

# UNIVERSIDAD OBERTA DE CATALUNYA

I.T. Informática de Sistemas

Aplicación web para el inventario  
de presiones en ríos con CartoDB

Alumno/a: Pío Alberto Sierra Rodríguez

Dirigido por: Víctor Velarde Gutiérrez

Co-dirigido por: Antoni Pérez-Navarro

CURSO 2012-13 (Septiembre/Enero)

## RESUMEN

Esta memoria describe el desarrollo un Trabajo de Fin de Carrera de Ingeniería de Informática de Sistemas, con el objetivo de crear una aplicación de mapas para Internet (SIG WEB). Dicha aplicación consistirá en gestionar el inventario de presiones de la Confederación Hidrográfica de Tajo. Como requisitos iniciales se especifica que el proyecto ha de utilizar la plataforma CartoDB para almacenar la información, y que para la visualización geográfica se utilizará una librería de mapas para JavaScript a elegir entre GoogleMaps, Leaflet u OpenLayers. Por este motivo también será necesario un proceso de selección previa, entre estas librerías.

La presente memoria incluye información sobre los sistemas GIS, los servicios Cloud, la arquitectura de CartoDB, un somero análisis de algunas de las citadas bibliotecas de mapas disponibles y toda la información relacionada con el propio desarrollo de la aplicación.

En lo posible se ha procurado priorizar la información que puede tener un carácter de innovación, aunque por supuesto queda al arbitrio de cada lector y sus conocimientos previos el considerar así cada apartado.

## ÍNDICE DE CONTENIDOS

Resumen .....	2
Índice de Ilustraciones.....	5
1. Introducción .....	6
1.1. Objetivos.....	6
1.2. Planificación.....	7
1.3. Productos obtenidos.....	9
1.3.1. Memoria del Trabajo.....	9
1.3.2. Presentación en Vídeo.....	10
1.3.3. Aplicación Web.....	10
2. Análisis del dominio .....	12
3. SIG en un entorno Cloud.....	13
3.1. Definición de entorno Cloud.....	13
3.2. CartoDB. Un SIG basado en cloud.....	14
3.2.1. Características principales de CartoDB.....	16
3.2.2. Arquitectura de CartoDB.....	19
3.3. Ventajas e inconvenientes de CartoDB como entorno Cloud .....	21
3.3.1. Ventajas .....	21
3.3.2. Inconvenientes .....	22
4. Diseño del SIG Web.....	23
4.1. Análisis para el diseño del SIG Web .....	23
4.1.1. Casos de uso .....	23
4.1.2. Diagrama de secuencia .....	26
4.2. Diseño de la base de datos .....	27
4.3. Selección de una librería de mapas .....	29
4.3.1. Google Maps .....	31
4.3.2. Leaflet.....	33
4.3.3. OpenLayers.....	33

# Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

4.3.4.    Selección final de la librería (Leaflet) .....	34
4.4.    Codificación del SIG Web .....	35
5.    Conclusiones .....	38
5.1.    Futuras líneas de trabajo. ....	39
Glosario .....	40
Bibliografía.....	42
Anexos.....	44
Anexo A: Seguimiento del Plan .....	44
Anexo B: Recursos.....	46
Anexo C: CartoDB.js .....	46

## ÍNDICE DE ILUSTRACIONES

Ilustración 1: Presentación del trabajo .....	10
Ilustración 2: Interfaz web de la aplicación (SIG WEB).....	11
Ilustración 3 Grupos y tipos de presiones .....	12
Ilustración 4: Planes de uso de CartoDB.....	14
Ilustración 5: Ejemplo de uso de CartoDB por The Wall Street Journal .....	15
Ilustración 6: Formatos de archivo de datos admitidos por CartoDB.....	17
Ilustración 7: Funciones de CartoDB comparadas con servicios similares.....	18
Ilustración 8: Estructura interna de CartoDB.....	19
Ilustración 9: Casos de uso .....	24
Ilustración 10: Diagrama de secuencia .....	26
Ilustración 11: Modelo ER de la base de datos.....	27
Ilustración 12: Tablas de la base de datos .....	29
Ilustración 13: Ejemplos de CartoDB.....	30
Ilustración 14: Valoración de funcionalidades de las librerías de mapas.....	31
Ilustración 15: Diferencias técnicas en licencias de Google Maps .....	32
Ilustración 16: Ejemplo de uso de CartoDB con OpenLayers .....	34
Ilustración 17: Estilos de CartoCSS para las presiones .....	36
Ilustración 18: Estructura de la aplicación Sig Web .....	37
Ilustración 19: Uso de visualizaciones con CartoDB.js .....	47
Tabla 1: Plan de trabajo.....	7

## 1. INTRODUCCIÓN

Los ríos se ven sometidos en nuestra sociedad a gran número de presiones, definiendo como tales toda aquella acción, directa o indirecta, que puede resultar un riesgo para su estado. Así presiones pueden ser desde la existencia de una población, hasta infraestructuras, vertidos o casos de deforestación.

Debido al alto valor ecológico de los ríos y a que son una importante fuente de recursos, es importante mantener un control sobre estas presiones. Con ese propósito se crea en el ámbito de la unión Europea la Directiva Marco del Agua (DMA-200/60/CE) que busca garantizar la calidad de las aguas mediante una gestión integrada de sus recursos. Una de las acciones a las que emplaza a las administraciones públicas es un inventario de presiones como el que es objeto del presente proyecto.

### 1.1. OBJETIVOS

Durante la realización del presente Trabajo de Fin de Carrera (TFC), se ha buscado la consecución de determinados objetivos, que engloban tanto competencias concretas como desarrollo de conocimientos más generales.

De forma global se han trabajado en los siguientes aspectos:

- Fundamentos básicos de los SIG, con especial foco en las nuevas herramientas sobre ellos disponibles a través de Internet.
- Desarrollo a partir de dichos conocimientos de una aplicación web de SIG.

Dichos objetivos se concretan en:

- Crear un Plan de Trabajo para el proyecto.
- Realizar un seguimiento del mismo durante toda la duración del proyecto.
- Adquirir la competencia necesaria para el manejo de CartoDB como usuario y, sobre todo, como programador.
- Realizar un trabajo de selección de una librería para mapas para JavaScript.
- Adquirir la competencia necesaria en el uso de la librería seleccionada.
- Desarrollar una aplicación web con CartoDB y la librería seleccionada que cumpla los requisitos indicados en proyecto (cliente).

# Aplicación web para el inventario de presiones en ríos con CartoDB

PIÓ ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

- Construir una Memoria (este documento) del TFC que se adecúe a los requisitos de calidad y contenidos exigidos.
- Sintetizar la información sobre el trabajo en una presentación en vídeo.

## 1.2. PLANIFICACIÓN

Para realizar el Trabajo se preparó una planificación acorde a los requisitos que se procuró seguir lo más fielmente posible. La siguiente tabla presenta la planificación de las tareas con sus horas correspondientes.

**TABLA 1: PLAN DE TRABAJO**

	Tarea / Hito	Horas	Fecha inicio	Fecha fin
<b>1</b>	Definición del proyecto		19/09/2012	26/09/2012
<b>1-1</b>	Descarga de documentos	1,5	19/09/2012	20/09/2012
<b>1-2</b>	Análisis del enunciado	6	21/09/2012	25/09/2012
<b>1-3</b>	Videoconferencia Inicial	1,5	26/09/2012	26/09/2012
<b>2</b>	Plan de Trabajo		25/09/2012	02/10/2012
<b>2-1</b>	Definir hitos	1	25/09/2012	25/09/2012
<b>2-1</b>	Crear calendario para el trabajo	0,5	26/09/2012	26/09/2012
<b>2-2</b>	Redactar borrador	1,5	26/09/2012	26/09/2012
	Entregar borrador		26/09/2012	26/09/2012
<b>2-3</b>	Corregir borrador	4	01/10/2012	02/10/2012
	Entregar Plan		02/10/2012	02/10/2012
<b>3</b>	PEC2		03/10/2012	06/11/2012
<b>3-1</b>	Estudio inicial sobre SIG		03/10/2012	08/10/2012
<b>3-1-1</b>	Recopilación de documentación	2	03/10/2012	04/10/2012
<b>3-1-2</b>	Lectura de documentación	7	03/10/2012	07/10/2012
<b>3-1-3</b>	Redacción de la Memoria	4	08/10/2012	08/10/2012
<b>3-2</b>	Estudio sobre JavaScript y AJAX		09/10/2012	18/10/2012
<b>3-2-1</b>	Recopilación de documentación	2	09/10/2012	09/10/2012
<b>3-2-2</b>	Lectura y realización de prácticas	20	09/10/2012	18/10/2012
<b>3-3</b>	Análisis breve del dominio		19/10/2012	19/10/2012

## Aplicación web para el inventario de presiones en ríos con CartoDB

PIÓ ALBERTO SIERRA RODRÍGUEZ

UOC - TFC - Curso 2012/2013 Semestre 1

<b>3-3-1</b>	Descarga del documento maestro de DMA	0,5	19/10/2012	19/10/2012
<b>3-3-2</b>	Lectura sobre DMA	1	19/10/2012	19/10/2012
<b>3-3-3</b>	Redacción de la Memoria	1,5	19/10/2012	19/10/2012
<b>3-4</b>	Estudio de CartoDB		20/10/2012	25/10/2012
<b>3-4-1</b>	CartoDB como usuario	3	20/10/2012	21/10/2012
<b>3-4-2</b>	Estudio de SQL API	4	21/10/2012	22/10/2012
<b>3-4-3</b>	Estudio de MAPs API	4	23/10/2012	24/10/2012
<b>3-4-4</b>	Redacción de la memoria	4	25/10/2012	25/10/2012
<b>3-5</b>	Elección de la librería para JavaScript		26/10/2012	02/11/2012
<b>3-5-1</b>	Estudio sobre GoogleMaps	2	26/10/2012	26/10/2012
<b>3-5-2</b>	Estudio sobre Leaflet	2	27/10/2012	27/10/2012
<b>3-5-3</b>	Estudio sobre OpenLayers	2	28/10/2012	28/10/2012
<b>3-5-4</b>	Primeros prototipos para escoger librería	6	29/10/2012	31/10/2012
<b>3-5-6</b>	Redacción de la Memoria	3	02/11/2012	02/11/2012
	Entregar borrador PEC2		02/11/2012	02/11/2012
<b>3-6</b>	Corregir borrador	5	03/11/2012	05/11/2012
	Entregar PEC2		06/11/2012	06/11/2012
<b>4</b>	PEC3		07/11/2012	11/12/2012
<b>4-1</b>	Creación de la base de datos geográfica		07/11/2012	14/11/2012
<b>4-1-1</b>	Diseño de la estructura	3	07/11/2012	08/11/2012
<b>4-1-2</b>	Introducción de datos de prueba	8	09/11/2012	13/11/2012
<b>4-1-3</b>	Redacción de la memoria	4	14/11/2012	14/11/2012
<b>4-2</b>	Diseño del visor SIG		15/11/2012	04/12/2012
<b>4-2-1</b>	Análisis del software	2	15/11/2012	15/11/2012
<b>4-2-2</b>	Diseño del software	4	16/11/2012	17/11/2012
<b>4-2-3</b>	Codificación del visor	24	18/11/2012	29/11/2012
<b>4-2-4</b>	Pruebas con datos	5	30/11/2012	02/12/2012
<b>4-2-5</b>	Documentación del software	2	03/12/2012	03/12/2012
<b>4-2-6</b>	Redacción de la memoria	4	04/12/2012	04/12/2012
	Entregar borrador de contenidos de PEC3		05/12/2012	07/12/2012
<b>4-3</b>	Corregir contenidos de PEC3	9	07/12/2012	11/12/2012
	Entregar PEC3		11/12/2012	11/12/2012
<b>5</b>	PEC4		12/12/2012	07/01/2013
<b>5-1</b>	Revisar la memoria		12/12/2012	17/12/2012
<b>5-1-1</b>	Unificar partes redactadas	5	12/12/2012	14/12/2012
<b>5-1-2</b>	Redactar conclusiones	2	15/12/2012	15/12/2012
<b>5-1-3</b>	Redactar resto de apartados	2	16/12/2012	16/12/2012



## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

5-2	Crear la presentación		17/12/2012	27/12/2012
5-2-1	Identificar y seleccionar contenidos	5	17/12/2012	19/12/2012
5-2-2	Realizar presentación	7	20/12/2012	23/12/2012
5-2-3	Grabar vídeo de presentación	6	26/12/2012	27/12/2012
	Entregar borrador de la memoria		28/12/2012	28/12/2012
5-3	Corregir borrador de la memoria	9	02/01/2013	07/01/2013
	Entregar Memoria		07/01/2013	07/01/2013
	Debate virtual		23/01/2013	25/01/2013
	Respuestas a preguntas		23/01/2013	25/01/2013
	Fin del proyecto (resultado)		29/01/2013	29/01/2013
	TOTAL (horas)	190		

### 1.3. PRODUCTOS OBTENIDOS

El trabajo de fin de carrera ha consistido en, con el objetivo de alcanzar los objetivos generales y específicos de su enunciado, la realización de los siguientes productos.

