

## Vissir2: Evolución del visor web del ICC hacia tecnologías abiertas.

O. Fonts<sup>(1)</sup>, A. Vidal<sup>(2)</sup>

- (1) Centro de Visualización Interactiva (CeVI), Universitat Jaume I (UJI), Castellón.  
fonts@sg.uji.es.
- (2) Unidad de Desarrollo Informático, Institut Cartogràfic de Catalunya (ICC).  
antoni.vidal@icc.cat

### RESUMEN

*En la web del Institut Cartogràfic de Catalunya se puede navegar por la cartografía oficial de Catalunya, así como descargarla en diferentes formatos, gracias a la aplicación Vissir2, evolución de la histórica Visir (VISor del Servidor de Imágenes Ráster).*

*Nos propusimos que esta nueva versión superase ciertas limitaciones de la herramienta existente: debía poder evolucionar fácilmente en el tiempo, y acercar el uso de nuestros datos y servicios al público más amplio posible. Por ello, se optó por una aplicación basada en software libre y orientada a servicios web con protocolos estándar.*

*En la parte de cliente se ejecuta OpenLayers, una aplicación Javascript muy completa y fácilmente adaptable. En la parte del servidor, TileCache sirve los datos cartográficos rápidamente y siempre actualizados gracias a una adaptación propia. Pero Vissir2 también utiliza servicios web de producción propia para la impresión, la localización de topónimos, la conversión de coordenadas o la consulta de productos descargables, todos ellos con un protocolo ligero y sencillo.*

*Gracias a esta aplicación los usuarios pueden hacer búsquedas predictivas de topónimos, enlazar a una vista determinada, incrustar un pequeño mapa en su web, descargar un jpeg con la vista actual –o imprimirla–, descargar los datos en diversos formatos, consultar la leyenda del mapa, medir longitudes y áreas, transformar coordenadas, consultar la fecha de elaboración de un producto determinado, o superponer capas con transparencia.*

**Palabras clave:** Web Map, OpenLayers, TileCache, SOA.

## INTRODUCCIÓN

### Antecedentes

El Institut Cartogràfic de Catalunya (en adelante ICC) fue pionero en ofrecer la descarga de sus productos de forma gratuita a través de la red, con la aplicación Catalunya Full a Full (CFAF), hace ya más de ocho años. Con la incorporación de los mapas continuos a finales de 2004 se puso en marcha la aplicación Vissir, que permitía navegar a través de todo el territorio catalán sobre los mapas topográficos, ortofotografías y mapas geológicos a diferentes escalas. Vissir incorporó también la descarga de productos, dejando CFAF obsoleto.

Paralelamente, se fueron desarrollando una serie de visores web para la consulta de datos específicos: mapas de progreso, catálogo de vuelos, o señales geodésicas. Mientras estos últimos visores se basaron en tecnología de ESRI (arcIMS), Vissir se desarrolló sobre una API de Lizardtech. Los visores web dependían de la tecnología de cada fabricante y las tecnologías eran difícilmente integrables entre sí.

Más tarde nacieron los geoservicios, que ofrecían todas las capas de información mencionadas mediante el protocolo WMS de OGC, con lo que pudieron consultarse conjuntamente desde diferentes aplicaciones remotas.

En 2006 surge la necesidad de elaborar visores bajo un nuevo paradigma que permitiera superar las limitaciones existentes, mediante el uso de estándares OGC y la modularidad y reutilización de sus componentes. Bajo este nuevo paradigma nace Vissir2.

### Objetivos

Vissir2 es la aplicación de consulta y descarga de la cartografía oficial del ICC. Permite el acceso a los datos de forma rápida, a pantalla completa y dispone de herramientas aptas para un uso tanto general como profesional. Además se han tenido en cuenta criterios de usabilidad y se ha mantenido la interfaz lo más sencilla posible.

El objetivo es maximizar la accesibilidad a la cartografía del ICC a través de la red, teniendo en cuenta sus diversos usos: consulta en pantalla, enlace a una vista, descarga de datos, impresión, envío por mail, e incluso la incrustación de un pequeño visor interactivo en otras páginas.

