

UNIVERSIDAD OBERTA DE CATALUNYA

I.T. Informática de Sistemas

**APLICACIÓN WEB PARA EL INVENTARIO DE PRESIONES EN RIOS CON
CARTODB**

Alumno/a: Antonio Cabezón Cascante

Dirigido por: Victor Velarde Gutierrez

Co-dirigido por:

CURSO 2012-2013 (Febrero)

UNIVERSITAT OBERTA DE CATALUNYA

1. Índice

1 Índice	2
2 Glosario	3
3 Plan de proyecto	5
4 Conocer los SIG	15
4.1 Neo-geografía	15
4.2 DMA	17
5 Análisis de la plataforma CartoDB	18
5.1 Desarrollo en CartoDB	18
5.1.1 CartoDB MAPs API	18
5.1.2 CartoDB SQL API	21
5.2 Uso de CartoDB	22
6 Origen de datos	26
6.1 Capas de ríos de CyL	26
6.2 Capa de riesgos	27
6.3 Capa de REN y zonas sensibles	27
7 Diseño	28
7.1 Diseño de la Base de Datos en CartoDB	28
7.2 Diseño de la aplicación	30
8 Implementación	31
8.1 Visualización de capas de CartoDB	31
8.2 Consultas a CartoDB	31
8.3 Conversiones de sistemas de coordenadas	31
8.4 Edición geográfica	32
9 Posibles Mejoras	32
10 Bibliografía	32
11 Anexo I Manual de uso	33
1 Interfaz	33
2 Gestión de capas	34
3 Consulta de datos, edición y eliminado	36
4 Añadir un nuevo riesgo	38

2. Glosario

- XYZ: Grid que utilizan los SIG para representar pirámides de imágenes.
- BYTEA: Tipo de dato especial en Postgres para almacenar datos binarios en postgres .
- API: Application Programming Interface / Interfaz para la programación de aplicaciones.
- CSS: Cascade Style Sheet, estándar de estilos para HTML.
- WMS: Web Map Service / Servidor de Mapas web.
- JSON: Javascript Notation Object.
- Postgis: Extensiones espaciales para Postgis.
- SQL: Structural Query Language.
- WFS: World Feature Service.
- WCS: Web Coverage Service.
- TMS: Tile Manager Server.
- CartoCSS: Especificación propietaria del estándar CSS, usado para la aplicar estilos en páginas HTML.
- SHP: “Shapefile”, formato de datos espaciales vectoriales de *ESRI*.
- REN: Acrónimo. Red de Espacios Naturales
- MARM: Acrónimo. *Ministerio de Medio Ambiente, Medio Rural y Marino de España*
- SCROLL: Barra de desplazamiento vertical que nos permite desplazarnos arriba y abajo por un contenido
- LEAFLET: API para el desarrollo de mapas en *JavaScript* en fase de desarrollo Alfa.
- GEOCODING: Servicio que nos permite entregar un topónimo, calle... y recibir unas coordenadas de su posición.

3. Plan de proyecto

Este apartado describe el Plan de Proyecto a seguir para la realización del *Trabajo de fin de Carrera*. El trabajo se titula **Aplicación web para el inventario de presiones en ríos con CartoDB**.

Después de esta breve presentación del proyecto, se presentan los objetivos ideales del proyecto.

Posteriormente se detalla la estructura del documento memoria final que se entregará tras la finalización del proyecto y que indica los puntos de los que se compone.

Se describen también las tareas y actividades en las que se descompone el proyecto.

Después se detalla el calendario de trabajo. En él se indican las horas de trabajo que se dedican al proyecto y los días en los que se realiza.

A continuación aparecen los hitos del proyecto. Con ellos, se establece la planificación general del proyecto y en la que se distribuyen las tareas y actividades en el tiempo y se establece el contenido de cada entrega

Más tarde se puede ver el diagrama de dependencias y el diagrama de Gantt para cada una de las entregas a efectuar.

Finalmente aparece el material necesario para llevar a cabo el proyecto, los riesgos y/o incidencias posibles en el transcurso del mismo (cuya solución es el plan de contingencia) y la bibliografía que se ha utilizado para la confección del presente documento.

Objetivos y alcance del proyecto

Los objetivos del trabajo son:

- Conocer los fundamentos básicos de los SIG y las nuevas herramientas geoespaciales en Internet.
- Aplicar los conocimientos adquiridos en SIG al desarrollo de una aplicación web para la gestión de información ambiental.
- Reflexionar sobre el concepto de computación en la nube aplicado a la información geográfica y sus implicaciones en las organizaciones actuales.
- Analizar los servicios web de una plataforma y desarrollar software sobre su API con solvencia.

Estructura del TFC

La estructura de la memoria será la siguiente:

Apartado	Subapartado	Páginas
Portada		1
Agradecimientos		1
Resumen		1
Índice contenidos		1
Índice figuras		1
Introducción		
	Justificación	1
	Objetivos	1
	Enfoque y método a seguir	1
	Planificación	1
	Productos Obtenidos	1
	Descripción de capítulos	1
Estudio general sobre SIG y dominio de la aplicación		
	Definiciones	3
	Cartografía	5
	CartoDB usuario	4
	DMA	3
Análisis de la plataforma CartoDB		
	CartoDB desarrollador	2
	Postgis	1
	Openlayers	1
Desarrollo de la aplicación web SIG		
	Introducción	1
	Diseño de capas	5
	Implementación	7
	Pruebas	2
Conclusiones		3
Glosario		2

Bibliografía		2
Anexos		8
Total		60

Tareas, planificación, hitos

Se indica a continuación una descripción de las tareas y actividades en que se dividirá el trabajo.

Tareas y actividades

A continuación se describen las tareas y sus actividades.

1. Definición del proyecto

1.1. Obtener la documentación inicial. Descarga de la información ofrecida por el consultor y otros existentes en la biblioteca.

1.2. Leer enunciado. Leer el enunciado junto con la documentación que se pueda antes del encuentro presencial.

1.3. Encuentro presencial.

2. Plan de trabajo (PAC1)

2.1. Obtener la documentación (encuentro presencial).

2.2. Leer documentación. Estructurar y esquematizar la documentación leída (otras memorias, otros planes de proyecto, documentos de estilo, documentos aportados por el consultor...).

2.3. Buscar bibliografía. Búsqueda de bibliografía útil para el proyecto a medio plazo: sitios web, libros, documentos ...

2.4. Redactar índice del borrador. Primero, redactar el índice para tener claros los puntos a incluir en el documento.

2.5. Redactar borrador. Descripción, objetivos, actividades, tareas, hitos, estructura del PFC, posibles incidencias y soluciones...

2.6. Planificar tareas. Establecer los hitos del proyecto así como los contenidos de PAC2 y PAC3. Establecer el calendario de trabajo. Teniendo en cuenta los posibles riesgos, temporizar las tareas proporcionalmente al esfuerzo que se calcula requerirán.

2.7. Entregar Plan de Proyecto. Este será el Plan de proyecto definitivo, que coincide con la PAC1.

3. Conocer los SIG y dominio de la aplicación

3.1. Recoger documentación.

3.2. Estudiar los SIG. Indicar qué son y los elementos que lo forman. Explicar cómo son los modelos de datos SIG, y qué incluyen. Conocer las herramientas de escritorio (QGis) para el análisis de los datos.

3.2. CartoDB. Analizar la plataforma desde el punto de vista del usuario, conocer los métodos para publicar nuevas capas, aplicar estilos...

3.2. Estudiar DMA. Conocer las presiones, su tipología, conocer donde obtener esos datos, su formato...

3.3. Redactar la documentación. Redactar la documentación asociada a la tarea repasando semántica y ortografía.

4. Análisis de la plataforma CartoDB

4.1. Obtener documentación del API de CartoDB y conocer la autenticación, el acceso a las capas subidas en modo usuario.

4.2. Conocer Postgis, extensión de Postgres para la gestión de datos espaciales.

4.3. Aprender a utilizar Openlayers como método de representación de los datos. Comprender los controles del API y funciones útiles.

4.6. Redactar la documentación. Documentar la tarea sin olvidar los repasos semántico y sintáctico.

5. Desarrollo de la aplicación web SIG

5.1. Diseño básico de la interfaz conforme a los requisitos del cliente. Preparación de las capas necesarias mediante los orígenes de datos y posterior publicación en la plataforma

5.2. Implementación de los controles necesarios y funcionamiento básico del mapa sobre openlayers.

5.3. Realización de un pequeño “set” de pruebas para verificar el correcto funcionamiento de la misma.

5.4. Redactar la documentación. Documentar la tarea sin olvidar los repasos semántico y sintáctico.

6. Confeccionar memoria

8.1. Redactar líneas futuras de trabajo. Indicar posibles caminos de continuación del proyecto.

8.2. Integrar en memoria. Recoger información redactada, darle sentido e integrarla en la memoria del proyecto.

8.3. Redactar conclusiones. Redacción de las conclusiones.

8.4. Redactar capítulos formales. Redactar capítulos formales (índice, introducción ...)

8.5. Hacer repaso semántico.

8.6. Hacer repaso sintáctico.

8.7. Hacer repaso ortográfico.

7. Confeccionar presentación

7.1. Escoger contenidos. Seleccionar contenidos de la memoria a incluir. Tener en cuenta diagramas, esquemas u objetos similares que ayuden a la comprensión del proyecto.