#### 1.3.1. MEMORIA DEL TRABAJO

Se trata de este documento. Para su diseño se ha utilizado de forma fundamental las recomendaciones del documento "Presentación de documentos y elaboración de presentaciones." (Montagut) disponible en el aula de la UOC. En la medida de lo posible, se han intentado seguir todas las pautas de estilo que marca.

En lo referente a la redacción del contenido, se ha utilizado como guía principal otro de los documentos disponibles en el aula, "Redacción de textos científico técnicos." (Higueras, y otros).

Se consideró en un principio la posibilidad de redactar la memoria con LyX, pero finalmente se optó por utilizar Word, más conocido por el autor, para no tener que añadir una más a las competencias (ya varias) a desarrollar durante el trabajo.

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

### 1.3.2. PRESENTACIÓN EN VÍDEO

Para presentar el trabajo se realiza una presentación en PowerPoint que es grabada en vídeo con la correspondiente locución, además incluye una demostración del resultado final de la aplicación. (Ilustración 1)



ILUSTRACIÓN 1: PRESENTACIÓN DEL TRABAJO

### 1.3.3. APLICACIÓN WEB

Por último, se ha producido una aplicación web, cuyos fuentes se encuentran disponibles como parte de la entrega del proyecto, y que también puede ser directamente accesible online en la dirección <http://labrujula.org/sigweb> (Ilustración 2).

Además de los archivos incluidos, la aplicación trabaja utilizando una cuenta gratuita de CartoDB, como se describe más adelante. No se incluye actualmente un script de instalación para crear las tablas de dicha cuenta, aunque sí se describe cuales son los

# Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

requisitos y como modificar el usuario y la API Key del código para hacerlo funcionar con una cuenta distinta.

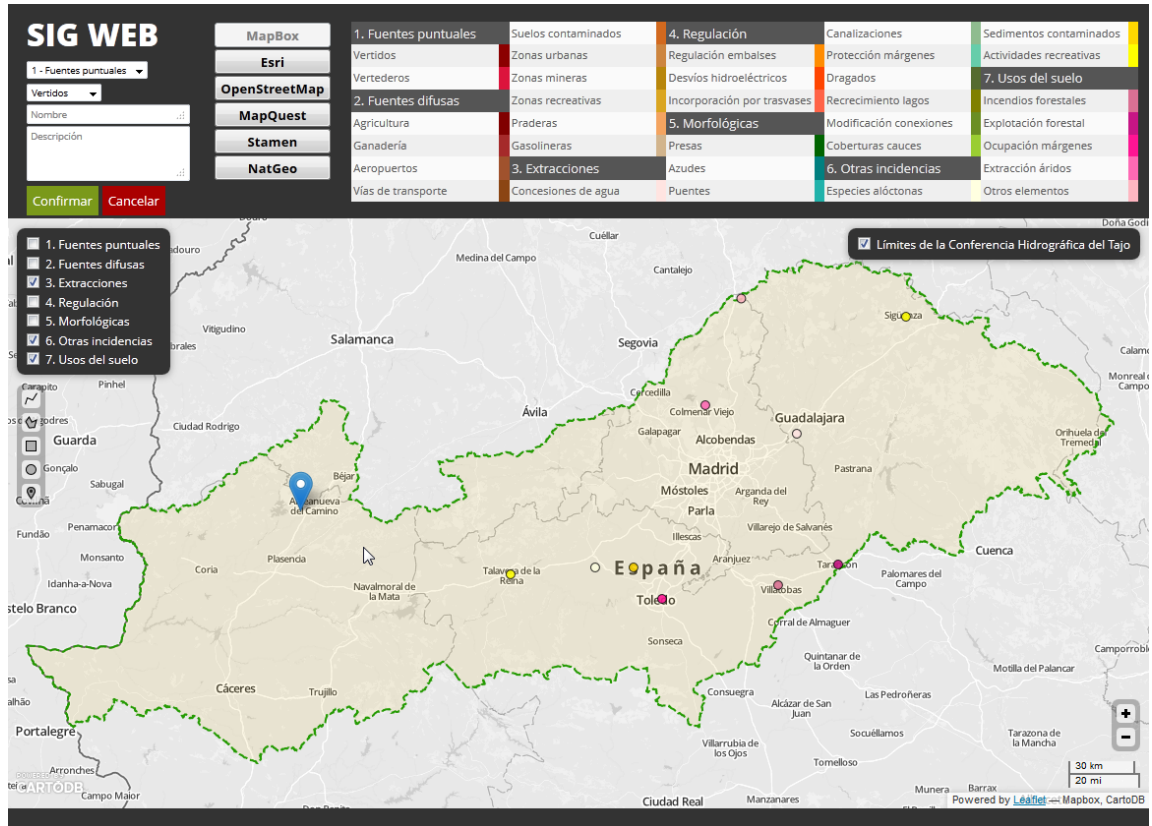


ILUSTRACIÓN 2: INTERFAZ WEB DE LA APLICACIÓN (SIG WEB)

## 2. ANÁLISIS DEL DOMINIO

Para la realización del trabajo se escoge la Confederación Hidrográfica del Tajo, de la que disponemos de abundante información<sup>1</sup>. Con ello hemos definido el ámbito geográfico. Respecto a las presiones, objeto del trabajo, estas vienen definidas en el "Manual para la identificación de las presiones y análisis del impacto en aguas superficiales."<sup>2</sup> (Ilustración 3) elaborado por la Dirección General del Agua del Ministerio de Medio Ambiente.

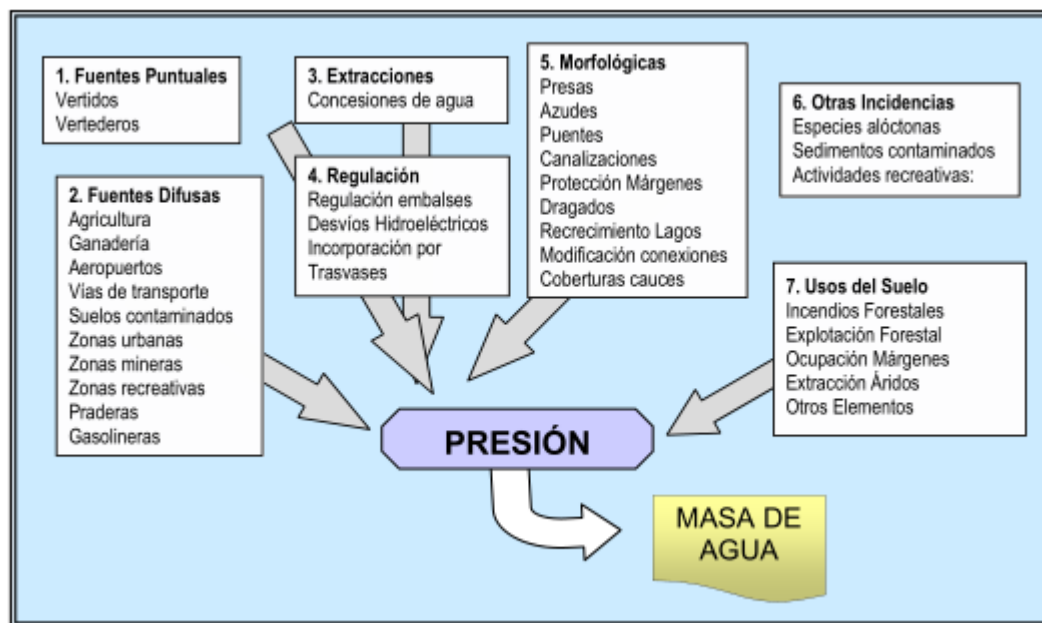


ILUSTRACIÓN 3 GRUPOS Y TIPOS DE PRESIONES<sup>3</sup>

Era un requisito que éstas se puedan identificar por tipo, así pues cada presión viene definida, al menos, por sus coordenadas, su grupo y su tipo. Además es interesante incluir informaciones como el nombre, una descripción o la fecha de alta.

<sup>1</sup> <http://www.chtajo.es/Servicios/Paginas/DescargaDCapas.aspx>

<sup>2</sup> [http://www.magrama.gob.es/es/agua/publicaciones/impress\\_tcm7-29158.pdf](http://www.magrama.gob.es/es/agua/publicaciones/impress_tcm7-29158.pdf)

<sup>3</sup> Manual para la identificación de las presiones y análisis del impacto en aguas superficiales. (Dirección del Agua, Ministerio de Medio Ambiente)

### 3. SIG EN UN ENTORNO CLOUD

A continuación se expondrá el concepto de Cloud y se dará cuenta de cómo afecta a los sistemas SIG y como ellos pueden evolucionar en dicho contexto.

Los Sistemas de Información Geográfica son uno de los campos más beneficiados de lo que podríamos denominar la “revolución Cloud”. La disponibilidad de recursos y datos, junto con el aumento del ancho de banda y la creación de numerosos APIs para interconectar servicios, hacen que sea posible acercar a usuarios desarrollos que antes necesitaban de un gran espacio de almacenamiento, datos que no siempre estaban disponibles, o utilizar entornos complejos de configurar como el software SIG para servidores.

En este contexto, la realización de “mashups” en los que integrar varios recursos Cloud para ofrecer una aplicación SIG online se convierte no solo en algo posible sino también en una opción adecuada para muchos casos.

#### 3.1. DEFINICIÓN DE ENTORNO CLOUD

Según la definición del National Institute of Standards and Technology (U.S. Department of Commerce), existen unas características esenciales que definen un servicio de Cloud Computing:

- **Autoservicio a discreción:** Un usuario puede unilateralmente provisionar recursos automáticamente sin la intervención del proveedor.
- **Acceso con gran ancho de banda:** Los recursos están disponibles en la red.
- **Bolsa de recursos:** Los recursos del proveedor se encuentran agrupados para dar servicio a varios usuarios de acuerdo a las necesidades y a la demanda.
- **Elasticidad veloz:** Los recursos pueden ser provisionados rápida y elásticamente, de forma que escalen adecuadamente.
- **Medición del servicio:** Los sistemas Cloud han de contar con métodos de medición del servicio y algún tipo de abstracción para evaluar los recursos consumidos.