Vissir2 debía ser capaz de evolucionar con el tiempo, y sus componentes debían ser reutilizables en otras aplicaciones. Esto nos ha llevado a establecer una arquitectura basada en una colección de servicios web y de componentes javascript que los gestionan. Otro requisito importante es mantener su independencia respecto a un fabricante o tecnología de servidor específico, o, dicho de otra manera, garantizar su interoperabilidad. Con lo que los servicios implementan formatos y protocolos abiertos y de uso común en la web.

Así, Vissir2 es una aplicación que se ejecuta en el navegador de cada usuario, proporcionando un interfaz interactivo. Pero Vissir2 también es la colección de servicios que responden con los datos que les son requeridos. Al tratarse de servicios independientes, si algún servicio falla en un momento dado, el resto de la aplicación sigue funcionando.

Se han utilizado para su construcción TileCache y OpenLayers, productos de calidad que cumplen con la filosofía adoptada y dan respuesta a la mayoría de requisitos. Al tratarse de software libre, se han podido adaptar siempre que ha sido necesario.

En los siguientes apartados se describe la interfaz de usuario, las adaptaciones y mejoras incluidas en el cliente, el diseño de la caché de datos, los servicios web utilizados, y la capacidad de integrar Vissir2 con otras aplicaciones.

## INTERFAZ DE USUARIO

La interfaz del Vissir2 destaca por su diseño líquido que permite adaptar la vista de la aplicación a cualquier tamaño de ventana o pantalla, y por la distribución de los elementos gráficos según los cánones habituales en los visores de mapas, lo que posibilita un aprendizaje rápido por parte del usuario.

La interfaz se divide en dos zonas principales: el mapa y los controles.

### El Mapa

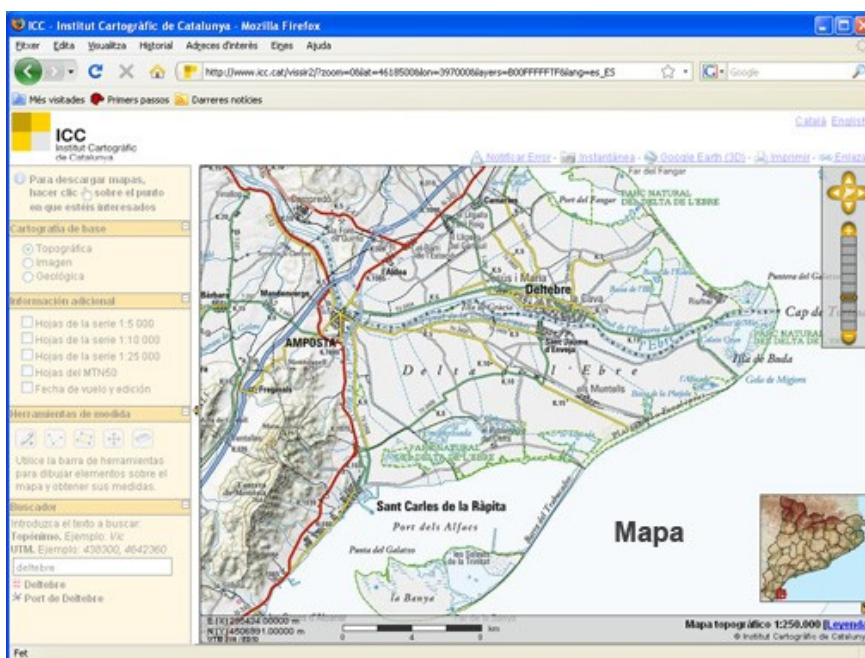


Figura 1. Vista de la aplicación Vissir2.

La zona del mapa destaca por su amplitud y la opción de visualizarla a pantalla casi completa. La navegación por el mapa puede hacerse con el ratón, utilizando la función de arrastrar y soltar para el desplazamiento y la rueda central para el cambio de escala, lo que se ha convertido en un estándar de facto en las aplicaciones de mapas. Este uso facilita enormemente la navegación.

Todas las funcionalidades se han desarrollado teniendo en cuenta la facilidad, procurando mantener un interfaz no más simple e intuitivo posible. La función principal de descarga de productos se ha resuelto permitiendo al usuario que haga clic con el ratón sobre el punto del mapa del cual desea obtener cartografía.