7.2. Redactar presentación. Uso de Powerpoint en la misma. Repaso semántico, sintáctico y ortográfico.

8. Debate virtual

8.1. Contestar preguntas. Permanecer a la espera de las preguntas del tribunal. Contestar lo mejor y más rápido posible.

Calendario de trabajo

Detalle del tiempo diario estimado dedicado al proyecto:

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
2h	2h	2h	2h	2h	5h	5h

Un total de 20 horas semanales, algunos días dependiendo de la cantidad de trabajo puede que sean 3 en vez de 2.

Semana	Nº Semana	Horas	Festivos
17/09	0	5	
24/09	1	10	
1/10	2	20	
8/10	3	23	12 fiesta
15/10	4	20	
22/10	5	20	
29/10	6	23	1 fiesta
5/11	7	20	
12/11	8	20	
19/11	9	20	
26/11	10	20	
3/12	11	23	6 fiesta
10/12	12	20	

17/12	13	20	
24/12	14	23	25 fiesta
31/12	15	23	1 fiesta
7/1	16	20	
14/1	17	20	
21/1	18	20	
28/1	19	20	
4/2	20	25	
11/2	21	25	
18/2	22	20	
25/2	23	8	
		468	Total horas aprox.

Hitos

Se establecen los hitos del proyecto, que serán directamente las entregas parciales.

Hitos	Fecha
Inicio	19/09/2012
Entrega del plan	02/10/2012
Entrega borrador de la PAC2	01/11/2012
Entrega de la PAC2	06/11/2012
Entrega borrador de la PAC3	03/12/2012
Entrega de la PAC3	11/12/2012
Entrega del borrador de la memoria y presentación	04/01/2013
Entrega de la memoria y presentación	07/01/2013
Fin del proyecto	25/01/2013

Planificación

Se muestra a continuación la planificación del proyecto, incluyendo el

contenido de cada PAC (se apunta el número de la tarea, su descripción, las actividades que contiene, el coste en horas y la fecha de inicio y de fin):

Tarea	Nombre	Actividades	Horas	F.Ini	F.Fin
1	Definición del proyecto		5	17/09	23/09
		Obtener documentación			
		Leer enunciado			
		Encuentro presencial			
2	Plan de trabajo		15	24/09	2/10
		Obtener documentación	1	24/09	
		Leer documentación	3	25/09	
		Buscar bibliografía	1	26/09	
		Redactar índice borrador	5	27/09	29/09
		Planificar tareas	5	30/09	02/10
	Entregar Plan del proyecto			02/10	
3	Conocer los SIG ...		38		
		Recoger documentación	2	03/10	
		Estudiar los SIG	10	04/10	6/10
		CartoDB	12	07/10	11/10
		Estudiar DMA	8	11/10	13/10
		Redactar la documentación	6	13/10	14/10
4	Análisis de la plataforma CartoDB		48		
		API de CartoDB	15	15/10	20/10
		Conocer Postgis	5	20/10	21/10
		Openlayers	20	22/10	27/10
		Redactar la documentación	8	28/10	30/10
	Entrega borrador			1/11	
	Correcciones		18	2/11	06/11

	Entrega definitiva			06/11	
5	Desarrollo de la aplicación web SIG		80		
		Diseño básico	20	07/11	13/11
		funcionamiento básico y pruebas	50	13/11	01/12
		Redactar la documentación	10	01/12	03/12
	Entrega borrador			03/12	
	Correcciones		22	05/12	11/12
	Entrega definitiva			11/12	
6	Confeccionar memoria		36		
		Redactar líneas futuras de trabajo	5	12/12	15/12
		Integrar en memoria	5	15/12	16/12
		Redactar conclusiones	5	16/12	18/12
		Redactar capítulos formales	10	18/12	22/12
		Hacer repaso semántico	5	22/12	23/12
		Hacer repaso sintáctico	4	24/12	25/12
		Hacer repaso ortográfico	2	25/12	26/12
7	Confeccionar presentación		28		
		Escoger contenidos	10	26/12	29/12
		Redactar presentación	18	29/12	03/01
	Entrega borrador			03/01	
	Correcciones		16	03/01	07/01
	Entrega definitiva			07/01	
8	Debate virtual				
		Contestar preguntas	10	25/01	

Diagrama de Gantt

Se muestra el diagrama de la Tarea 1:

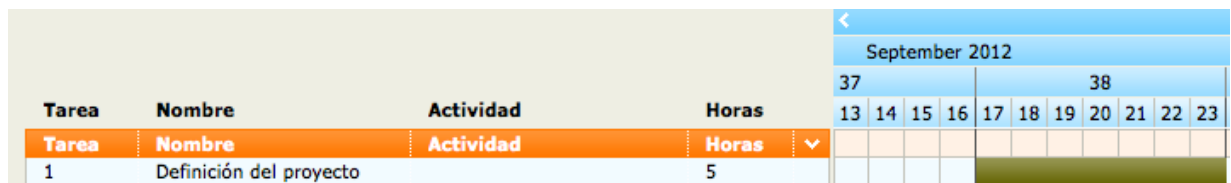


Diagrama de la tarea 2:



Diagrama de la tarea 3:



Diagrama de la tarea 4:



Diagrama de la tarea 5:

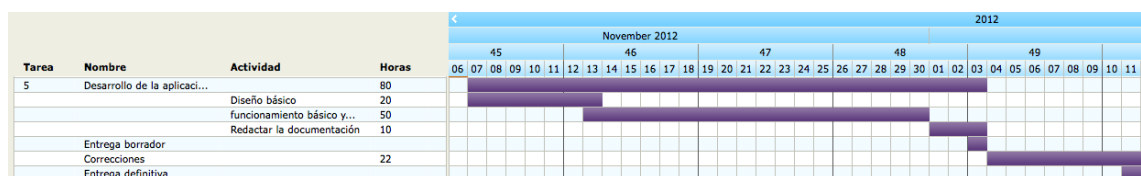


Diagrama de la tarea 6:



Diagrama de la tarea 7:



Diagrama de la tarea 8:



Material

Se indica a continuación el material necesario para llevar a cabo el proyecto.

QGis Lisboa (Software Gis), *CODA* (Editor CSS HTML y JAVASCRIPT) y *Firefox + Firebug* (Navegador y debugger)

Incidencias, riesgos y plan de contingencia

Se indican a continuación los riesgos e incidencias que se pueden producir a lo largo del proyecto. A continuación se indica la solución que se llevará a cabo en caso de producirse.

Avería de la estación de trabajo

Podría ocurrir que la estación de trabajo sufriera algún tipo de percance.

Plan de contingencia

Se intentará conseguir otro equipo rápidamente. Se efectuará un *commit* diario en un servidor *svn* propio.

Efectuar horas extras en el trabajo

Se prevén horas extras en el trabajo.

Plan de contingencia

Si hay cargas de trabajo se repartirán en los días siguientes y se aprovecharán los fines de semana para compensar

4. Conocer los SIG

4.1 Neogeografía

De 10 años a esta parte, los visores de mapas estaban restringidos a grandes administraciones, que creaban complejos desarrollos para publicar sus datos en internet y facilitar el acceso de los clientes. Grandes compañías como *ESRI* tenían casi el monopolio de estos sistemas de información y vendían sus desarrollos cerrados.

Esfuerzos comunitarios comenzaron el desarrollo de productos para servir mapas “libres” como por ejemplo *Mapserver* o *Geoserver*.

En 2006 se produjo lo que a mi criterio fue un punto de inflexión en el desarrollo de los mapas en internet, apareció *Google Maps* y *Earth*. De pronto los usuarios se dieron cuenta que los mapas y datos geográficos no eran un tabú, era increíble como se podía presentar información en un mapa, de manera intuitiva y muy útil.

Además estos servicios liberaron al usuario de necesitar una infraestructura gigante para poder servir sus mapas, ya que ellos sólo tenían que poner su información mediante sus APIs.

Los usuarios comenzaron a crear y publicar sus datos, en gran parte por la democratización de los dispositivos *GPS* y a la publicación de grandes cantidades de datos por parte de las administraciones, debido a los requerimientos de los usuarios que utilizaban estos formatos.