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

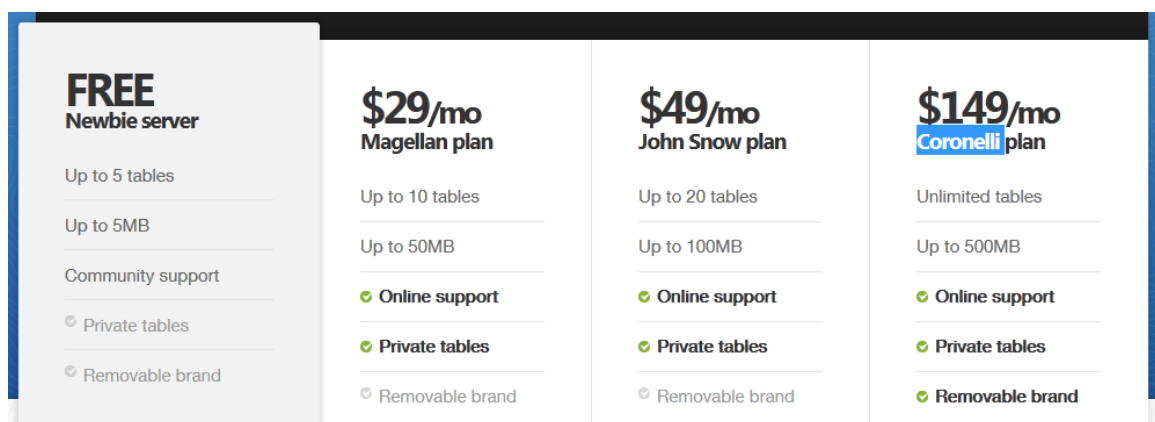
UOC – TFC – Curso 2012/2013 Semestre 1

Además, se propone una definición de los tipos de servicio ofertados, según el modelo:

- **Software como Servicio** (*Software as a Service, SaaS*): La capacidad de proporcionar aplicaciones que funcionan sobre un entorno Cloud a los usuarios.
- **Plataforma como Servicio** (*Platform as a Service, PaaS*): La capacidad de proporcionar una plataforma formada por librerías, lenguajes, herramientas, etc., sobre la que el usuario puede crear y desplegar sus propias aplicaciones.
- **Infraestructura como Servicio** (*Infrastructure as a Service, IaaS*): La capacidad de proveer al usuario con recursos de computación, almacenamiento y comunicación sobre los que el usuario pueda instalar el software que desee.

### 3.2. CARTODB. UN SIG BASADO EN CLOUD.

CartoDB es un SIG cloud que brinda distintas funcionalidades, tanto para usuarios como para desarrolladores. Está desarrollado por Vizzuality<sup>1</sup>, una empresa fundada por Javier de la Torre y Sergio Álvarez Leiva, con sede en Nueva York y Madrid. A nivel comercial ofrece diversos planes, incluyendo uno gratuito limitado a 5 tablas que ocupen hasta 5 MB, que será el utilizado en este proyecto (Ilustración 4).



FREE Newbie server	\$29/mo Magellan plan	\$49/mo John Snow plan	\$149/mo Coronelli plan
Up to 5 tables	Up to 10 tables	Up to 20 tables	Unlimited tables
Up to 5MB	Up to 50MB	Up to 100MB	Up to 500MB
Community support	Online support	Online support	Online support
Private tables	Private tables	Private tables	Private tables
Removable brand	Removable brand	Removable brand	Removable brand

ILUSTRACIÓN 4: PLANES DE USO DE CARTODB

<sup>1</sup> <http://vizzuality.com/>

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

CartoDB es una base de datos geoespacial con licencia Open Source. Utiliza en su infraestructura PostGIS, Node.js, Mapnik, Redis y Varnish. Por una parte CartoDB proporciona una interfaz para importar y visualizar datos geoespaciales, pero además también proporciona una serie de APIs para acceder a los datos mediante sentencias SQL y *Tiles* (teselas). Este carácter dual hace que CartoDB pueda ser utilizado para múltiples tipos de aplicaciones y visualizaciones online, desde aplicaciones móviles, visualizaciones online, portales geoespaciales y prácticamente cualquier tipo de aplicación geoespacial.

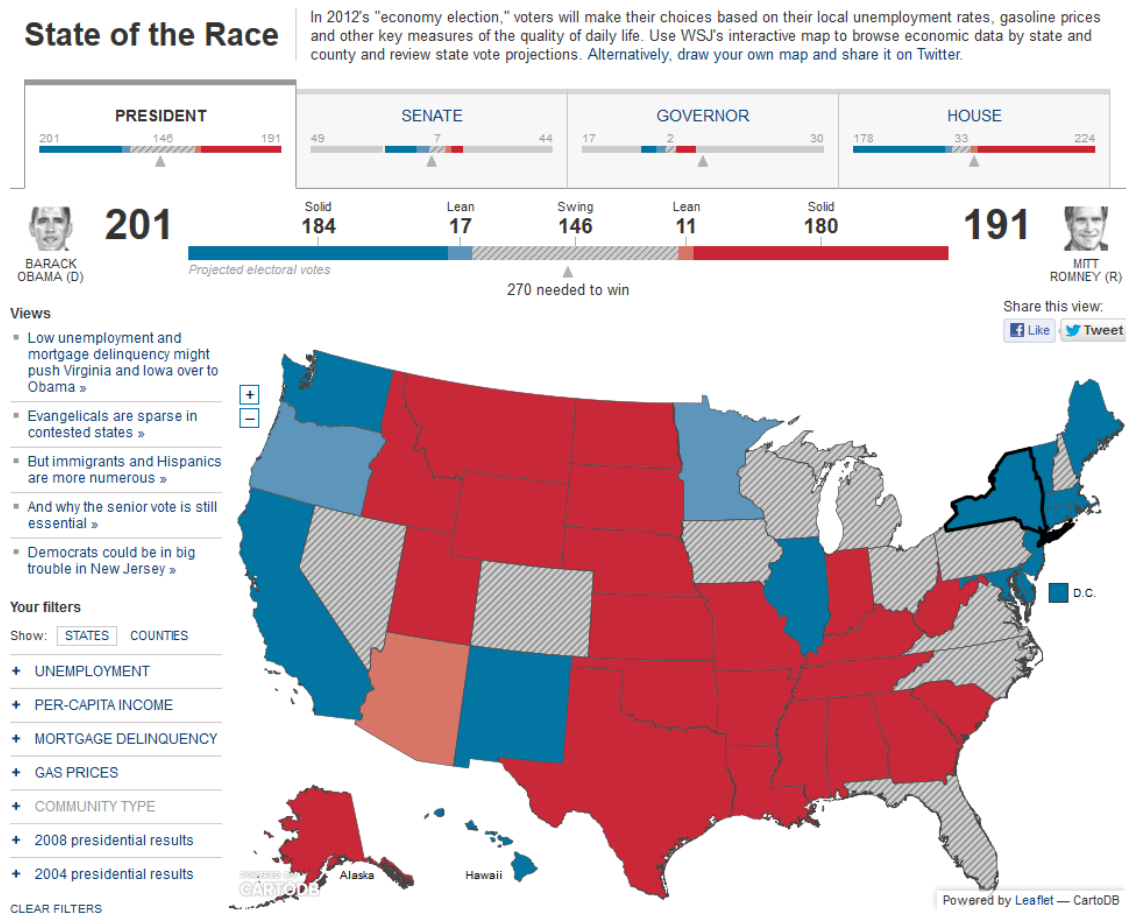


ILUSTRACIÓN 5: EJEMPLO DE USO DE CARTODB POR THE WALL STREET JOURNAL

La versión de la base de datos utilizada en la última versión (CartoDB 2.0) es PostGIS 2.0 y por tanto soporta datos rasterizados, búsquedas por proximidad basadas en índices y

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

nuevas funciones para trabajar utilizando GeoJSON. Las posibilidades para el desarrollo de aplicaciones web que consultan datos rasterizados mediante geometrías son muy amplias. CartoDB aprovecha la tecnología VisualRaster para la visualización de datos rasterizados en el navegador utilizando HTML5.

CartoDB puede ser utilizado con un lenguaje de programación a través de sus APIs. Existen librerías para simplificar la conexión en Ruby, PHP, Python y JavaScript, pero sirve cualquier otro lenguaje que pueda realizar conexiones HTTP.

CartoDB es utilizado actualmente por diversas empresas e instituciones, como por ejemplo The Guardian, ONU, NASA, RTVE, y las universidades de Oxford, Berkley o Yale. (Ilustración 5)

Entre las principales características de CartoDB podemos destacar que tiene un activo grupo de desarrolladores que aportan documentación y ejemplos<sup>1</sup>. El carácter abierto de su API hace que continuamente se estén desarrollando nuevas integraciones y aumentando las capacidades del conjunto en general, con nuevas librerías<sup>2</sup> que se ponen a disposición de los desarrolladores.

### 3.2.1. CARACTERÍSTICAS PRINCIPALES DE CARTODB

Entre sus características podemos destacar:

- *Diseño de mapas:* Para sus capas de datos, es posible utilizar CartoCSS para editar fácilmente el formato y la apariencia con la que se generarán.
- *Integración con otros servicios cartográficos (Google Maps, MapBox):* CartoDB produce las capas de datos, y para la capa de mapa utiliza GoogleMaps y, desde la versión 2.0 MapBox. Estos mapas incluyen las funciones básicas de zoom, desplazamiento, etc.
- *Integración con otras librerías:* CartoDB cuenta ya con varias librerías a su alrededor que permiten extender su uso o integrar otros servicios.
- *Geocodificación:* Es posible obtener información geográfica a partir de otros elementos distintos de las coordenadas.

---

<sup>1</sup> <http://blog.cartodb.com/>

<sup>2</sup> <https://github.com/Vizzuality>



## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

- *Capacidad de importar datos fácilmente:* CartoDB permite introducir datos directamente en las tablas a partir de su panel, añadir datos vía SQL o lectura desde URLs, pero también se pueden importar colecciones de datos directamente en múltiples formatos (Ilustración 6)

.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

\* ESRI Shape Files require at least three separate files. CartoDB requires that you include all of the files in a single compressed archive.

### ILUSTRACIÓN 6: FORMATOS DE ARCHIVO DE DATOS ADMITIDOS POR CARTODB

- *Realizar peticiones SQL con componentes espaciales:* Gracias al uso de PostGIS, CartoDB permite consultar y combinar conjuntos de datos utilizando los datos geoespaciales para realizar la combinación.
- *Tablas públicas y privadas:* Como corresponde a un servicio cloud, CartoDB permite diferenciar entre un uso de las tablas público, o un uso privado.

Para la realización de nuestro trabajo nos interesan especialmente la integración con otras librerías, la capacidad de realizar peticiones SQL, y el uso de tablas públicas y privadas.

En la Ilustración 7 puede verse el alcance de las funcionalidades de CartoDB en comparación con otros servicios similares según el estudio de Richard Donohue y Timothy Wallace: “Keeping pace with emerging web mapping technologies” (Wallace).