El clic del ratón abre una ventana con la lista de los productos disponibles para su descarga. Además de la descripción de las hojas y los enlaces a distintos formatos de descarga (MrSID, DGN, EXP, DXF, Miramón), se puede descargar una imagen JPEG georreferenciada de la vista actual y consultar las coordenadas del punto seleccionado en los dos datum oficiales.

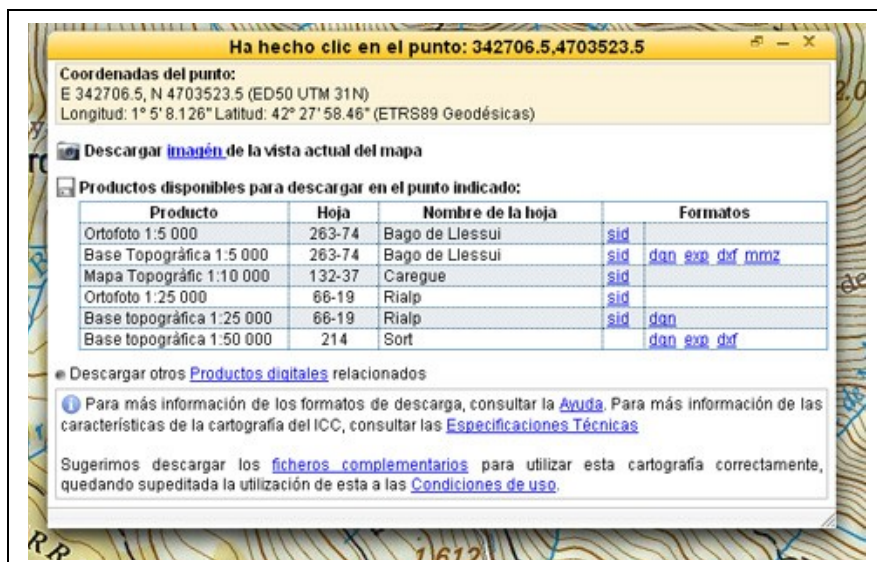


Figura 2. Ventana de productos descargables

Las ventanas abiertas por el usuario, tanto la de descarga de productos como la de validación de usuario, enlazar, etc., son ventanas construidas en javascript<sup>1</sup>, con lo que se evita el problema del bloqueo de *pop-ups* de los navegadores.

## Controles

La zona de los controles se halla alrededor del mapa y se organiza por bloques. En la parte izquierda se encuentran los bloques de elementos relacionados con el mapa. En la parte superior se encuentran los enlaces relacionados con funcionalidad de la aplicación: cambio de idioma, impresión, enlazar, apertura de kml, función snapshot y notificación de errores.

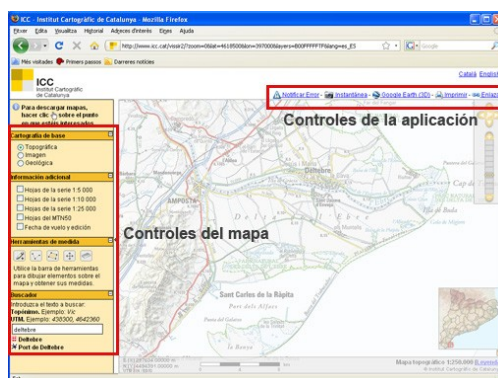


Figura 3. Disposición de los controles

<sup>1</sup> <http://prototype-window.xilinus.com/>

### III Jornadas de SIG Libre

El primer bloque de control del mapa es la selección de la capa base con *radio-button*, puesto que sólo se puede tener una capa de base activa: topográfica, ortoimagen o mapa geológico. En caso de seleccionar ortoimagen o mapa geológico, un control de transparencia permite superponer la topografía de forma translúcida.

El siguiente bloque se seleccionan las capas de información adicional que se superponen a la capa de base. Se constituye de diferentes *check-box* que el usuario puede seleccionar a su voluntad.