Empresas comenzaron a desarrollar un mercado de servicios alrededor de esta nueva necesidad de mapas y de información geográficamente referenciada que fueron floreciendo y otras se animaron a crear nuevas tecnologías.

Habiendo cada vez más información geográfica publicada, los usuarios disponían de herramientas para poder hacer sus propios mapas relevantes (Mashups). Surgieron grandes comunidades para crear datos en común como es el caso de *OpenStreetMap* con millones de actualizaciones de registros geográficos diarios.

En los últimos tiempos el número de protocolos en los servicios de mapas se ha incrementado, de los estándares básicos como *WMS* (*Web Map Service*) o *WFS* (*Web Feature Service*) se han desarrollado otros como por ejemplo *WFS-T* (Vectoriales con capacidad de edición al vuelo), *WCS* (*Web Coverage Service*) y *TMS* (*Tile Map Service* para coberturas del terreno). *SOS* para la monitorización...

Otro paradigma importante que ha cambiado en los últimos tiempos es donde se encuentra la información de los mapas, inicialmente esta información se debía publicar en servidores propios con los costes que esto tenía. Ahora se ofrecen servicios gratuitos y competentes (Amazon Web Services, *Google Engine*, Microsoft Cloud...) que nos permiten gran capacidad de almacenamiento procesamiento y acceso para hacer públicos nuestros datos y garantizar la interoperabilidad entre diferentes plataformas y servicios. Toda la gestión citada anteriormente de los datos se hace de forma transparente al usuario, de manera que puede acceder a ella desde cualquier parte del mundo.

4.2 DMA

Para el desarrollo de este proyecto se ha elegido la cuenca hidrográfica del Río Duero. En este proyecto no se va a utilizar en toda su extensión, sólo una parte de la misma, la que se encuentra englobada dentro de la Comunidad de Castilla y León, esto viene motivado por la dificultad de encontrar datos con coherencia para integrarlos y unirlos.

Para obtener estos datos se ha recurrido a contactos en las Administraciones Regionales (SIGMENA), que gustosamente me han permitido acceder a los datos.

Los datos que se han facilitado son:

- **Cuencas Cedex:**
Capa de polígonos, cuencas de los ríos en CyL.
- **Ríos:**
Capa lineal, que contiene los ríos de CyL.
- **Municipios:**
Capa de polígonos, con los municipios CyL
- **Provincias:**
Capa polígono, con las provincias de CyL.
- **Cascos urbanos:**
Capa polígonos.
- **Corine Land Cover:**
Capa polígonos, usos del suelo.
- **Mapa forestal:**
Capa polígonos, especies silvícolas en CyL
- **Accidentes geográficos:**
Puntos, Presas, construcciones relevantes en los medios fluviales.

Se han intentado obtener capas de vertidos o de industrias, pero ha sido imposible, a modo de prueba se ha creado una capa que contiene diferentes tipos de geometrías para las diferentes tipologías de vertidos.

5. Análisis de la plataforma *CartoDB*

La plataforma *CartoDB* ofrece dos perfiles, destinadas a los diferentes tipos de sus usuarios, el primer perfil es la del usuario sin conocimientos de programación, para ello ofrece una interfaz amigable para gestionar capas, realizar consultas, editar estilos... Por otro lado nos ofrece otro perfil para el usuario con conocimientos en desarrollo para el acceso a estos datos desde otras interfaces. Se comienza detallando el perfil técnico para después conocer el del usuario.

5.1 Desarrollo en *CartoDB*

El API de desarrollo de *CartoDB* se estructura de una manera muy parecida a cómo lo hace un servicio *REST*. El sistema no guarda estados de transacciones ni sesiones, para cada petición que hacemos tenemos que detallar parámetros de autenticación (Llave de nuestra sesión de usuario).

Dependiendo del API que queramos utilizar pasamos unos parámetros para indicar al servicio que es lo que necesitamos. Además indicamos también el “espacio de usuario” al que nos referimos al inicio de la URL.

`http://{cuenta}.cartodb.com`

El servicio nos concede el acceso a 2 APIS, una para el acceso a datos y vectores que denomina *CartoDB SQL API* y otra para el acceso a imágenes llamada *CartoDB MAPs API*.

5.1.1 *CartoDB MAPs API*

CartoDB MAPs API se trata de un servicio basado en *Mapnik*, que es una aplicación que “renderiza” los datos contenidos en un almacén, *Postgres* en este caso, en imágenes. Para ello usa definiciones de estilos *CartoCSS* que le indican a la aplicación que crea las imágenes

los colores, grosores y otros atributos de los datos. *Mapnik* genera mosaicos de imágenes que inmediatamente pasan a estar a disposición de los clientes.

Estos mosaicos de imágenes, son servidos mediante este API a los clientes.

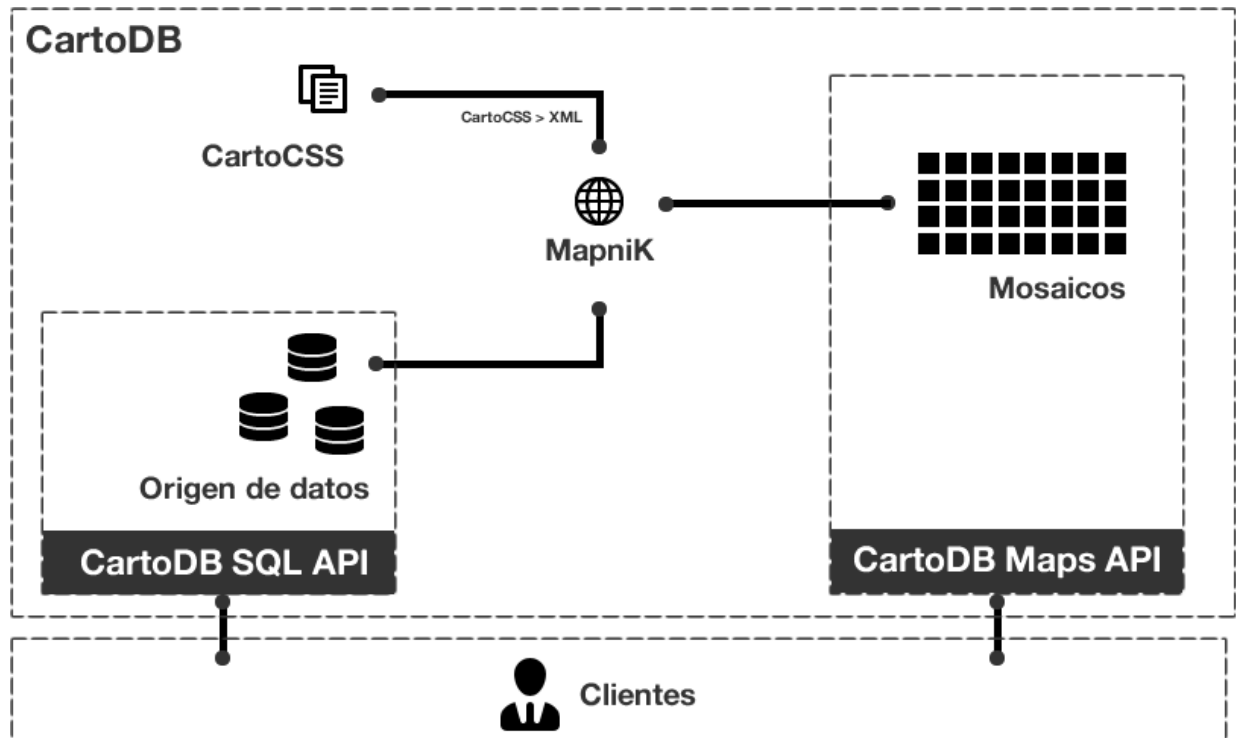


Fig.1 Esquema del servicio, elaboración propia

Los mosaicos o pirámides se ordenan en una rejilla llamada XYZ: esto es las imágenes se ordenan mediante el nivel de zoom y después la posición X, Y.

