // mundiales a latitud/longitud
  latLng = mapa.getProjection().fromPointToLatLng(coordMundial);

  return latLng;
}

// Parámetros:
//   coordTile: google.maps.Point
//   zoom: número
function obtenerUrlTile(coordTile, zoom) {
  var urlTile;
  var x0, y0, so, ne;

  // a partir de las coordenadas del "tile", calculamos las coordenadas
  // en píxeles de la esquina superior izquierda del "tile"
  x0 = coordTile.x * this.tileSize.width;
  y0 = coordTile.y * this.tileSize.height;

  // calculamos latitud y longitud de las coordenadas SO y NE del "tile"
  // estas coordenadas indican al servicio WMS/GetMap el área del mapa
  // que queremos obtener
  so = xyzALatLng(x0, y0 + this.tileSize.height, zoom, this.opcionesWms.mapa);
  ne = xyzALatLng(x0 + this.tileSize.width, y0, zoom, this.opcionesWms.mapa);

  // genera la URL que contiene la imagen del "tile"
  // la imagen se obtiene llamando al servicio WMS/GetMap con los
  // parámetros apropiados
  urlTile = this.opcionesWms.url +
    "?SERVICE=WMS" +
    "&VERSION=1.1.1" +
    "&REQUEST=GetMap" +
    "&LAYERS=" + this.opcionesWms.capas +
    "&STYLES=" + this.opcionesWms.estilos +
    "&SRS=EPSG:4326" +
    "&BBOX=" + so.lng() + "," + so.lat() + "," + ne.lng() + "," + ne.lat() +
    "&WIDTH=" + this.tileSize.width +
    "&HEIGHT=" + this.tileSize.height +
    "&FORMAT=" + this.opcionesWms.formatoImagen;

  return urlTile;
}
```

## Uso\_de\_servicios\_OGC

```
}  
  
function InicializarMapa() {  
    var opciones, mapa;  
    var opcionesMapaTopo, mapaTopo;  
    var opcionesServicioTopo;  
    var opcionesMapaTemp, mapaTemp;  
    var opcionesServicioTemp;  
  
    // establece las opciones y crea un nuevo mapa  
    opciones = {  
        center: new google.maps.LatLng(42.706928, 0.902322),  
        zoom: 11,  
        mapTypeId: google.maps.MapTypeId.ROADMAP,  
        mapTypeControlOptions: {  
            mapTypeIds: [google.maps.MapTypeId.ROADMAP, "WMS"]  
        },  
        mapTypeId: "WMS"  
    }  
    mapa = new google.maps.Map(document.getElementById("elemento_mapa"), opciones);  
  
    // especifica las opciones del servicio WMS  
    opcionesServicioTopo = {  
        url: "http://shagratt.icc.es/lizardtech/iserv/ows",  
        capas: "mtc250m",  
        formatoImagen: "jpeg",  
        mapa: mapa  
    }  
  
    // crea el mapa WMS...  
    opcionesMapaTopo = {  
        getTileUrl: obtenerUrlTile,  
        tileSize: new google.maps.Size(256, 256),  
        maxZoom: 15,  
        minZoom: 6,  
        name: "MTC250M",  
        opcionesWms: opcionesServicioTopo  
    }  
    mapaTopo = new google.maps.ImageMapType(opcionesMapaTopo);  
  
    // ... y lo agrega como un tipo de mapa base  
    mapa.mapTypes.set("WMS", mapaTopo);  
  
    // especifica las opciones del servicio WMS  
    opcionesServicioTemp = {  
        url: "http://www.opengis.uab.es/cgi-bin/ACDC/MiraMon5_0.cgi",  
        capas: "clima_anual-catalunya",  
        estilos: "Tmit",  
        formatoImagen: "jpeg",  
        mapa: mapa  
    }  
  
    // crea el mapa WMS...  
    opcionesMapaTemp = {  
        getTileUrl: obtenerUrlTile,  
        tileSize: new google.maps.Size(256, 256),  
        opacity: 0.3,  
        opcionesWms: opcionesServicioTemp  
    }  
    mapaTemp = new google.maps.ImageMapType(opcionesMapaTemp);  
  
    // ... y lo superpone a los mapas base  
    mapa.overlayMapTypes.insertAt(0, mapaTemp);  
}
```

```
</script>
</head>
<body onload="InicializarMapa()">
<div id="elemento_mapa" style="width: 500px; height: 300px"></div>
</body>
</html>
```

El resultado es el siguiente:

## Resumen

En este módulo hemos aprendido a ir más allá de las capacidades de Google Maps, conectándolo con bases de datos de mapas externas mediante el protocolo OGC/WMS. Esta forma de trabajar nos da acceso a una gran variedad de cartografía temática (muchas veces de carácter local) mucho más precisa que la ofrecida por Google, a la vez que permite el acceso a unos mismos datos a través de diversas fuentes, aumentando así la disponibilidad y la confiabilidad de los datos.

Para poder utilizar mapas procedentes de una fuente WMS en Google Maps, hemos aprendido, en primer lugar, a obtener mapas WMS a través de un navegador web, de la misma forma que accederíamos a cualquier sitio de la red. Una vez hecho esto, para poder visualizarlos, ha sido necesario crear mapas personalizados que posteriormente hemos agregado como una capa en alguno de los tipos de mapa predeterminados de Google Maps (mapa, satélite o relieve), o en uno de nuevo.

Al final, hemos aprendido a combinar mapas de Google Maps, con mapas WMS, e incluso mapas WMS procedentes de diferentes fuentes.

## Notas

1. ? Wikiloc es una aplicación web basada en Google Maps que permite compartir rutas georreferenciadas. Entre los tipos de mapa disponibles, incluye fuentes WMS.
2. ? Una marca de agua es una imagen translúcida que se muestra sobre otra imagen o texto y que sirve para identificar el origen del documento.
3. ? Se puede consultar la especificación del protocolo en:  
<http://www.opengeospatial.org/standards/wms>.
4. ? Institut Cartogràfic de Catalunya - <http://www.icc.cat>.
5. ? Los datos proporcionados por el servicio WMS del ICC se detallan en la página:  
<http://www.icc.cat/esl/Home-ICC/Geoinformacion-digital/Servicios-en-linea-Geoservicios/Para-usuarios-SIG>.
6. ? Los datos proporcionados por el servicio WMS del IGC se detallan en la página:  
<http://www.igc.cat/web/es/geoserveis.php>.
7. ? Los datos proporcionados por el servicio WMS del CREA se detallan en la página:  
<http://www.opengis.uab.es>.