Otro bloque agrupa las herramientas de dibujo sobre el mapa. Estas permiten la creación de puntos, líneas y polígonos así como su modificación o eliminación. Según la figura representada se ofrecen las coordenadas, o cálculos de distancia, perímetro o área.

El último bloque permite la búsqueda por topónimos al estilo *suggest* para guiar al usuario en la búsqueda del topónimo que le interesa. Según se introducen caracteres en el campo de texto, aparece una lista de candidatos ordenados por similitud. Asimismo puede optarse por introducir un par de coordenadas en formato UTM.

En la parte superior del visor se encuentran las distintas herramientas de la aplicación, donde se irán añadiendo nuevas funcionalidades.

Para comprobar la usabilidad de la aplicación se han realizado diversos tests de usuario. Asimismo el análisis de las métricas de Google Analytics<sup>2</sup> y de otras herramientas como Click Heat<sup>3</sup> permiten continuar investigando en este sentido.

## EL CLIENTE JAVASCRIPT

### OpenLayers y Prototype

Para el desarrollo de visores web de mapas, la caja de herramientas por excelencia es OpenLayers<sup>4</sup>. Permite construir un visor tipo *slippy map* (mapa desplazable) a partir de orígenes de datos muy diversos que pueden combinarse entre sí: Imágenes locales, servicios WMS, de teselas, e incluso los datos de Google Maps o Yahoo. Además de imágenes ráster, se puede incluir información vectorial en diversos formatos: GML (y WFS), KML, geoRSS, geoJSON, etc. A los datos vectoriales se les puede aplicar estilos o comportamientos, que van desde la identificación de una entidad hasta su edición en línea.

El código de OpenLayers es 100% javascript, y es orientado a objetos. Esto hace que el código y los datos estén encapsulados en distintas clases, y que se pueda utilizar mecanismos de herencia para extenderlas o adaptarlas en caso necesario.

---

<sup>2</sup> <http://www.google.com/analytics>

<sup>3</sup> <http://www.labsmedia.com/clickheat/index.html>

<sup>4</sup> Términos de la licencia en: <http://svn.openlayers.org/trunk/openlayers/license.txt>

A pesar de ser un producto muy potente, existe cierta barrera de entrada que hay que vencer. No hay un 'libro oficial' o un tutorial paso a paso que sirvan de referencia. Existe documentación sobre la API, pero en muchos casos resulta escasa. Para iniciarse en el uso de OpenLayers lo mejor es recurrir a la colección de ejemplos disponible en la web del proyecto. El estudio del código fuente de estos ejemplos es suficiente para comprender y utilizar las funciones principales. En caso de tener que desarrollar alguna funcionalidad nueva, se recomienda estudiar directamente el código fuente, heredar de alguna clase existente, e intentar seguir los mismos criterios de diseño. En última instancia, existen dos listas de correo (para usuarios y desarrolladores) donde resolver dudas.

Prototype<sup>5</sup> es un framework que aumenta la potencia de Javascript. OpenLayers ha heredado varios de sus criterios. Así, conocer Prototype ayuda a comprender cómo funciona OpenLayers.

Vissir2 es una adaptación de OpenLayers y hace uso de Prototype para algunas de sus funciones.

### **Multiidioma**

Vissir2 proporciona un interfaz plurilingüe. Su idioma original es el catalán, pero la aplicación puede autotraducirse al castellano y al inglés mediante una lista de textos equivalentes y un módulo de javascript que, al ser invocado, recorre el documento y sustituye cada texto por su equivalente. Este módulo depende de Prototype pero no de OpenLayers. Puede incluirse en cualquier aplicación web que deba ser localizada sin necesidad de PHP, Java u otras tecnologías de contenido dinámico en el servidor.

### **Gestión de capas**

Para el control de visualización de las diferentes capas cartográficas se ha modificado ligeramente la clase OpenLayers.Control.LayerSwitcher. La adaptación consiste en permitir la superposición semitransparente de dos capas de base.

Existen tres capas de base, correspondientes a las tres pirámides de tiles generadas (ver sección sobre tilecaché). A ellas se puede superponer información adicional sobre la distribución de las hojas 1:5 000, 1:10 000, 1:25 000 y 1:50 000, además de poder superponer un corte con indicaciones sobre la fecha de vuelo y edición de los productos mostrados.