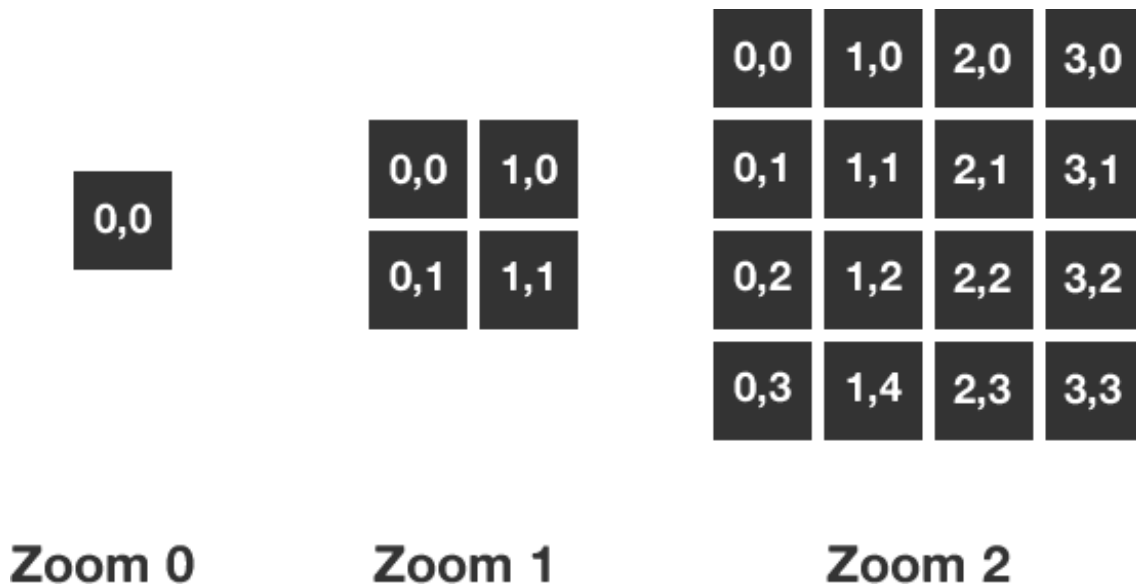


Fig.2 Pirámides, elaboración propia

Como se puede ver en el nivel de zoom 0 tenemos por ejemplo una única imagen, esta se referenciaría dentro del servicio como:
0/0/0.png

Para el siguiente zoom tenemos 4 imágenes por cada imagen del nivel anterior, en este caso 4 y como hemos visto antes se referenciarían de la misma manera pero cambiando el nivel de zoom:

1/0/0.png, 1/1/0.png, 1/0/1.png, 1/1/1.png

Sucesivamente por cada nivel de zoom se cuadruplica el número de imágenes.

0 – 1
1 – 4
2 – 16
3 – 256

Esta solución está siendo utilizada por las grandes compañías de datos ya que permite una gran escalabilidad horizontal, de manera que para servir estas imágenes no se necesitan servidores complejos sino simplemente servidores de contenido estático (*Google Maps*, *Bing Maps*...). Estos conjuntos de imágenes son sencillos de transportar a otros equipos y mejorar la redundancia y el tiempo de acceso al servicio. El almacenaje de estas imágenes, por lo menos en

este servicio, no se almacenan en ficheros de imagen sino en Postgres mediante campos del tipo *bytea*.

En contraprestación con otros protocolos, aunque esto no es un protocolo, sino que corresponde más a una arquitectura, como el **WMS** es que cada vez que hay cambios hay que redibujar las imágenes afectadas por el cambio. Dependiendo de la cantidad de datos y su resolución este proceso puede ser largo y no instantáneo.

5.1.2 CartoDB SQL API

Este API se encarga de devolver los datos contenidos en las tablas creadas. Para poder interactuar con ellas, indicamos en la URL una parámetro llamado “q” de query, para indicarle la consulta que queremos ejecutar sobre nuestro conjunto de datos. Estas consultas se especifican con la sintaxis de SQL estándar que Postgres cumple. CartoDB delega la responsabilidad de las consultas, sintaxis, errores sobre la propia base de datos Postgres, se limita a hacer de intermediario entre el usuario y este.

En estas consultas podemos utilizar las operaciones espaciales que Postgis nos aporta. Por ejemplo:

```
SELECT st_astext(ubicacion) FROM imagen  
>"POINT(28.27483403 -16.49179164)"
```

Podemos hacer operaciones de manera muy sencilla para saber si dos geometrías intersectan, crear un buffer a cierta distancia.

```
SELECT st_intersects(st_geomfromtext('POINT(28.27483403 -  
16.49179164)',4326), st_geomfromtext('POINT(28.27483403 -  
16.49179164)',4326))  
> t
```

En este ejemplo unimos varias operaciones para saber si un punto con un buffer a 1 Km intersecta con otro.

```
select st_intersects(st_buffer(st_geomfromtext('POINT(28 -  
16)',4326),1000), st_geomfromtext('POINT(28.27483403 -  
16.49179164)',4326))  
> t
```

Nos permite también realizar conversiones entre sistemas de coordenadas.

```
select st_astext(st_transform(st_geomfromtext('POINT(28.27483403
-16.49179164)',4326), 25830))
> "POINT(3985370.23787895 -2114662.30424947)"
```

Se ha experimentado otras muchas operaciones, interesantes, cálculo de áreas, perímetros, volúmenes, alturas, longitudes, distancias... [st_area, st_perimeter...]

El servidor responde a todas estas peticiones con datos en *JSON*, o su especificación *GeoJSON* que nos permite utilizar la respuesta inmediatamente desde el cliente sin tener que procesarla ya que son objetos *Javascript*.

El API también nos permite la inserción de registros y la actualización de los mismos mediante esta misma interfaz. Utilizando en vez de comandos *SELECT*, comandos *INSERT*, *UPDATE* y *DELETE*. De esta manera es muy fácil mantener el conjunto de datos desde el cliente.

```
INSERT INTO imagen (ubicacion) VALUES
(st_geomfromtext('POINT(28.27483403 -16.49179164)',4326))
```

Para clientes habituados a trabajar con *SQL* la forma de trabajar con el API es muy sencilla e intuitiva. No se necesitan herramientas de desarrollo propias ya que con el propio navegador es más que suficiente.

5.2 Uso de *CartoDB*

El uso de *CartoDB* es muy sencillo, una vez registrado en el sistema podemos acceder a nuestras tablas.



The screenshot shows the CartoDB web interface for a table named 'untitled_table'. At the top, there's a header with the table name, a 'PUBLIC' status button, and an 'edit tags' link. Below this are two tabs: 'Table' (selected) and 'Map'. To the right of the tabs is a 'SHARE THIS MAP' button. The main area displays a table with three columns: 'cartodb_id' (Number), 'the_geom' (Geometry), and 'description' (String). The first row shows the value '1' in the 'cartodb_id' column. There are icons for adding new rows and editing existing ones.

cartodb_id	the_geom	description
1		

Fig.3 Tablas en CartoDB

Cuando añadimos una nueva tabla se nos añaden una serie de campos, por defecto.

- **Cartodb_id**: Un número de serie único para poder hacer referencia a cada registro.
- **the_geom**: Una columna con nombre clásico de columna geométrica en *Postgres* que nos permite almacenar la geometría
- **description y name**: Dos columnas de descripción y nombre del registro.
- **created_at** : Columna de tipo estampa de tiempo, que nos indica cuando se creó el registro.
- **modified_at**: Columna del mismo tipo que la anterior que nos indica la fecha de la última modificación.

El interfaz nos ofrece herramientas para filtrar, ordenar, añadir nuestros datos, de una manera intuitiva y coherente. También se nos permite importar datos desde fuentes de datos propias como por ejemplo:

.CSV	Comma-separated values and Tab delimited file
.ZIP *	ESRI Shape File
.KML, .KMZ	Keyhole Markup Language
.XLS, .XLSX	Excel Spreadsheet
.GEOJSON, .JSON	GeoJSON
.GPX	GPS eXchange Format
.OSM, .BZ2	Open Street Map
.ODS	OpenDocument Spreadsheet
.ZIP	GeoTIFF