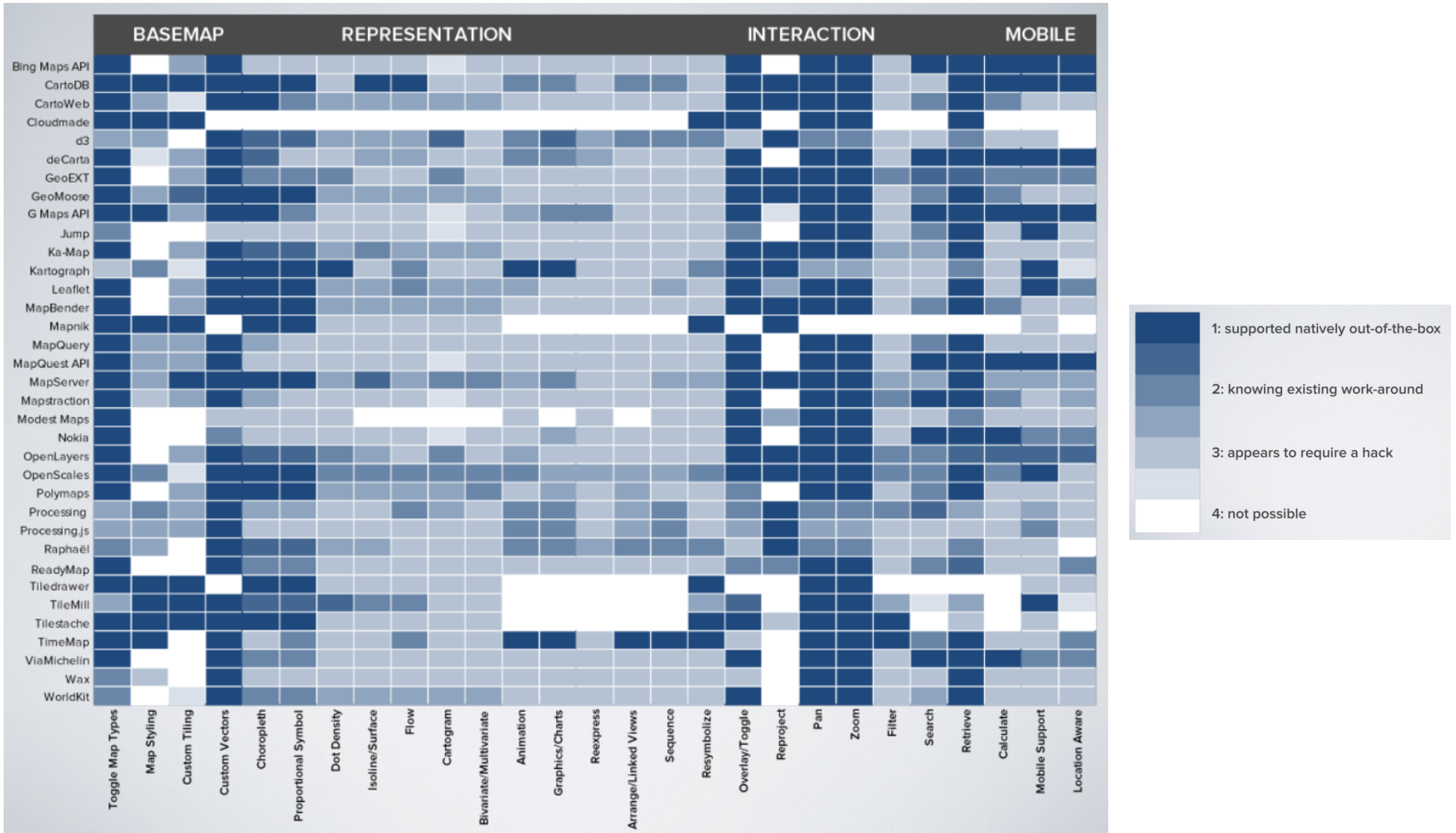


ILUSTRACIÓN 7: FUNCIONES DE CARTODB COMPARADAS CON SERVICIOS SIMILARES.

### 3.2.2. ARQUITECTURA DE CARTODB

La estructura interna de CartoDB puede verse en la Ilustración 8.

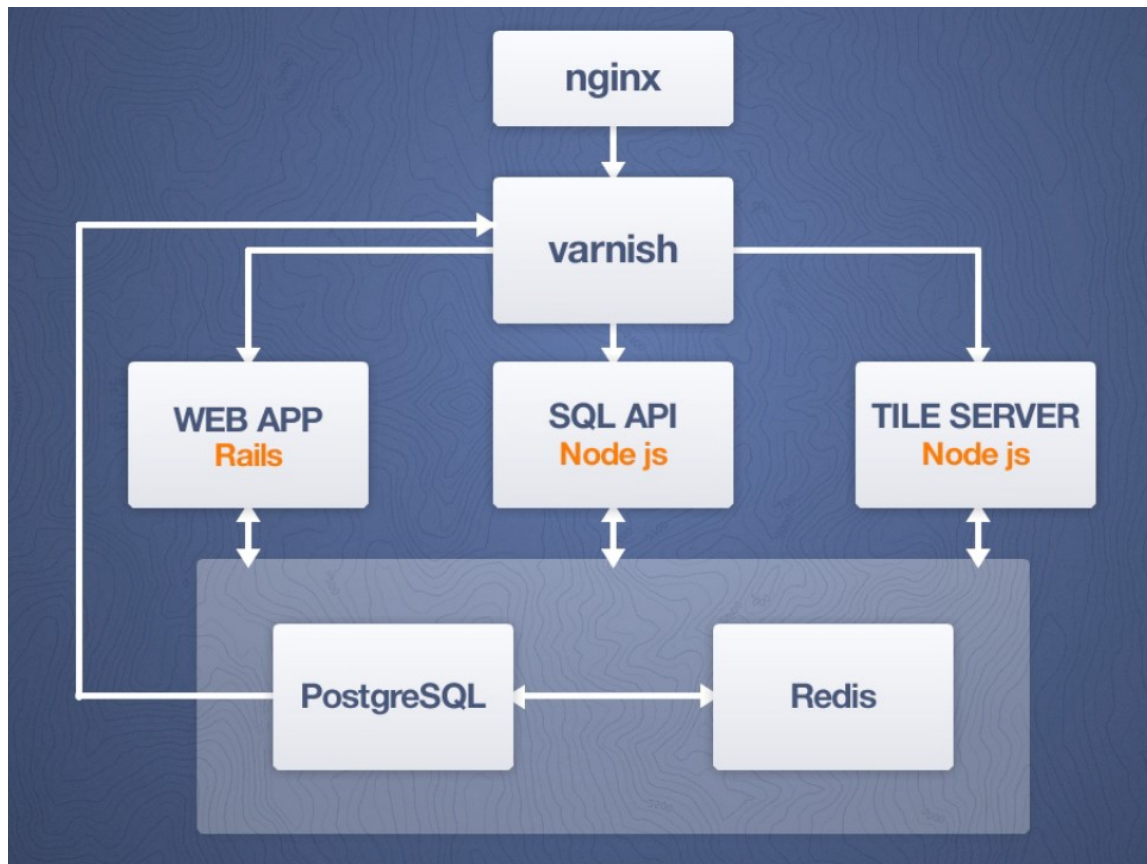


ILUSTRACIÓN 8: ESTRUCTURA INTERNA DE CARTODB<sup>1</sup>

Además, se puede diferenciar entre varios componentes distintos que integran el conjunto de CartoDB:

- Un interfaz de usuario (Dashboard) en el que es posible añadir, visualizar, editar y exportar los datos geoespaciales.

---

<sup>1</sup> [http://www.sigte.udg.edu/jornadassiglibre2012/uploads/presentaciones\\_12/p46.pdf](http://www.sigte.udg.edu/jornadassiglibre2012/uploads/presentaciones_12/p46.pdf)

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

- Una base de datos geoespacial construida sobre PostgreSQL y PostGIS 2.0.
- Una API SQL para realizar peticiones SQL sobre HTTP, que ofrece resultados en formatos KML y GeoJSON.
- Un creador de *tiles* para mapas con SQL y que utiliza CartoCSS para darles formato.
- Un sistema de autenticación basado en OAuth y a través de API key.

CartoDB está basado en una implementación de PostGIS<sup>1</sup>, sobre la que se han añadido interfaces para acceder directamente a las bases de datos mediante la web, así como para mostrar sus contenidos en mapas. Además se han implementado varias librerías y APIs, los más importantes de los cuales son los que permiten realizar directamente consultas SQL (SQL API) y el que permite generar mapas (MAPS API).

Para cada tabla CartoDB crea un campo de geometría llamado "the\_geom" en el que almacena en proyección WGS 84 (EPSG:4326) los datos geoespaciales en formato well-known binary<sup>2</sup> (están implementadas funciones de conversión a, por ejemplo, GeoJSON). Aunque PostGIS admite más geometrías, CartoDB convierte todas a uno de estos tres tipos: POINTS, MULTILINESTRINGS, o MULTIPOLYGONS. Por lo demás, ofrece directamente a través del interfaz de usuario la posibilidad de utilizar campos NUMBER, STRING y DATE, pero en realidad mediante el uso de SQL se puede acceder a cualquier tipo de datos soportados por PostGIS<sup>3</sup>.

He aquí la lista completa de componentes utilizados en CartoDB:

- Ubuntu 10.04
- Postgres 9.1.x
- Redis 2.2+
- Ruby 1.9.2
- NodeJS 0.8.x
- CartoDB-SQL-API
- GEOS 3.3.4
- GDAL 1.9.0
- PostGIS 2.0.x
- Mapnik 2.1.1
- Windshaft-cartodb
- Varnish 2.1+

---

<sup>1</sup> <http://postgis.refractor.net/>

<sup>2</sup> [http://edndoc.esri.com/arcswde/9.0/general\\_topics/wkb\\_representation.htm](http://edndoc.esri.com/arcswde/9.0/general_topics/wkb_representation.htm)

<sup>3</sup> <http://developers.cartodb.com/documentation/analysis.html>

### 3.3. VENTAJAS E INCONVENIENTES DE CARTODB COMO ENTORNO CLOUD

Existen varios modelos de despliegue Cloud que pueden ser utilizados con éxito para despliegues de soluciones SIG. Está claro que por un lado se pueden utilizar recursos de infraestructura, el modelo IaaS, simplemente para desplegar la aplicación, el uso realizado de servicios de CDN (Content Delivery Networks) para disponer de algunas librerías en el presente trabajo es un ejemplo de ello. Sin embargo ese es un uso más genérico y menos específico, por lo que a continuación destacaremos las ventajas de una solución SIG como CartoDB, que funciona tanto como modelo SaaS como PaaS.

#### 3.3.1. VENTAJAS

- *Elasticidad/Escalabilidad:* Al menos en teoría, utilizar un servicio Cloud permite eliminar la variable del dimensionamiento a la hora de iniciar el proyecto o realizar el despliegue. Se puede empezar con pocos recursos y aumentar estos paulatinamente de forma transparente a medida que van siendo necesarios.
- *Coste total de propiedad (TCO) reducido:* Al no tener que invertir en infraestructura dedicada, licencias ni mantenimiento interno, el TCO de una solución Cloud suele ser menor que el de otras opciones, aunque es necesario estudiar el tipo de licencia de servicio en cada caso.
- *Facilidad para compartir la información:* Desplegar sobre un servicio Cloud permite compartir la información más fácilmente, en caso de que esto sea necesario, debido a la propia naturaleza online del servicio.
- *Alta disponibilidad:* Los servicios Cloud cuentan (o deberían) con sistemas de redundancia y alta disponibilidad, de forma que esas funciones van incluidas en el servicio y no es necesario implementarlas como si llevásemos a cabo una instalación interna.
- *Interconexión con otros servicios:* Una de las ventajas más interesantes es la facilidad que otorgan los servicios web para interconectar unos con otros. En este trabajo nos aprovecharemos de esta ventaja para utilizar varios de ellos en el desarrollo de la aplicación.

## **Aplicación web para el inventario de presiones en ríos con CartoDB**

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

### **3.3.2. INCONVENIENTES**

- *Seguridad/Privacidad:* Los datos se encuentran en servidores a los que no tenemos acceso directo, con lo que no hay garantía de que podamos utilizar dichos datos. Además, al transmitirse a través de Internet estos pueden ser interceptados si no utilizamos los protocolos de encriptación adecuados.
- *Control sobre el servicio:* No podemos garantizar al 100% que la empresa o entidad que presta el servicio seguirá haciéndolo ininterrumpidamente. Por desgracia son ya varios los casos de proveedores de servicios Cloud que han dejado de funcionar.

## 4. DISEÑO DEL SIG WEB

En este apartado se describe todo el proceso de análisis, diseño y codificación de la aplicación. Dado el desconocimiento previo del autor sobre muchas de las tecnologías, así como el limitado tiempo disponible, se optó por trabajar siguiendo los principios de metodologías de tipo ágil, para conseguir que se tratase de un proceso incremental. Al tratarse de una labor no en equipo no tenía sentido el utilizar plenamente ninguna en concreto, tipo Scrum o XP.

### 4.1. ANÁLISIS PARA EL DISEÑO DEL SIG WEB

A continuación se presenta el trabajo de análisis realizado para diseñar la aplicación. Algunos de estos análisis resultaron más complicados que otros debido al desconocimiento previo del autor de las clases y capacidades de muchas de las librerías, lo que hizo que el diseño fuese modificado constantemente a medida que surgían nuevos retos o se encontraban soluciones alternativas.

#### 4.1.1. CASOS DE USO

De acuerdo al los requisitos de la aplicación, no se identifica otro actor distinto del usuario de la aplicación. En principio no se diferencian roles, ni existen sistemas de autenticación que los separen, o actores externos. Así pues quedan identificados los siguientes casos de uso (Ilustración 9) que se describen a continuación.

# Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

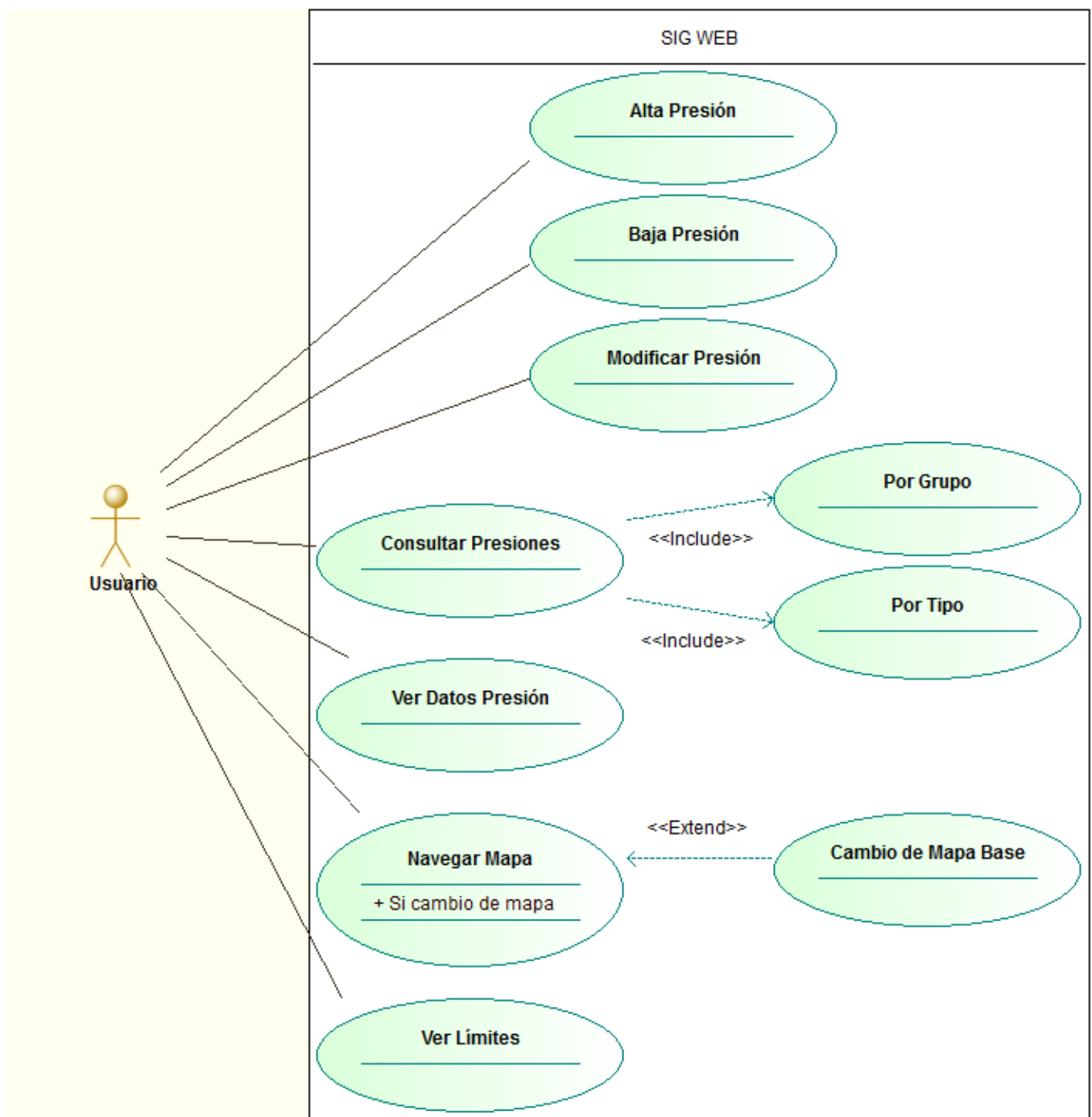


ILUSTRACIÓN 9: CASOS DE USO



## **Aplicación web para el inventario de presiones en ríos con CartoDB**

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

### **Alta Presión**

El usuario puede dar de alta una presión. Para ello dibujará un área en el mapa (punto o polilínea) tras lo cual introducirá en un formulario unos datos obligatorios (Tipo y Grupo) y otros opcionales (Nombre y Descripción) Si el usuario acepta, la base de datos se actualizará con los mismos.

### **Baja Presión**

Una vez seleccionada una presión en el mapa, el usuario puede darla de baja, tras lo que será eliminada de la base de datos y de la vista actual del mapa.

### **Modificar Presión**

El usuario puede modificar la geometría y el resto de campos de la presión, tras lo que se actualizará la base de datos.

### **Consultar Presiones**

El usuario puede, mediante unos controles, seleccionar el tipo de presiones que verá en el mapa. En concreto podrá ver todas y activar o desactivar conjuntos de ellas definidos por grupos o tipos.

### **Ver Datos Presión**

Al situar el puntero sobre una presión, aparecerá un popup con toda la información de la misma.

### **Navegar Mapa**

El sistema incluirá los controles típicos de navegación, zoom y desplazamiento. Como la región está claramente definida, no se considera necesario incluir sistemas de búsqueda o de entrada directa de latitud y longitud. Además el usuario podrá cambiar el mapa base entre una selección de ellos.

### **Ver Límites**

El usuario podrá activar o desactivar una capa con los límites de la conferencia hidrográfica en cuestión.

# Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

## 4.1.2. DIAGRAMA DE SECUENCIA

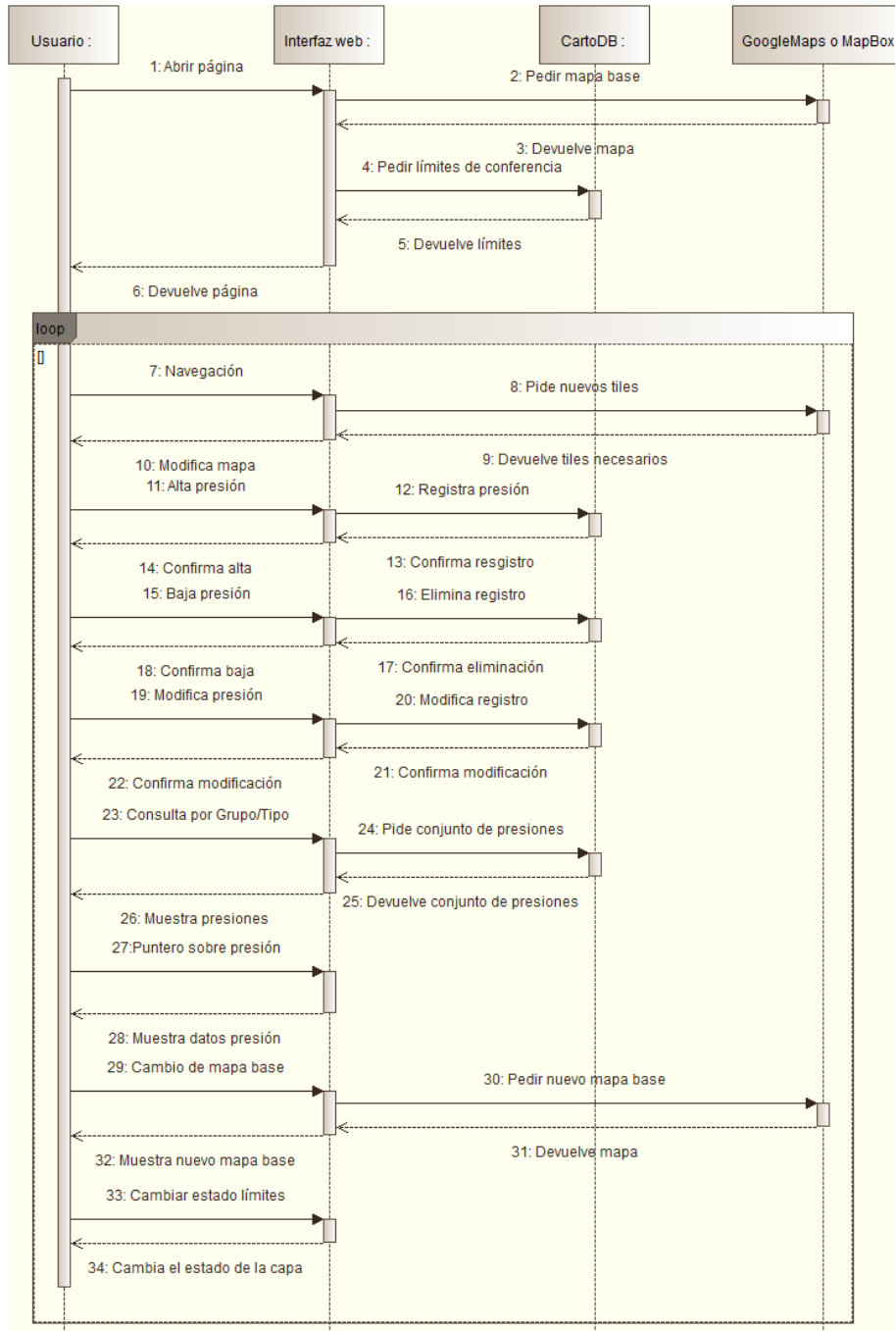


ILUSTRACIÓN 10: DIAGRAMA DE SECUENCIA

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

### 4.2. DISEÑO DE LA BASE DE DATOS

Se ha realizado un estudio para crear las bases de datos necesarias que mantengan la información que vamos a utilizar. Con los requisitos de la aplicación, se identifican de forma básica las siguientes entidades (Ilustración 11).

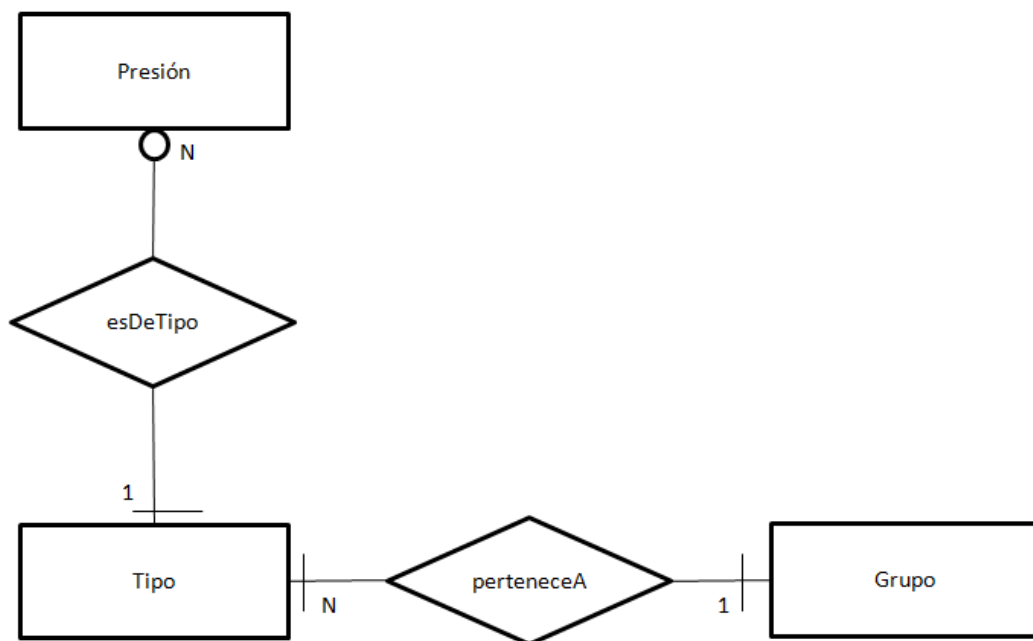


ILUSTRACIÓN 11: MODELO ER DE LA BASE DE DATOS

Por supuesto, se podría ampliar el modelo para incluir nuevos requisitos, de igual forma que hemos de adaptarlo a las limitaciones que tenemos al utilizar CartoDB como base de datos.