### **Herramientas de medida**

Barra de herramientas basada en las clases OpenLayers.Control.EditingToolbar y OpenLayers.Control.Measure<sup>6</sup>. Permite dibujar sobre el mapa puntos (y consultar sus coordenadas), líneas (y consultar su longitud) y polígonos (y consultar su área y perímetro). Los elementos añadidos pueden editarse y borrarse a voluntad.

---

<sup>5</sup> Licencia MIT.

<sup>6</sup> Adaptación de un control que no pertenece al código base de OpenLayers, <http://trac.openlayers.org/browser/sandbox/tschaub/feature/lib/OpenLayers/Control/Measure.js>. Bajo la misma licencia que OpenLayers.

### Búsqueda de topónimos

La herramienta para la búsqueda de topónimos es una extensión de OpenLayers.Control. Es de tipo predictivo: muestra resultados a medida que se va escribiendo en la caja de búsqueda. Para ello, realiza peticiones AJAX al servicio de toponimia (descrito en la sección de servicios web), que proporciona una lista de resultados con los siguientes datos: Nombre del topónimo, tipo de elemento, municipio y comarca al que pertenece, y coordenadas UTM.

Los resultados se muestran en una lista bajo la caja de búsqueda. Haciendo clic en uno de los resultados, el mapa se centra en el topónimo indicado.

Este control se puede reutilizar en cualquier otro visor basado en OpenLayers.

### Barra de escala

Se ha encapsulado la barra de escala gráfica de MapServer<sup>7</sup> en una clase derivada de OpenLayers.Control. Al pasar el puntero del ratón por encima, muestra un *tooltip* indicando la resolución del mapa en metros por píxel.

### Información de la escena

Proporciona información sobre los datos visibles: Nombres de los productos y enlaces a las respectivas leyendas. Esta información se obtiene mediante la consulta al servicio web getSceneInfo (consultar sección sobre servicios web).

## CACHEADO DE LOS DATOS

Vissir2 es un visor que ha de poder trabajar a pantalla completa y con unos tiempos de respuesta comparables a Google Maps. Los datos cartográficos se obtienen del servicio WMS ofrecido por shagrat<sup>8</sup>, accesible públicamente. El tiempo de respuesta a una petición WMS de 256x256 píxeles es de unos 300 ms. Recuperar un mapa de la extensión de una pantalla de 1024x768 conllevaría una demora inaceptable, e incluso puede llegar a comprometer la estabilidad del servidor, en especial durante episodios de carga con múltiples usuarios simultáneos.

Si es posible conseguir que las peticiones de los clientes sean predictibles y repetitivas, podemos generar la respuesta una sola vez, almacenarla y reutilizarla tantas veces como sea necesario. Este es el principio de un servicio de caché de teselas o *Tile Caché*. Es necesario pues restringir las peticiones a un número limitado de resoluciones. Dividir el mapa en una matriz de teselas de tamaño fijo, generalmente de 256x256 píxeles, también permite desplazamientos suaves sobre el mapa, y la reutilización de las imágenes en el cliente.

---

<sup>7</sup> Tim Schaub, 2005. Bajo licencia LGPL 2.1 o posterior.

<sup>8</sup> <http://shagrat.icc.cat/lizardtech/iserv/ows?REQUEST=GetCapabilities&SERVICE=WMS>

## Diseño de la caché

Los productos a mostrar se corresponden con las capas que proporciona el servicio WMS de shagrat:

Tabla 1: Productos cartográficos visualizables en Vissir2

Nombre de la capa	Descripción
mtc5m	Base topográfica 1:5 000 (50 cm/píxel)
mtc10m	Mapa topográfico 1:10 000 (1 m/píxel)
mtc25m	Base topográfica 1:25 000 (2 m/píxel), incompleta
mtc50m	Mapa topográfico 1:50 000 (5 m/píxel)
mtc250m	Mapa topográfico 1:250 000 (25 m/píxel)
orto5m	Ortoimagen 1:5 000 (50 cm/píxel)
orto25m	Ortoimagen 1:25 000 (2,5 m/píxel)
sat250m	Imagen landsat 1:250 000 (25 m/píxel)
mgc50m	Mapa geológico 1:50 000 (5 m/píxel)
mgc250m	Mapa geológico 1:250 000 (25 m/píxel)

Las capas están optimizadas para ser vistas en la resolución indicada. Por tanto, no tendrá sentido ver la base topográfica 1:5 000 a 25 m/píxel (imposible apreciar detalles), ni el mapa topográfico 1:250 000 a 50 cm/píxel (interpolación excesiva).