Fig.4 Formatos de importación

Podemos en cualquier momento visualizar los datos, en el mapa

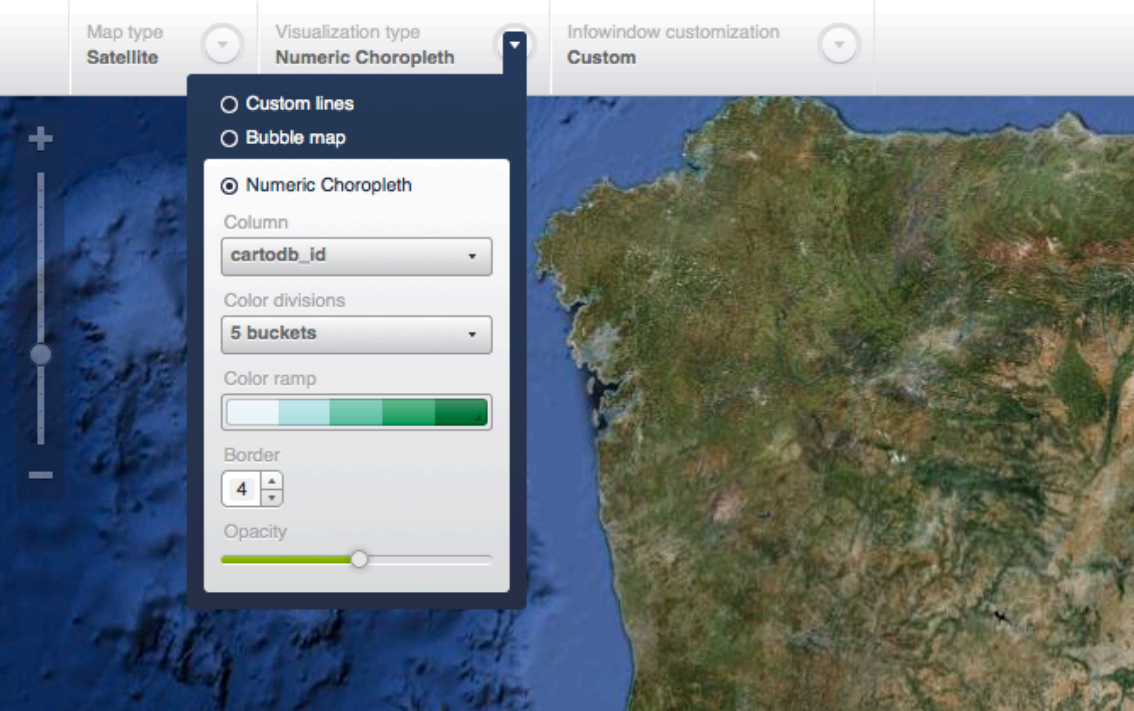


Fig.4 Estilos en el mapa

En esta sección accedemos a otra parte importante a la hora de mostrar nuestros datos, los estilos. De una manera sencilla podemos asignar colores, realizar temáticos...

Para un control mayor de los estilos también se nos permite acceder a un editor de estilos. El formato es muy parecido a los *CSS* de *HTML* como se ha mencionado anteriormente.

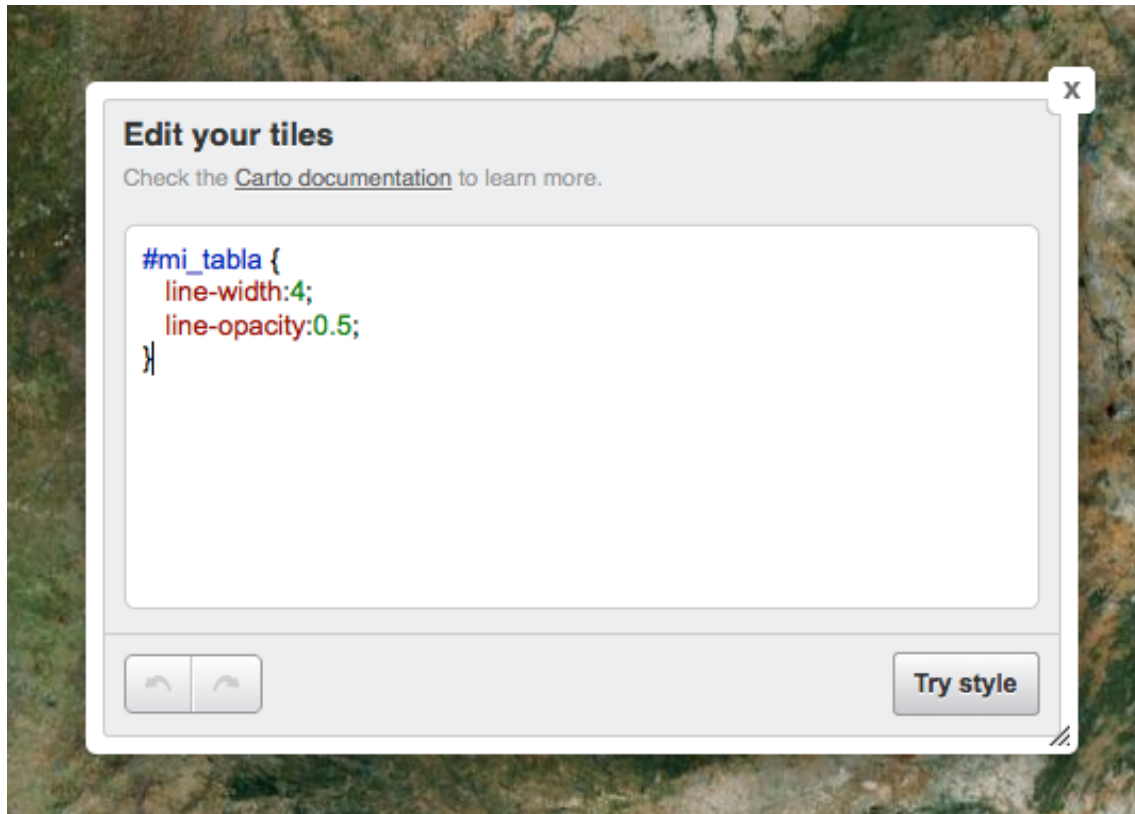


Fig.5 Editor de estilos

Para ello definimos un identificador de nuestra tabla con una almohadilla, y acto seguido le asignamos los atributos.

Con estos estilos podemos controlar todos los parámetros de visualización de los estilos, por ejemplo al nivel de zoom al que se muestran o que queremos que cambien.

```
#mi_tabla {  
  line-width: 4;  
  [zoom < 10] {  
    line-width: 3;  
    [zoom < 3] {  
      line-width: 2;  
    }  
  }  
}
```

Con esto conseguimos que dependiendo del nivel de zoom la línea tenga un atributo, en este caso el ancho, de manera que mejora la visualización ya que si hay muchas líneas se llenaría el mapa.

```
#mi_tabla {  
  line-width:4;  
  [ancho > 5] {  
    line-width:6;  
  }  
}
```

Podemos aplicar estilos utilizando la lógica, en este ejemplo, si la línea tiene un atributo ancho mayor que 5 aumentamos el tamaño de la misma, este ejemplo es útil para representar por ejemplo diferentes tipos de vías...

Finalmente podemos exportar todos nuestros datos de una manera sencilla a formatos como Excel, GeoJSON... También publicar nuestras creaciones y compartirlas con otros usuarios de una manera sencilla.

6. Origen de datos

6.1 Capa de ríos de Castilla y León

La capa de ríos, de la que parte la de vertidos, primeramente se limpió quitando elementos de otras cuencas y ríos procedentes de otras comunidades. Una vez simplificada y reproyectada al EPSG 4326 (procedía de 25830) se publicó en el servicio *CartoDB*, asignándole una representación sencilla e intuitiva, con los nombres de los ríos apareciendo a diferentes escalas. Esta importación se hizo mediante un archivo SHP, comprimido.

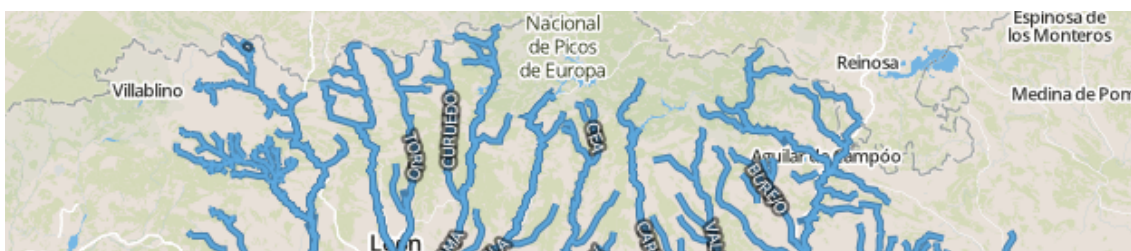


Fig. 6: Capa Ríos en *CartoDB*

6.2 Capa de riesgos

Para obtener un juego de datos coherentes, utilizando la capa de ríos, se ha realizado un buffer a 20 metros de todos los ríos y se han insertado 1500 puntos aleatorios, 150 líneas y 200 polígonos preferentemente en cascos urbanos. A estas geometrías se les ha dotado de su tipo de riesgo. Una vez subidas al servicio mediante una consulta SQL, se diseñaron los cartoCSS para cada tipo de riesgo.

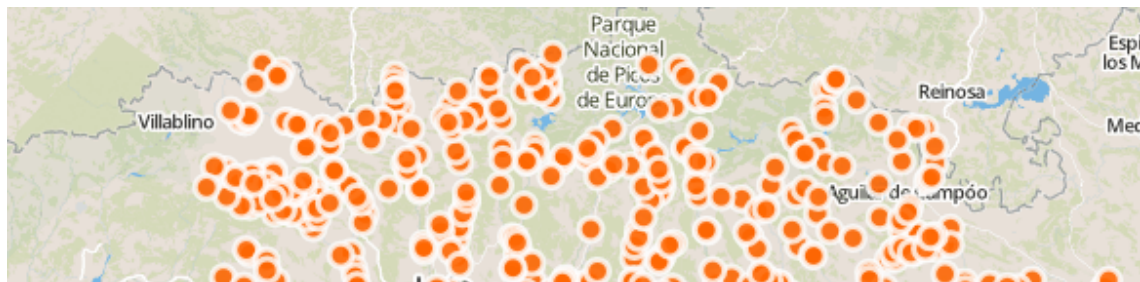


Fig. 7: Capa de vertidos

6.3 Capa REN y zonas vulnerables a vertidos de nitratos.

Estas capas se obtuvieron del [MARM](#) que las distribuye mediante archivos *shp* en su página web. En este caso las capas se re-proyectaron, cortándolas antes con el contorno de Castilla y León. Posteriormente fueron cargadas en *CartoDB*.