La bases de datos de CartoDB, aunque basadas en PostGIS, ofrecen un número limitado de tipos de datos a utilizar de forma directa, aunque se pueden crear más con SQL:

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

- **Geometry:** por defecto en una columna llamada `the_geom` para cada base de datos. En ella, de los diversos tipos de geometría que ofrece PostGIS, CartoDB solo almacena POINTS, MULTISTRINGS y MULTIPOLIGONS, y convierte a uno de estos formatos cualquier otra geometría. Es importante destacar que no solo la columna `the_geom` utiliza solo uno de estos tres tipos, sino que además solo utiliza uno de ellos (el primer dato introducido define de qué tipo será la columna). Esto ha resultado de importancia, ya que no descubrió este hecho hasta que el diseño estaba muy avanzado trabajando solo con puntos. En principio para una aplicación como esta la mejor solución sería utilizar MULTIPOLIGONS, que debería ser la que más casos posibles englobe con un solo tipo.
- **String:** en UTF-8. Sin límite de tamaño (en realidad se trata de un *text field* de PostGIS)
- **Number:** Para cualquier tipo numérico (tipo *numeric* de PostGIS).
- **Date:** Cada base de datos incluye directamente los campos `created_at` y `updated_at`, con la fecha y hora correspondiente basada en el reloj de su servidor (basado en GMT+1, lo que resulta conveniente para este trabajo)

Como no ha sido necesario realizar operaciones numéricas en la base de datos, y en cambio es necesario propagar todos los datos como string en las peticiones a la API SQL de CartoDB, se ha optado por utilizar como tipo String incluso para la codificación de los grupos, aunque estos se han codificado por el número correspondiente numerados del 1 al 7.

Teniendo esto en cuenta, se identifican las siguientes necesidades para el trabajo.

- Una tabla que almacene información sobre las presiones.
- Una tabla que almacene información sobre los límites de conferencia.

Existen otras posibilidades basadas en el modelo ER tal y como fue definido, pero debido a la imposibilidad de incluir directamente disparadores en la base de datos, y teniendo en cuenta el número limitado de tablas que se pueden utilizar en la cuenta gratuita, se ha optado por reducir las tablas a estas dos e incluir en la lógica de la aplicación las relaciones entre tipos y grupos.

Estas tablas representan el mínimo imprescindible para poder cumplir los requisitos exigidos, pero a su vez permiten realizar sencillas ampliaciones por medio de otras tablas (Ilustración 12).

Debido a que CartoDB genera automáticamente un campo clave (`cartodb_id`) se ha considerado que este es suficiente para tal fin, ya que además en los requisitos no se

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

identifica ningún otro posible campo clave (los nombres de las presiones no son obligatorios, la geometría puede coincidir, etc.)

public.limite_conferencia		
<input checked="" type="radio"/> cartodb_id	numeric	« pk nn »
<input checked="" type="radio"/> the_geom	geometry(GEOMETRYCOLLECTION, 4326)	« nn »
<input type="radio"/> area_km2	numeric	
<input checked="" type="radio"/> nombre	text	« nn »
<input checked="" type="radio"/> created_at	date	« nn »
<input checked="" type="radio"/> updated_at	date	« nn »

public.presiones		
<input checked="" type="radio"/> cartodb_id	numeric	« pk nn »
<input checked="" type="radio"/> the_geom	geometry(GEOMETRYCOLLECTION, 4326)	« nn »
<input type="radio"/> descripcion	text	
<input checked="" type="radio"/> grupo	text	« nn »
<input type="radio"/> nombre	text	
<input checked="" type="radio"/> tipo	text	« nn »
<input checked="" type="radio"/> created_at	date	« nn »
<input checked="" type="radio"/> updated_at	date	« nn »

ILUSTRACIÓN 12: TABLAS DE LA BASE DE DATOS

### 4.3. SELECCIÓN DE UNA LIBRERÍA DE MAPAS

Para la realización del trabajo se ha pedido la evaluación de tres librerías de mapas para decidir cuál es la más conveniente para utilizar. CartoDB ofrece funcionalidades de base de datos SIG, y una API básica de mapas para la visualización. CartoDB nos servirá para almacenar la información y para realizar consultas, pero para la visualización necesitamos una librería de mapas que nos proporcione más funcionalidad para atender a los requisitos del Trabajo.

Una de las herramientas principales a la hora de evaluar las librerías, ha sido examinar como estas eran utilizadas en los distintos ejemplos de uso de CartoDB que incluye tanto en su web (Ilustración 13) como en su página de GitHub.

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

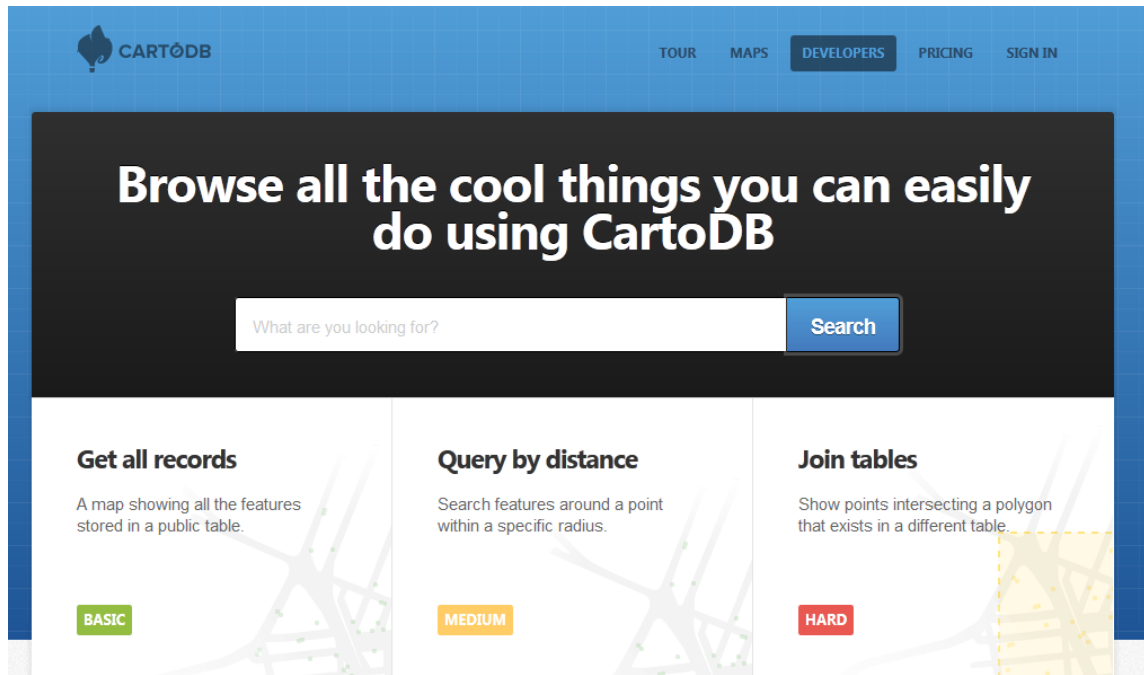


ILUSTRACIÓN 13: EJEMPLOS DE CARTODB

Para escoger la librería definimos una serie de criterios que serán los que nos permitan realizar la evaluación deseada:

- *Licencia*: el tipo de licencia de la librería.
- *Requisitos funcionales*: entre otros, presentación de mapas, posibilidad de modificar parámetros de la presentación de la cartografía, uso de capas con manejo independiente y controles para definir topologías mediante herramientas de dibujo.
- *Integración con CartoDB*: si ya hay un trabajo previo de integración con CartoDB, así como ejemplos.
- *Documentación*: la calidad y cantidad de la documentación del API.

Además de estudiar las librerías de forma independiente, se ha hecho uso de trabajos previos de comparación de diversas fuentes. En la Ilustración 14 puede verse la valoración de estas librerías según uno de los estudios(Wallace).

# Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

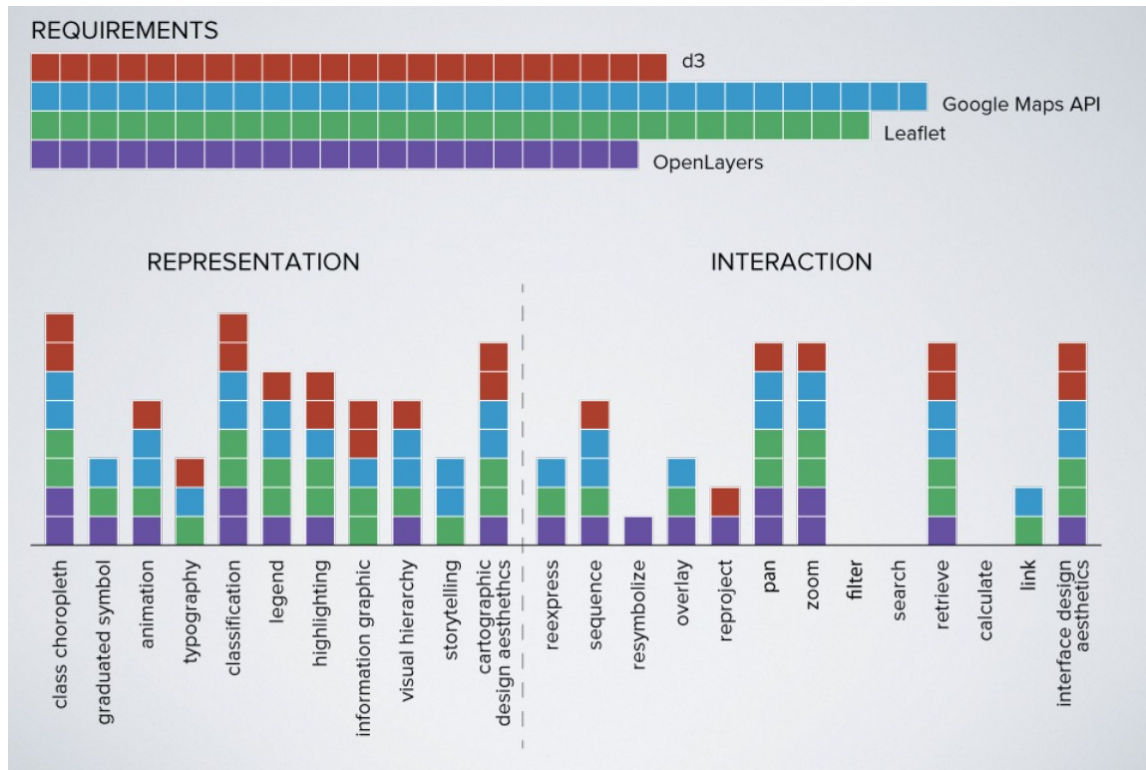


ILUSTRACIÓN 14: VALORACIÓN DE FUNCIONALIDADES DE LAS LIBRERÍAS DE MAPAS<sup>1</sup>.

## 4.3.1. GOOGLE MAPS

La versión 3 de Google Maps JavaScript API es una librería madura, con amplia documentación y con funcionalidades avanzadas. Cuenta con una gran comunidad de usuarios y, en su última versión, permite utilizar otros mapas como base además de los propios de Google.