Se han clasificado los 10 productos en tres categorías: Cartografía topográfica, imagen y mapa geológico, y se ha establecido una escala de resoluciones creciente, donde en la máxima escala (nivel 0) cabe todo el territorio a representar en 256x256 píxeles, y la mínima (nivel 10) se corresponde con el mayor nivel de detalle disponible. De este modo se han determinado tres conjuntos de tile caché o pirámides, tal como se muestra en la figura.

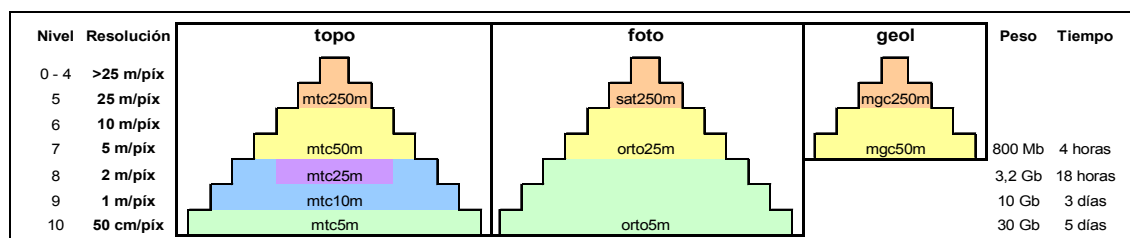


Figura 4: Las tres pirámides de caché construidas

En la figura se indican con distintos colores los productos con que se genera cada nivel de detalle. Se observa que el mapa geológico sólo llega a los 5 metros/píxel. Se ha indicado en las columnas de la derecha el espacio de disco que ocupan los niveles de más detalle, así como el tiempo necesario para generarlos. Estos datos son reales para la generación de la pirámide *foto*, pero pueden variar dependiendo de la naturaleza de los datos y del estado del servidor.

El nivel *topo* a 2 m/píxel es un caso especial. Al no disponer de cartografía 1:25 000 para todo el territorio, se utiliza el mapa topográfico 1:10 000 en las zonas donde falta aquella.



## Rendimiento

Para las pruebas de carga se ha utilizado el software JMeter<sup>9</sup> de Apache. El servicio WMS de shagrat proporciona un *throughput* de 3,5 teselas por segundo. El servicio de tile caché, una vez generadas las pirámides, proporciona un *throughput* de 75 teselas por segundo. Es decir, multiplica por 20 la capacidad. Este rendimiento se ha mantenido estable incluso bajo situaciones de estrés, con 50 peticiones simultáneas sostenidas durante un período de 5 minutos.

## Actualización automática

Los productos de escala 1:25 000 o mayor son series de hojas mosaicadas. La publicación de nuevas versiones de estas series nunca se produce sobre todo el territorio, sino que se producen actualizaciones parciales. Para hacer posible este tipo de actualizaciones se ha implementado una nueva función de TileCaché que invalida una región determinada de la caché y la regenera. Se ha incorporado este nuevo paso en el proceso de publicación de las series cartográficas, de modo que los datos del servicio WMS, de la caché y de los productos descargables es consistente.

## SERVICIOS WEB

Como se ha comentado, Vissir2 no es sólo el código Javascript que se ejecuta en el navegador. También forma parte de la aplicación la colección de servicios web que se consultan. El principal de estos servicios es el servicio de Tile Caché, que ha merecido una sección aparte. En esta sección se resumen el resto de servicios utilizados por Vissir2.