Fig. 8: Capa REN (Red de Espacios Naturales)



Fig. 9: Zonas vulnerables

7. Diseño

7.1 Diseño de la Base de datos en CartoDB

CartoDB, no permite, aunque en el sistema de base de datos en el que se apoya sí, la creación de claves primarias, relaciones y otras restricciones que se acostumbran a utilizar. Pero esto no es un error o falta de previsión el sistema, sino que es una ventaja ya que permite al usuario crear mapas de forma sencilla. Podríamos decir aunque no es correcto del todo que una tabla es un mapa, y en esta tabla tendremos todo lo necesario, no es correcto porque podemos crear relaciones entre tablas mediante consultas SQL y mediante “vistas” en la propia interfaz de *CartoDB* sin restricciones.

La capa de riesgos, como se ha comentado anteriormente procede de un análisis geográfico. Su diseño es muy simple, se compone de 3 campos propios más los que añade *CartoDB* por defecto.

- **Cartodb_id**: Un número de serie único para poder hacer referencia a cada registro.
- **the_geom**: Una columna con nombre clásico de columna geométrica en *Postgres* que nos permite almacenar la geometría, independientemente de su tipo, en este caso almacena líneas, puntos y polígonos.
- **descripcion**: Una columna que nos permite añadir información adicional del riesgo.
- **tipo**: Una columna de tipo numérico que almacena el tipo de riesgo.
- **created_at** : Columna de tipo estampa de tiempo, que nos indica cuando se creó el registro.
- **modified_at**: Columna del mismo tipo que la anterior que nos indica la fecha de la última modificación.

Para almacenar el tipo a cada categoría se le ha dado un valor de centena, y a cada subcategoría un valor de unidad, esto nos permite ampliar esta lista de manera sencilla por un lado y por otro almacenar este dato de manera sencilla en un único campo. Este es un listado de las diferentes categorías, subcategorías, tipos de geometrías y valores de identificación asignados.

- 100 Fuentes puntuales**
 - 101** Vertidos [*Punto*]
 - 102** Vertederos [*Punto*]
- 200 Fuentes difusas**
 - 201** Agricultura [*Polígono*]
 - 202** Ganadería [*Polígono*]
 - 203** Aeropuertos [*Punto*]
 - 204** Vías de transporte [*Línea*]
 - 205** Suelos contaminados [*Polígono*]
 - 206** Zonas urbanas [*Polígono*]
 - 207** Zonas mineras [*Polígono*]
 - 208** Zonas recreativas [*Polígono*]
 - 209** Praderas [*Polígono*]
 - 210** Gasolineras [*Punto*]
- 300 Extracciones de agua**
 - 301** Concesiones de agua [*Punto*]
- 400 Regulaciones de agua**
 - 401** Regulación embalses [*Línea*]
 - 402** Desvíos Hidroeléctricos [*Línea*]
 - 403** Inc. por Traslase [*Línea*]
- 500 Alteraciones morfológicas**
 - 501** Presas [*Punto*]
 - 502** Azudes [*Punto*]
 - 503** Canalizaciones [*Línea*]
 - 504** Protección Márgenes [*Línea*]
 - 505** Dragados [*Punto*]
 - 506** Recrecimiento Lagos [*Línea*]
 - 507** Modificación conexiones [*Línea*]
 - 508** Coberturas cauces [*Línea*]
- 600 Otras incidencias antropogénicas**
 - 601** Especies alóctonas [*Polígono*]
 - 602** Sedimentos contaminados [*Polígono*]
 - 603** Actividades recreativas [*Línea*]
- 700 Usos del suelo**
 - 701** Incendios forestales [*Polígono*]
 - 702** Explotación forestal [*Polígono*]
 - 703** Ocupación Márgenes [*Polígono*]
 - 704** Extracción Áridos [*Polígono*]
 - 705** Otros elementos [*Polígono*]

7.2 Diseño de la aplicación

Aunque en la propuesta del proyecto no era un requisito, se ha decidido dotar a la aplicación de un pequeño sistema de registro de usuarios, con el fin de proteger la edición de los datos y por otro lado para permitir la consulta de los mismos por parte de otros usuarios sin permisos en modo de visualización únicamente.

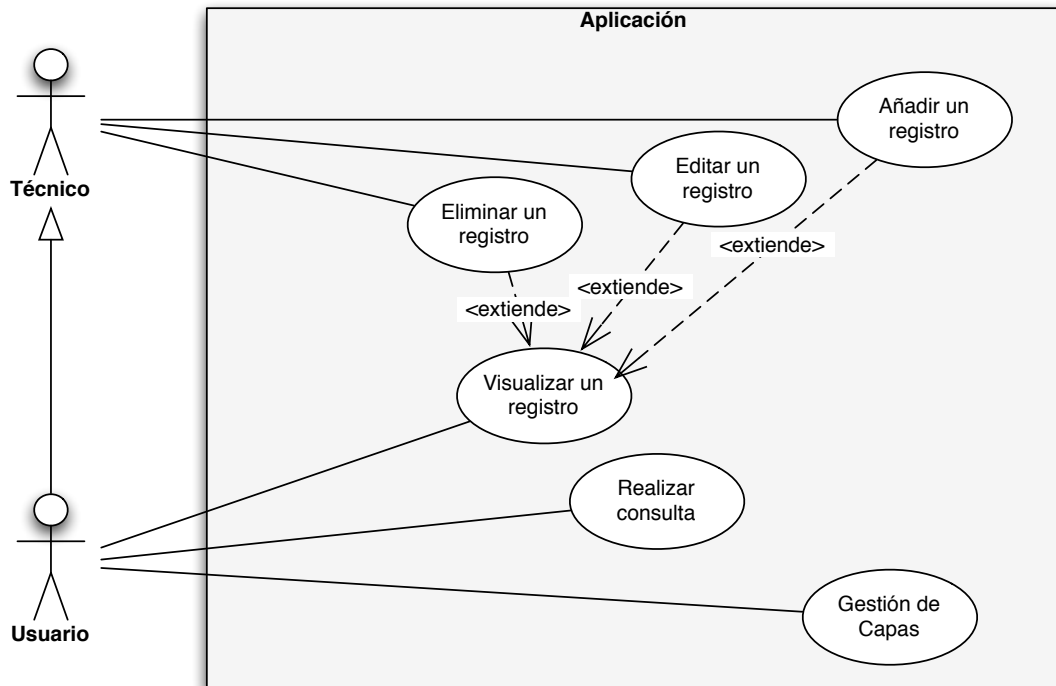


Fig. 10: Casos de uso

7.2.1 Caso de uso Gestión de capas

El usuario puede listar las capas, activarlas, desactivarlas, puede ver la leyenda de la capa, su descripción y activar y desactivar las subcapas.

7.2.2 Caso de uso Realizar consulta

El usuario puede realizar una petición para obtener un listado de los registros de resultado de una consulta.

7.2.3 Caso de uso Visualizar un registro

Permite al usuario ver los datos de un elemento en concreto, su posición, descripción...

7.2.4 Caso de uso Eliminar un registro

Permite al técnico eliminar un registro. Se apoya en la visualización del registro.

7.2.5 Caso de uso Añadir un registro

Permite al técnico añadir un nuevo registro. Se apoya en la visualización del registro.

7.2.6 Caso de uso Editar un registro

Permite al técnico editar un registro existente. Se apoya en la visualización del registro.

8. Implementación

Para el desarrollo de la aplicación se utiliza *Openlayers*, que es la solución que a primera vista ofrecía menos posibilidades nativas con el API de *CartoDB*. Con esta aplicación se ha intentado cubrir todas las opciones que ofrece el API de *CartoDB* (inserción, eliminado, consulta de datos, de imágenes...). Antes de comenzar se creó un test de desarrollo que especificaba todas las funciones que el API de *Openlayers* debía satisfacer para realizar el proyecto.

8.1 Visualización de capas de CartoDB:

Openlayers sí ofrece tipos de capas para conectarse a los servicios de *CartoDB*. La clase XYZ lo permite de una manera sencilla pero básica. No obstante se tuvo que extender esta clase para poder introducir los estilos y consultas dinámicas que ofrece *CartoDB*. Para ello se creó una clase *CartoDB.js* que solventa estas necesidades.