- *Licencia:* aunque se puede utilizar gratuitamente siempre que no sea para usos comerciales, Google Maps no cuenta con una licencia libre (Open Source). Esto se convierte en uno de sus principales problemas comparado con el resto de opciones. Aunque serviría para el desarrollo que vamos a

<sup>1</sup> <http://geography.wisc.edu/cartography/slides/emerging-web-mapping-donohue-et-al-nacis2012.pdf>

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

hacer para el trabajo, no podríamos utilizarlo luego sin conseguir una licencia en caso de que quisiésemos aplicarlo comercialmente<sup>1</sup>. Además, la licencia gratuita tiene una serie de limitaciones técnicas. (Ilustración 15)

Features	Maps API	Maps API for Business
Street View	✓	✓
Geocoding Web Service	2500 requests per day	100 000 requests per day
Directions Web Service	2500 requests per day with 10 waypoints per request	100 000 requests per day with 23 waypoints per request
Distance Matrix Web Service	100 elements per query 100 elements per 10 seconds 2500 elements per day	625 elements per query 1000 elements per 10 seconds 100 000 elements per day
Elevation Web Service	2500 requests per day with 25 000 samples per day	100 000 requests per day with 1 000 000 samples per day
Static Maps API maximum resolution	640 x 640	2048 x 2048
Static Map API maximum scale	2X	4X
Street View Image API maximum resolution	640 x 640	2048 x 2048

### ILUSTRACIÓN 15: DIFERENCIAS TÉCNICAS EN LICENCIAS DE GOOGLE MAPS<sup>2</sup>

- *Funcionalidades:* un estudio preliminar sugiere que la librería cuenta con todo lo necesario para llevar a cabo el trabajo. A destacar las herramientas de dibujo de capas, necesarias para dar de alta las presiones interactivamente de forma directa sobre el mapa.
- *Integración con CartoDB:* existen suficientes ejemplos<sup>3</sup> así como librerías<sup>4</sup> propias para la integración desarrolladas por Vizzuality (CartoDB Google Maps v3 library 0.54.) Una ventaja adicional es que, por defecto, la mayoría de ejemplos de CartoDB utilizan Google Maps.
- *Documentación:* Google Maps cuenta con una extensa documentación, tanto en inglés como en español. Es necesario tener en cuenta que hay varios cambios en la versión 3 que hacen obsoletos algunos ejemplos y procesos anteriores.

---

<sup>1</sup> [https://developers.google.com/maps/faq?hl=es#tos\\_commercial](https://developers.google.com/maps/faq?hl=es#tos_commercial)

<sup>2</sup> <https://developers.google.com/maps/licensing>

<sup>3</sup> <http://vizzuality.github.com/cartodb-gmapsv3/examples/>

<sup>4</sup> <https://github.com/Vizzuality/cartodb-gmapsv3/>



### 4.3.2. LEAFLET

Se trata de una librería OpenSource desarrollada por principalmente por Vladimir Agafinkin, de CloudMade. Aunque no tiene tanto tiempo como las otras librerías, su carácter Open Source y el hecho de que sea bastante ligera y manejable hacen que tenga bastante apoyo y esté avanzando adecuadamente. Utiliza un sistema de plugins para extender sus capacidades, entre los que se encontraron los necesarios para proveer la funcionalidad requerida. Está centrada sobre todo en ser funcional también para dispositivos móviles, lo que representa una ventaja si queremos tenerlos en cuenta.

- *Licencia:* Leaflet es Open Source (licencia BSD) y no tendríamos ningún problema para su uso.
- *Funcionalidades:* aunque directamente no incluye herramientas de dibujo, un análisis posterior comprobó que han sido añadidas como plugin (Leaflet.draw<sup>1</sup>). Permite crear *layers* a partir de datos en formato GeoJSON.
- *Integración con CartoDB:* al igual que ocurre con Google Maps, existen suficientes ejemplos así como librerías<sup>2</sup> propias para la integración desarrolladas por Vizzuality (CartoDB Leaflet library 0.55). De hecho, a partir de la inclusión de CartoDB.js, la integración es aún mayor, pero desafortunadamente la aparición de dicha librería tuvo lugar cuando el proyecto ya estaba muy avanzado, por lo que no ha sido utilizada en este caso.
- *Documentación:* Leaflet cuenta con una documentación muy bien estructurada, aunque no tan extensa como la de las otras opciones.

### 4.3.3. OPENLAYERS

Al igual que Leaflet, OpenLayers es una librería JavaScript Open Source. En este caso se trata quizá de la más amplia y con mayor recorrido de este tipo. En principio parecería una opción adecuada para realizar el trabajo, aunque quizá cuenta con menos ejemplos de integración con CartoDB (Ilustración 16).

---

<sup>1</sup> <https://github.com/jacobtoye/Leaflet.draw>

<sup>2</sup> <https://github.com/Vizzuality/cartodb-leaflet>

### Reading Features From CartoDB using GeoJSON

Demonstrates how to load features on OpenLayers using CartoDB SQL API.



ILUSTRACIÓN 16: EJEMPLO DE USO DE CARTODB CON OPENLAYERS<sup>1</sup>

- *Licencia:* OpenLayers es Open Source y no tendríamos ningún problema para su uso.
- *Funcionalidades:* en principio parece ser tan completa como Google Maps, y contar con todas las funciones que podamos llegar a necesitar.
- *Integración con CartoDB:* en este caso no contamos con una librería dedicada, aunque sí se ha podido encontrar un ejemplo de integración, al menos para consulta.
- *Documentación:* OpenLayers cuenta con una documentación que, aunque es muy extensa, quizá por eso mismo en ocasiones resulta algo farragosa de consultar. Sin duda lo mejor son sus ejemplos, muy numerosos y prácticos. Son la mejor herramienta para familiarizarse con el uso de OpenLayers.

#### 4.3.4. SELECCIÓN FINAL DE LA LIBRERÍA (LEAFLET)

Aunque en principio, tras la comprobación de que existen los plugins necesarios, parecía que Leaflet podría ser suficiente, se inició un proceso de creación de prototipos con cada una de las librerías, para comprobar sobre el terreno si cumplían las expectativas.

---

<sup>1</sup> <http://openlayers.org/dev/examples/cartodb-geojson.html>

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

En principio todas parecen disponer de las funciones necesarias para realizar el trabajo:

- Google Maps: es la más desarrollada, y además su integración con CartoDB es excelente, ya que antes de la versión 2.0 eran los únicos mapas base soportados y hay gran cantidad de ejemplos con ella. Su único aspecto negativo es el tipo de licencia, que no es de código abierto.
- Leaflet: es la menos desarrollada, pero también cuenta con una buena integración y cantidad de ejemplos con CartoDB. Sus llamadas son bastante sencillas y es de código abierto. Como aspectos negativos, el poder encontrar funciones aún no desarrolladas.
- OpenLayers: se trata de la librería de código abierto más avanzada, y por tanto cuenta con todas las funciones necesarias. Como aspecto negativo, se ha encontrado menos documentación sobre su integración con CartoDB, y no cuenta con librerías dedicadas para simplificarlo.

Teniendo en cuenta todo esto, finalmente se decidió para el trabajo utilizar Leaflet, con la inclusión de varios plugins.

### 4.4. CODIFICACIÓN DEL SIG WEB

Como se mencionaba al principio, para la codificación se procedió a trabajar en varios *sprints*, cada uno de los cuales tenía como objetivo alcanzar una determinada funcionalidad para la aplicación. Este sistema, escogido voluntariamente, ayudó a conseguir una comprensión más amplia de muchas de las tecnologías utilizadas, pero como contraprestación el código final generado es algo menos compacto de lo que podría haber sido, en el sentido de que se ha dado prioridad a conseguir funcionalidades antes de mantener una arquitectura equilibrada.

Las herramientas fundamentales utilizadas para la codificación han sido Aptana Studio, Firefox y Firebug. En este sentido, es de destacar la gran utilidad que han supuesto las herramientas estándar de depuración de Firefox.

Para confeccionar la aplicación se han utilizado finalmente las siguientes librerías:

- Leaflet: funcionalidad básica de Leaflet.
- Wax: funciones de interactividad para Leaflet.
- jQuery: simplificar tareas en JavaScript.
- CartoDB-leaflet: para cargar capas de CartoDB en Leaflet.
- CartoDB-popup: para crear pop-ups informativos de las presiones.

## Aplicación web para el inventario de presiones en ríos con CartoDB

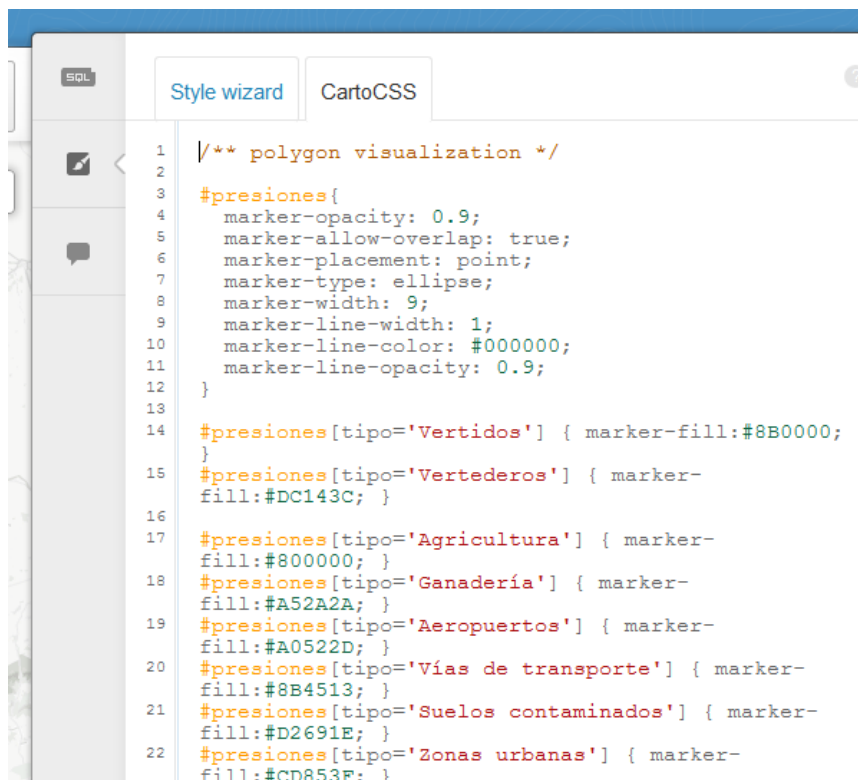
PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

- Leaflet.draw: añade controles de dibujo a Leaflet.

Con todo esto, se ha procedido a crear las funciones necesarias para la aplicación. La estructura utilizada ha sido la siguiente:

- La aplicación carga un mapa base entre varios disponibles.
- A continuación lee de CartoDB una capa por cada grupo de presiones (por motivos de diseño se ha considerado suficiente esta granularidad, los primeros prototipos creaban una capa para cada tipo, pero posteriormente se prefirió utilizar CartoCSS para dar formato a las presiones por tipos, y mostrarlas por capas a nivel de grupo (Ilustración 17)).



```
1  /** polygon visualization */
2
3  #presiones{
4    marker-opacity: 0.9;
5    marker-allow-overlap: true;
6    marker-placement: point;
7    marker-type: ellipse;
8    marker-width: 9;
9    marker-line-width: 1;
10   marker-line-color: #000000;
11   marker-line-opacity: 0.9;
12 }
13
14 #presiones[tipo='Vertidos'] { marker-fill:#8B0000;
15 }
16 #presiones[tipo='Vertederos'] { marker-
17 fill:#DC143C; }
18 #presiones[tipo='Agricultura'] { marker-
19 fill:#800000; }
20 #presiones[tipo='Ganadería'] { marker-
21 fill:#A52A2A; }
22 #presiones[tipo='Aeropuertos'] { marker-
23 fill:#A0522D; }
24 #presiones[tipo='Vías de transporte'] { marker-
25 fill:#8B4513; }
26 #presiones[tipo='Suelos contaminados'] { marker-
27 fill:#D2691E; }
28 #presiones[tipo='Zonas urbanas'] { marker-
29 fill:#CD853F; }
```

ILUSTRACIÓN 17: ESTILOS DE CARTOCSS PARA LAS PRESIONES

- Se crean controles de Leaflet para utilizar el mapa, visualizar las diversas capas, cambiar el nivel de zoom, etc.
- A partir de aquí, la aplicación trabaja de forma asíncrona y ofrece las distintas funcionalidades a través de diversos controles.

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

En la Ilustración 18 puede verse el esquema final de la aplicación.