## Capas de información adicional

A la cartografía de base se pueden superponer los cortes de las series 1:5 000, 1:10 000, 1:25 000 y 1:50 000, además de información sobre las fechas de vuelo y de edición de la porción de cartografía visualizada. Estas capas se publican mediante WMS<sup>10</sup>. Los datos provienen de capas arcSDE publicadas mediante arcIMS 9.1 y el conector WMS de la versión 4.

## Toponimia

El servicio de toponimia forma parte de un servicio más amplio de geocodificación. A partir de una cadena de texto, proporciona una lista de candidatos ordenados según la similitud con la cadena proporcionada. Cada candidato contiene: Nombre del topónimo, categoría, municipios y comarcas a los que pertenece, y coordenadas UTM. El servicio se invoca mediante http GET, y devuelve los datos estructurados en formato JSON, fácilmente incorporables como un objeto de Javascript tras una petición AJAX.

---

<sup>9</sup> Licencia Apache 2.0

<sup>10</sup> [http://galileo.icc.cat/wms/servlet/icc\\_talls\\_v\\_r?REQUEST=GetCapabilities&SERVICE=WMS](http://galileo.icc.cat/wms/servlet/icc_talls_v_r?REQUEST=GetCapabilities&SERVICE=WMS)

## Impresión y *snapshot*

Se trata de un servicio que opera conjuntamente con un control de OpenLayers. Una vista de OpenLayers puede contener un conjunto de capas raster, cada una formada por varias teselas y con un valor de transparencia variable. Además, puede contener geometrías vectoriales superpuestas. Una clase de OpenLayers específica es capaz de extraer toda la información de una escena y enviarla a un servicio web mediante http POST.

El servicio es capaz de soldar las teselas, fundir las diversas capas según su orden y transparencia, y dibujar las geometrías vectoriales, todo ello en una sola imagen del tamaño de la composición original. Esta imagen se envía de vuelta al cliente, bien como jpeg, o bien comprimida en un fichero zip para su descarga.

## Transformación de coordenadas

Se trata del servicio de calculadoras geodésicas<sup>11</sup> existente en la web del ICC con anterioridad a Vissir2. Al tratarse de un servicio histórico, la respuesta tiene un formato no estándar. Con este servicio, se puede cambiar tanto la proyección (UTM, geodésicas) como el datum (ED50, ETRS89) de un par de coordenadas dado.

## Productos descargables

Se trata de un servicio llamado *getSheet*, que a partir de un par de coordenadas dado, devuelve una lista de los productos descargables disponibles en ese punto. Se trata de una lista de hojas. Cada una proporciona el nombre del producto, el número de hoja, el nombre de la hoja, y una lista de formatos disponibles, cada uno con un enlace al recurso descargable. Esta información está estructurada también siguiendo la notación JSON, para su integración en aplicaciones web.

Para proceder a la descarga efectiva del producto es necesario haberse dado de alta como usuario en la web del ICC y estar autenticado.

## INTEGRACIÓN CON OTRAS APLICACIONES

Vissir2 no sólo se utiliza para la visualización y distribución de la cartografía generada. También sirve para representar las listas de coordenadas transformadas mediante las calculadoras geodésicas de la web del ICC<sup>12</sup>.

Vissir2 puede incluirse además en cualquier página web fuera del ICC. Existen varias opciones:

- Función *embedded*. A través de la opción 'enlazar' se obtiene el código HTML necesario para incluir una versión reducida de Vissir2 en cualquier página web. Por ejemplo:

```
<iframe width="425" height="350" frameborder="0"
scrolling="no" marginheight="0" marginwidth="0"
src="http://www.icc.cat/vissir2/?
zoom=3&lat=4504616&lon=306984&layers=B00FFFFFFF&lang=es_ES
&embedded=true" ></iframe>
```

<sup>11</sup> [http://www.icc.cat/web/content/ca/prof/geodesia/eines\\_geodesiques.html](http://www.icc.cat/web/content/ca/prof/geodesia/eines_geodesiques.html)

<sup>12</sup> <http://www.icc.cat/web/content/es/prof/geodesia/calculadores.html>

- Función *permalink*. La misma opción 'enlazar' proporciona una URL con la que se puede enlazar a la vista actual del mapa incluyéndola en mensaje de correo electrónico o documento html. Por ejemplo:

[http://www.icc.cat/vissir2/?zoom=3&lat=4504616&lon=306984&layers=B00FFFFFFF&lang=es\\_ES](http://www.icc.cat/vissir2/?zoom=3&lat=4504616&lon=306984&layers=B00FFFFFFF&lang=es_ES)

- Abrir la aplicación con un topónimo. Basta con incluir el parámetro 'toponim' en la dirección. Ejemplo:

[http://www.icc.cat/vissir2/?lang=es\\_ES&toponim=manlleu](http://www.icc.cat/vissir2/?lang=es_ES&toponim=manlleu)

- Abrir con una coordenada expresada en coordenadas geográficas (datum ETRS89). Gracias a esta funcionalidad, se puede abrir Vissir2 desde los artículos georreferenciados de la wikipedia. Ejemplo:

<http://www.icc.cat/vissir2/?etrs89x=2.183333&etrs89y=41.383333>

## TRABAJO FUTURO

A corto y medio plazo se planea incrementar las funcionalidades actuales así como las fuentes de datos.

- Ampliación de la base toponímica.
- Búsqueda por calle y número.
- Importación y exportación de archivos kml y gpx.
- Consulta de cotas ortométricas.
- Ortoimagen a escala 1:1 000 en zonas urbanas.

Cualquier sugerencia o propuesta de nueva funcionalidad por parte de la comunidad será estudiada. Se pretende así adaptar la oferta de servicios a las demandas de una sociedad en continuo cambio.

## CONCLUSIONES

A la vista de los resultados conseguidos, Vissir 2 satisface los objetivos iniciales. Prueba del potencial y extensibilidad del código es la continua incorporación de nuevas funcionalidades. Asimismo la solidez de los componentes de base (OpenLayers, TileCache, etc) ha sido comprobada de forma fehaciente durante el tiempo que lleva Vissir2 en producción. Esto permite ser optimistas respecto al futuro y considerar la aplicación como una base sólida sobre la que incorporar nuevos contenidos y funcionalidad.

El diseño modular e interoperable también ha dado sus frutos. Otras aplicaciones del ICC como el geoportal de la IDEC y la Guia de Carrers entre otros han incorporado OpenLayers en su núcleo, y reutilizan algunos módulos, como la funcionalidad multiidioma, la búsqueda por topónimos y la Tile Caché.

## AGRADECIMIENTOS

A la comunidad de desarrolladores de OpenLayers y TileCache, gigantes sobre cuyos hombros hemos podido alzarnos.

A nuestros compañeros de la Unidad de Desarrollo Informático del ICC, en especial a Juan Carlos González, que estableció las bases del cambio de paradigma en los visores web (unificación de código, modularidad de las herramientas y uso de estándares para la interoperabilidad), y a Carmen Caballero, por hacer posible la actualización automática de la caché extendiendo TileCaché.

Este desarrollo es un proyecto, aún en evolución, del Institut Cartogràfic de Catalunya.

La aplicación se mantiene en producción gracias al saber hacer de la Unidad de Sistemas y a las abnegadas tareas de mantenimiento del Centro de Cálculo.

## REFERENCIAS

- **OpenLayers – Agile Geography in a Browser**  
*Tim Schaub*. FOSS4G, 2007.  
[http://www.foss4g2007.org/presentations/view.php?abstract\\_id=228](http://www.foss4g2007.org/presentations/view.php?abstract_id=228)
- **Tiling Maps for the Web with TileCache**  
*Schuyler Erle*. FOSS4G, 2007.  
[http://www.foss4g2007.org/presentations/view.php?abstract\\_id=248](http://www.foss4g2007.org/presentations/view.php?abstract_id=248)
- **The Open Geo-Stack: OpenLayers, GeoServer, PostGIS**  
*Tim Schaub, Chris Holmes, Justin Deoliveira*. Where 2.0, 2008.  
<http://en.oreilly.com/where2008/public/schedule/detail/1724>
- **Painless Javascript Using Prototype**  
*Dan Webb*. Sitepoint, 2006.  
<http://www.sitepoint.com/article/painless-javascript-prototype/>