8.2 Consultas a CartoDB:

CartoDB en otros APIs (*Leaflet*), permite que las consultas de datos se hagan de un modo sencillo, que al hacer clic en un punto aparezca un pop-up prediseñado ya en *CartoDB*. En *Openlayers* esta opción no existe y hay que crear las consultas pertinentes para crear este tipo de comportamiento.

8.3 Conversiones entre sistemas de coordenadas:

Con los orígenes de datos de los que se disponían, *CartoDB* para sus imágenes utiliza el EPSG 900913 y para almacenar la información 4326 se necesitaban soluciones para convertir de uno a otro de manera sencilla.

8.4 Edición geográfica:

Se necesita poder crear nuevos puntos. Digitalizando directamente encima del mapa por parte del usuario. *Openlayers* tiene herramientas para resolver este requerimiento.

9. Posibles mejoras

Se podría dotar al visor de un sistema de localización de pueblos, calles, utilizando algún API de geocoding de proveedores como *Google* o *Yahoo*. Se debería invertir un poco más de tiempo en limar posibles errores no detectados en la aplicación, que seguro que los hay.

Con el fin de controlar un poco más los datos sería interesante introducir un sistema de permisos, de manera que cada usuario sólo pudiera editar sus puntos.

10. Bibliografía

- Manual *Postgis* 1.5
<http://postgis.refractory.net/documentation/manual-1.5/>
- Manual *Postgres* 9.2
<http://www.postgresql.org/docs/9.2/interactive/index.html>
- Manuales en *CartoDB*
<http://developers.cartodb.com/documentation/cartodb-apis.html>
- WorkShops de Openlayers
<http://workshops.opengeo.org/openlayers-intro/layers/cached.html>
- WiKi Openstreetmap
http://wiki.openstreetmap.org/wiki/Creating_your_own_tiles
- Manuales del API en *CartoDB*
<http://developers.cartodb.com/documentation/cartodb-apis.html>

- Documentación de *Openlayers*
<http://dev.openlayers.org>
- Trac de *Openlayers*
<http://trac.osgeo.org/openlayers/>

11. Anexo I Manual de uso

1 Interfaz

Se ha tratado al crear la aplicación que la interfaz del usuario fuera sencilla, limpia y clara. El mapa ocupa toda la pantalla y la información aparece ordenada en todo momento. La interfaz se divide en 3 partes básicas.

Acciones:

Se sitúa en la parte superior izquierda del mapa y contiene: un control de zoom, que nos permite acercarnos y alejarnos así como movernos en el mapa. Inmediatamente a su derecha tenemos el gestor de capas, que nos permite ver cuales tenemos activas e interactuar con ellas. A la derecha tenemos el botón de información que nos permitirá consultar datos en el mapa. Por último tenemos el botón para añadir un nuevo vertido.



Fig. 11: Zona de acciones del mapa

Mapa:

Ocupa toda la pantalla, por defecto nos permite actuar con el de manera natural, podemos arrastrarlo, acercarnos haciendo doble clic o con la rueda del ratón.

Paneles:

Ubicados en la parte de la derecha ocupando un tercio de la pantalla, muestran información relativa a la acción que estamos realizando, en unos casos será la gestión de capas, en otra los resultados de una búsqueda.

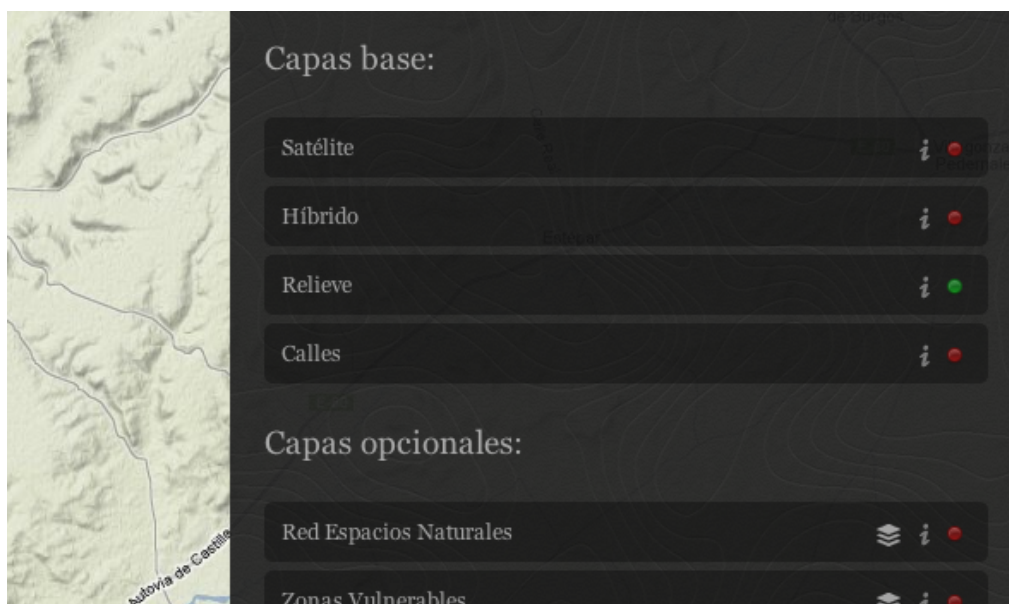


Fig. 12: Paneles de información

2 Gestión de capas

Este panel se abre por defecto al iniciar la aplicación, se puede abrir u ocultar en la zona de acciones, en el botón de gestor de capas. Contiene todas las capas que podemos activar y desactivar en el mapa. Se divide en dos partes bien diferenciadas, Capas base y Capas opcionales.

Capas base contiene capas entre las cuales una siempre está activada. Esta capa le da contexto al mapa. En todas estas capas se utiliza *Google* como proveedor de servicios. Por el contrario las capas opcionales, son, en sí las capas del proyecto, podemos activarlas o desactivarlas, no tienen que estar siempre activas.

Las capas se listan una bajo la otra, indicándonos su nombre y a la derecha unos iconos que nos permiten interactuar con ellas. El icono del punto verde o rojo nos indica si la capa está activada o desactivada podemos hacer clic para cambiar su estado. Si estamos

en las capas base, al activar una capa se desactiva la que estuviera activada.

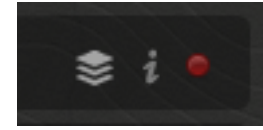


Fig. 13: Botón rojo, capa desactivada

En todas las capas disponemos además de un botón información representado con una “i” que al pulsar nos muestra la descripción de la capa. Si volvemos a hacer clic esta información se repliega y oculta otra vez.

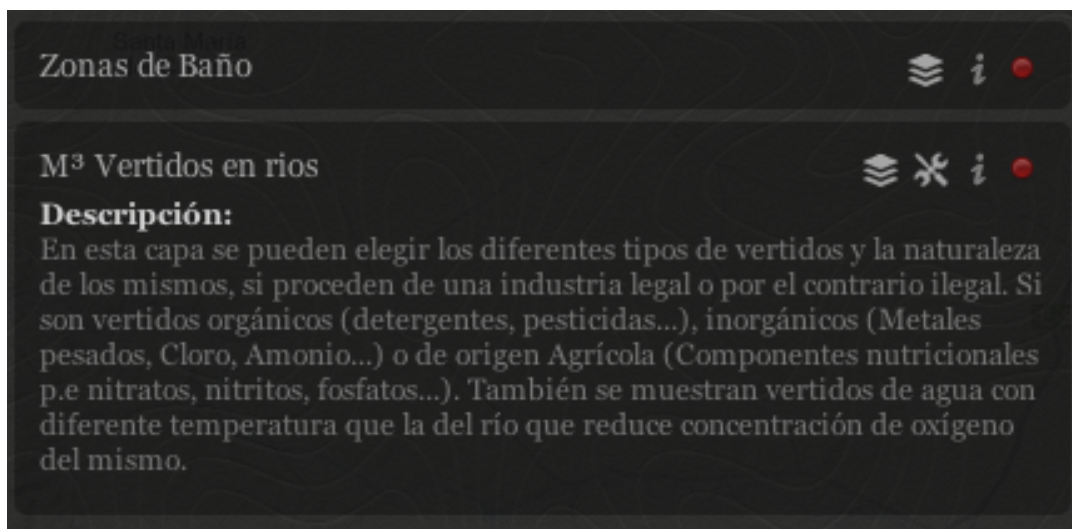


Fig. 14: Información de la capa

En algunas capas además disponemos de un icono igual que el del gestor de capas, que muestra la leyenda de la propia capa.

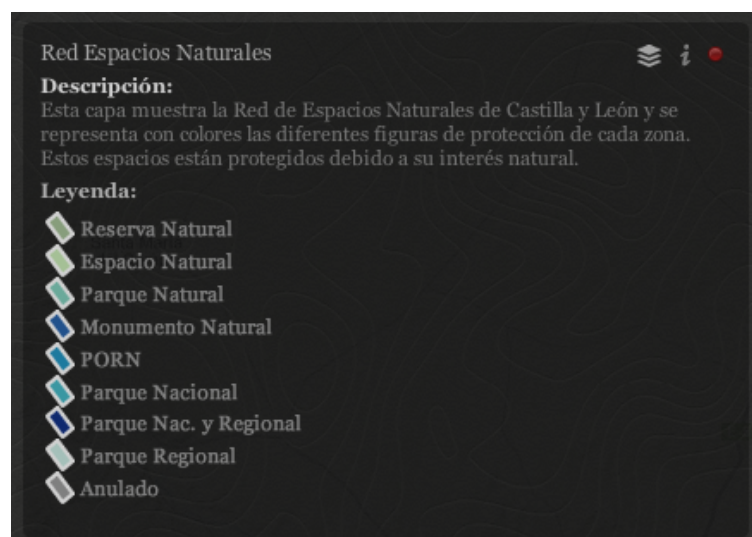


Fig. 15: Leyenda de la capa

Por último en algunas capas se permite, dentro de la propia capa seleccionar diferentes subtipos de riesgo.

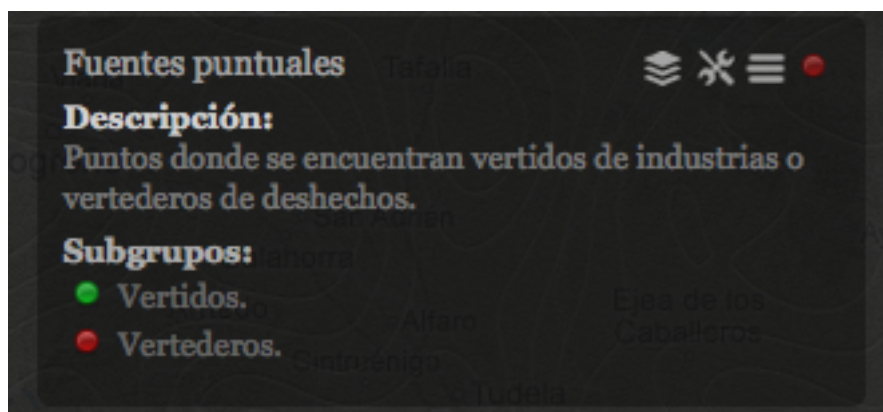


Fig. 16: Temáticos

Estos sub-tipos actúan como las capas base, únicamente uno puede estar encendido. La leyenda cambiará cada vez que cambiemos estas subcategorías ya que la representación es diferente.

Como se ha explicado estas herramientas se activan mostrando su contenido con un clic y se ocultan con otro, tratando de ocupar la mínima cantidad de espacio en el panel. Para resoluciones bajas, se mostrará un scroll para desplazarse por las capas verticalmente.

3 Consulta de datos, edición y eliminado

Una vez conocido el método de interacción con las capas, vamos a intentar obtener información de la única capa consultable, la capa de vertidos. Debemos tenerla encendida con cualquier temático activado.



Fig. 17: Capa de vertidos activada

Pulsamos en la zona de acciones, al lado de la barra de zoom el icono con la “i” . Se abre el panel de búsquedas, donde podemos

seleccionar la capa en la que queremos consultar y el botón para activar y desactivar la consulta. Una vez seleccionada la capa y habiendo hecho clic en el botón Consulta Desactivada, podemos crear un rectángulo en el mapa. El panel de la derecha cambiará y si hemos seleccionado algún riesgo, nos los mostrará textualmente, de la misma manera que las capas.

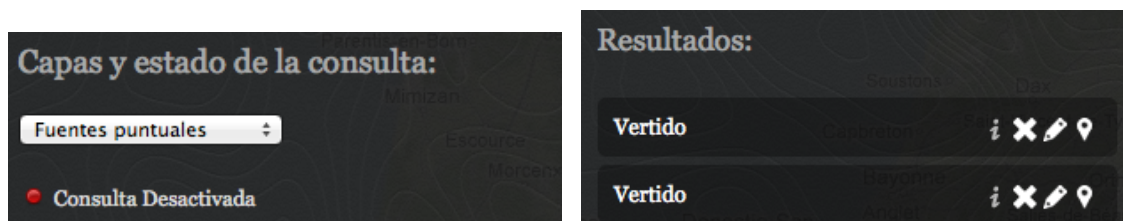


Fig. 18 y 19: Estado de la consulta y resultados encontrados en el mapa

Si somos técnicos nos aparecerán los botones de edición y eliminado de la capa además por cada registro encontrado nos aparecerá una línea en el listado, con opciones para cada resultado. Si pulsamos en la “i” de información del resultado, obtendremos los datos de la empresa y los datos de sus vertidos.

Podemos eliminar el registro, haciendo clic en el aspa que aparece al lado de cada resultado, previa confirmación del usuario. También podemos centrar y hacer zoom en el mapa a la posición de un vertido, mediante el icono de la marca.

Para parar de consultar información basta con pulsar el botón de información en la zona de acciones, este estará mas translúcido debido a que está en funcionamiento. Podemos si estamos registrados editar el registro seleccionado. Sólo podremos cambiar el registro a los riesgos que comparten el mismo tipo de geometría.

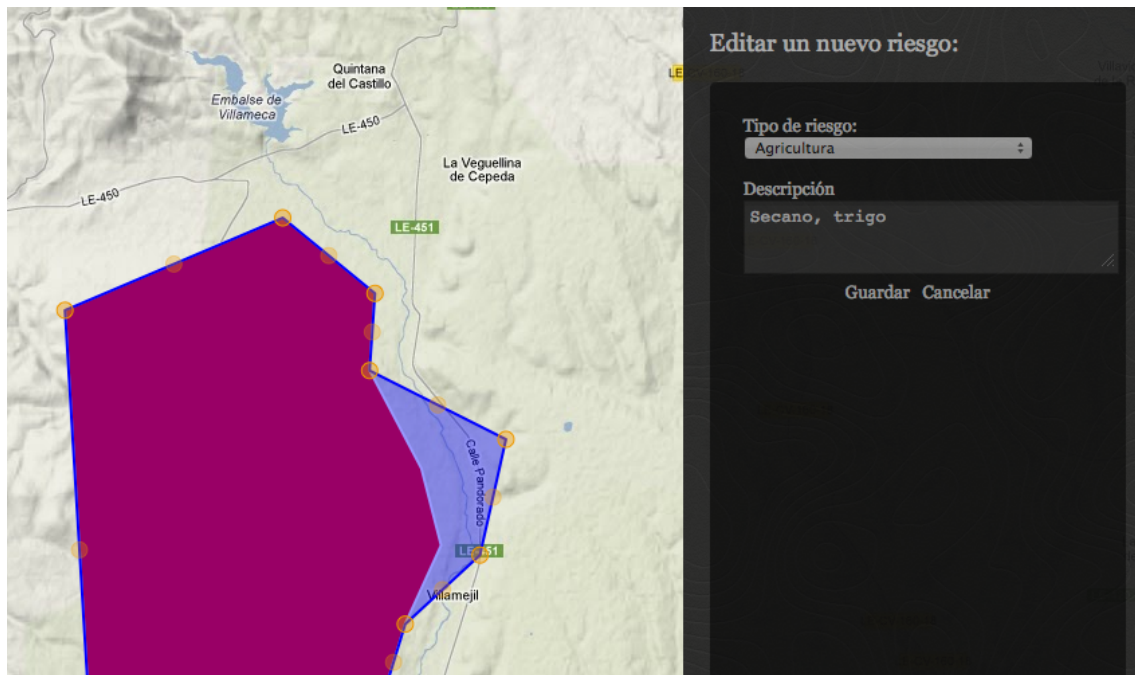


Fig. 20 Editando un registro

4 Añadir un nuevo riesgo

El usuario puede añadir un nuevo riesgo, para ello debe pulsar en la zona de acciones el botón con el signo más. En el panel aparece el formulario para introducirlo, dependiendo del tipo de riesgo la geometría a dibujar es diferente, si cambiamos y ya hemos dibujado algo, esta desaparecerá.



Fig. 21: Panel de alta del nuevo vertido

Una vez dibujada la geometría podemos cambiar la geometría en caso de que el resultado no sea satisfactorio, haciendo clic sobre ella. En cualquier momento podemos cancelar este proceso haciendo clic en el botón donde hemos comenzado o en el de cancelar del formulario.