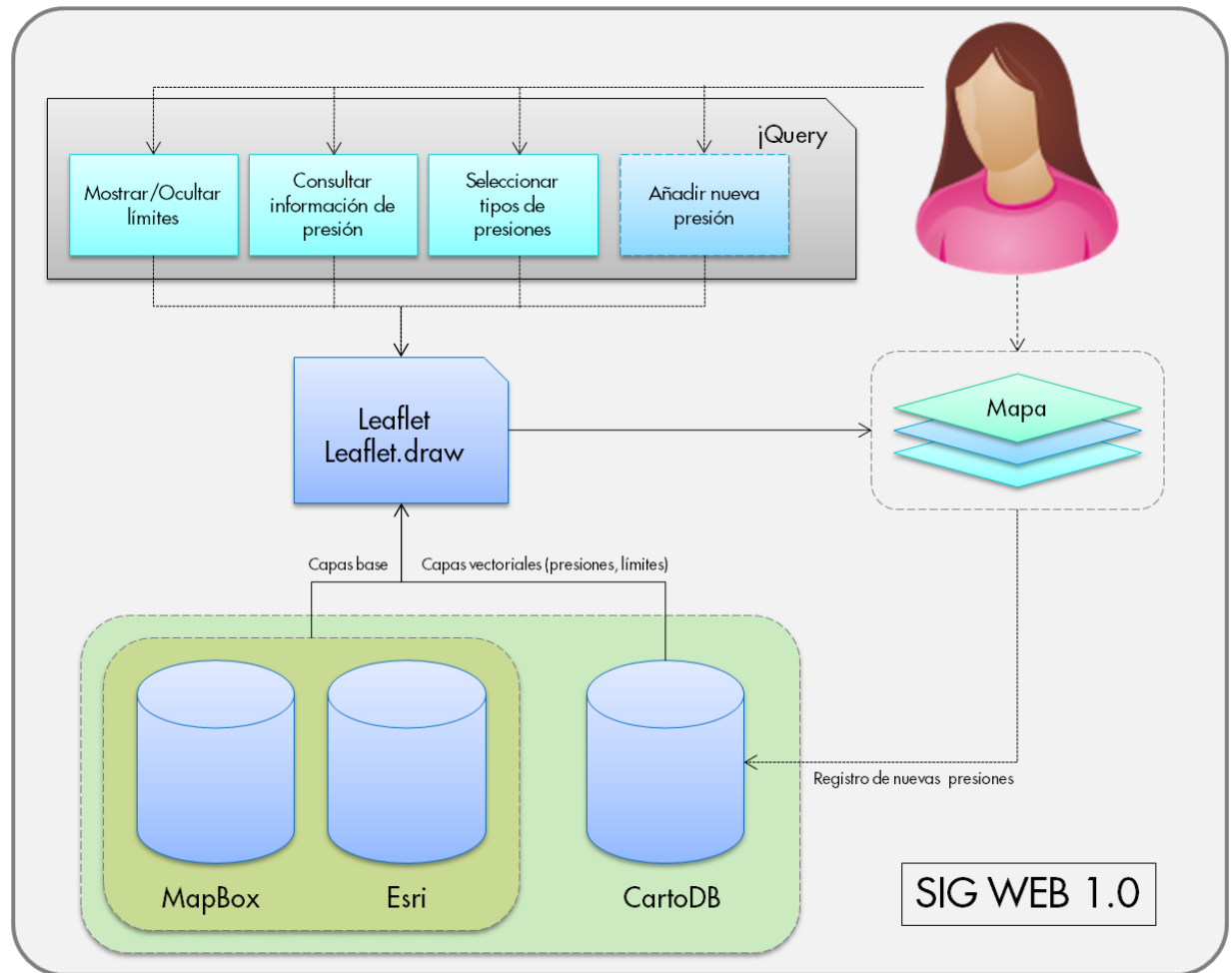


ILUSTRACIÓN 18: ESTRUCTURA DE LA APLICACIÓN SIG WEB

Es importante resaltar que, aunque se ha intentado mantener todos los elementos de estilo fuera del HTML, se ha llegado a algunos compromisos para poder solucionar algunos problemas sin tener que reconstruir gran parte del código.

Además, algunas librerías fueron adaptadas para poder ofrecer la funcionalidad deseada. Por ejemplo, pequeñas modificaciones de estilo en CartoDB-popup (no relevantes), o, de más interés, una traducción al español de todos los textos de leaflet.draw. El siguiente paso sería incluir esa traducción de forma dinámica en la librería original, aunque no será sencillo al no estar diseñado el código para aceptar múltiples lenguajes.

## 5. CONCLUSIONES

La aplicación ha sido probada con éxito con las últimas versiones de Firefox 17.0.1, Chrome 23.0, Opera 12.12 e Internet Explorer 9.0, aunque con este último aún existen algunos bugs por solventar.

Actualmente faltan las funciones de modificar presión, así como de dar de alta presiones en forma de polígono. El darse cuenta tarde que el campo *the\_geom* the CartoDB solo puede tener datos de un tipo (algo por otro lado obvio) ha provocado este retraso. En todo caso el autor siente ha aprendido mucho con la realización del proyecto y está convencido de que esta experiencia le será de mucha utilidad en una nueva ocasión.

Existen algunas conclusiones concretas que merecen la pena ser destacadas:

- CartoDB es un servicio estable. Durante la codificación y las pruebas funcionó con regularidad y, aunque podría disfrutar de algo más de rendimiento, su disponibilidad fue siempre completa.
- Utilizar librerías o tecnologías inmaduras tiene sus riesgos. Durante el tiempo, no tan extenso, que tuvo el proyecto, apareció una nueva versión de Leaflet, lo cual no supuso un gran problema. Mucho más impacto tuvo la aparición de Cartodb.js, que vuelve algunas librerías obsoletas y, en general, da un giro en la forma de interactuar con CartoDB. Nada de esto ha impedido la realización de la aplicación, pero sí hace que algunas de las soluciones implementadas se queden ahora en vía muerta, al utilizar librerías obsoletas (cartodb-leaflet).
- Proyectos que involucran muchas tecnologías distintas suponen un reto exponencial si no se conocen con anterioridad. Gran parte del tiempo empleado en el proyecto ha sido destinado a familiarizarse con librerías y tecnologías que el autor no había utilizado antes. Si bien esto fue correctamente tenido en cuenta en la planificación, a posteriori el reto a resultado aún más grande (pero no imposible) de lo esperado, en el sentido de consumo de tiempo. En este sentido el autor hace suya la idea de que la mejor forma de desarrollar un producto de software es planificar tiempo para hacer dos productos completos, el primero para entender bien el problema sin apriorismos y familiarizarse con la tecnología, y después un segundo para aplicar todo lo aprendido.
- Leaflet es sencilla, tiene una impronta pequeña, y está bien documentada, pero aún carece de algunas funciones que se pueden encontrar en Google Maps u OpenLayers. En concreto, a pesar de su sistema de plugins, carece de un sistema equivalente al de los otros para editar polígonos u otros elementos. Además no existen tantos ejemplos y material previo como para OpenLayers. A posteriori el

## **Aplicación web para el inventario de presiones en ríos con CartoDB**

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

autor no está seguro si escoger OpenLayers no habría sido una mejor opción para la aplicación en sí, incluso aunque la integración con CartoDB fuese menor.

### **5.1. FUTURAS LÍNEAS DE TRABAJO.**

La primera acción será crear una nueva versión utilizando la nueva librería CartoDB.js. En esta nueva versión se corregirían muchos de los errores iniciales que se han detectado en la actual.

Además existe un interés en trabajar al menos el segundo de los apartados opcionales:

- Proporcionar un método para la exportación de datos desde el visor a otros formatos (ej .CSV).
- Integración de fotografías y/o videos asociados a las presiones, alojados en un servicio externo como Flickr, Youtube, etc.
- Incorporación de datos ambientales complementarios tales como servicios WMS / WFS de organismos públicos (ej. espacios naturales protegidos).

## GLOSARIO

**API:** *Application Programming Interface* o Interfaz de Programación de Aplicaciones. Conjunto de funciones, y procedimientos o métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**CartoCSS:** Lenguaje basado en CSS para dar estilo a elementos cartográficos.

**CSS:** (*Cascading Style Sheets*) Hojas de estilo en cascada. Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

**Disparador:** Procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación en una base de datos.

**Geocodificación:** Proceso para la obtención de las coordenadas geográficas a partir de una dirección postal o cualquier referencia textual.

**GeoJSON:** Es un formato basado en JSON con el añadido de que permite codificar varias estructuras de datos geográficos.

**JSON:** *JavaScript Object Notation*, formato de intercambio de datos en texto plano. Constituye una alternativa al uso de XML ya que, en principio, su estructura es más comprensible de forma directa y más fácil de analizar.

**KML:** *Keyhole Markup Language*, lenguaje XML utilizado por Google Earth para representar datos geográficos en tres dimensiones. Desde 2008 es un estándar aprobado por la OGC.

**Layer:** Capa. Cada una de las capas con las que se forma una imagen superpuesta o mapa.

**Modelo ER:** Modelo Entidad-Relación. Herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

**OAuth:** *Open Authorization*, protocolo abierto que permite autorización segura de un API de modo estándar y simple para aplicaciones de escritorio, móviles, y web. Proporciona a los usuarios un acceso a sus datos al mismo tiempo que protege las credenciales de su cuenta.

**Rasterizar:** Convertir una imagen vectorial en un mapa de bits. Se llama imagen rasterizada a una imagen compuesta por un mapa de bits, en contraposición a una imagen vectorial.



## **Aplicación web para el inventario de presiones en ríos con CartoDB**

PIÓ ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

**Scrum:** Marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

**Shapefile:** Formato de datos geográficos vectoriales de ESRI, formado por varios ficheros, entre ellos: .shp (con las entidades geométricas), .dbf (atributos de los objetos) y .shx (índice de entidades geométricas).

**SIG/GIS:** Sistemas de Información Geográfica / *Geographic Information Systems*.

**Sprint:** Cada una de las iteraciones realizadas por la metodología SCRUM.

**Tile:** cada uno de los archivos unitarios con los que los servidores geográficos componen las imágenes.

**XP:** Programación extrema o *eXtreme Programming* (XP) es una metodología de desarrollo de la ingeniería de software basada en procesos ágiles.

## BIBLIOGRAFÍA

**Butler, Howard.** The GeoJSON Format Specification. [Online]  
<http://www.geojson.org/geojson-spec.html>.

**CloudMade.** Leaflet. [Online] <http://leafletjs.com/reference.html>.

**Dirección General del Agua del Ministerio de Medio Ambiente.** Manual para la identificación de las presiones y análisis del impacto en aguas superficiales. *Manual para la identificación de las presiones y análisis del impacto en aguas superficiales*. [Online]  
[http://www.magrama.gob.es/es/agua/publicaciones/impress\\_tcm7-29158.pdf](http://www.magrama.gob.es/es/agua/publicaciones/impress_tcm7-29158.pdf).

**Google.** Google Maps Javascript API Version 3. [Online]  
<https://developers.google.com/maps/documentation/javascript/>.

**Higueras, Nita Sáenz and Vidal Oltra, Rut.** *Redacción de textos científico técnicos*.

**Montagut, Roser Beneito.** *Presentación de documentos y elaboración de presentaciones*.

**National Institute of Standards and Technology.** The NIST Definition of Cloud Computing. [Online] <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.

**Network Working Group .** Request for Comments: 4627 . [Online]  
<http://www.ietf.org/rfc/rfc4627.txt>.

**OpenLayers.** OpenLayers Documentation. [Online]  
<http://trac.osgeo.org/openlayers/wiki/Documentation>.

**PostGIS.** PostGIS 2.0.2SVN Manual. *PostGIS 2.0.2SVN Manual*. [Online]  
<http://postgis.refractor.net/docs/>.

**Refsnes Data.** JavaScript Tutorial. *W3Schools Online Web Tutorials*. [Online]  
<http://www.w3schools.com/js/default.asp>.

**Vizzuality.** CartoDB Blog. *CartoDB Blog*. [Online] <http://blog.cartodb.com/>.

—. Vizzuality github Page. [Online] <https://github.com/Vizzuality>.

## **Aplicación web para el inventario de presiones en ríos con CartoDB**

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

**Wallace, Richard Donohue y Timothy.** KEEPING PACE with EMERGING web mapping technologies. [Online] <http://geography.wisc.edu/cartography/slides/emerging-web-mapping-donohue-et-al-nacis2012.pdf>.

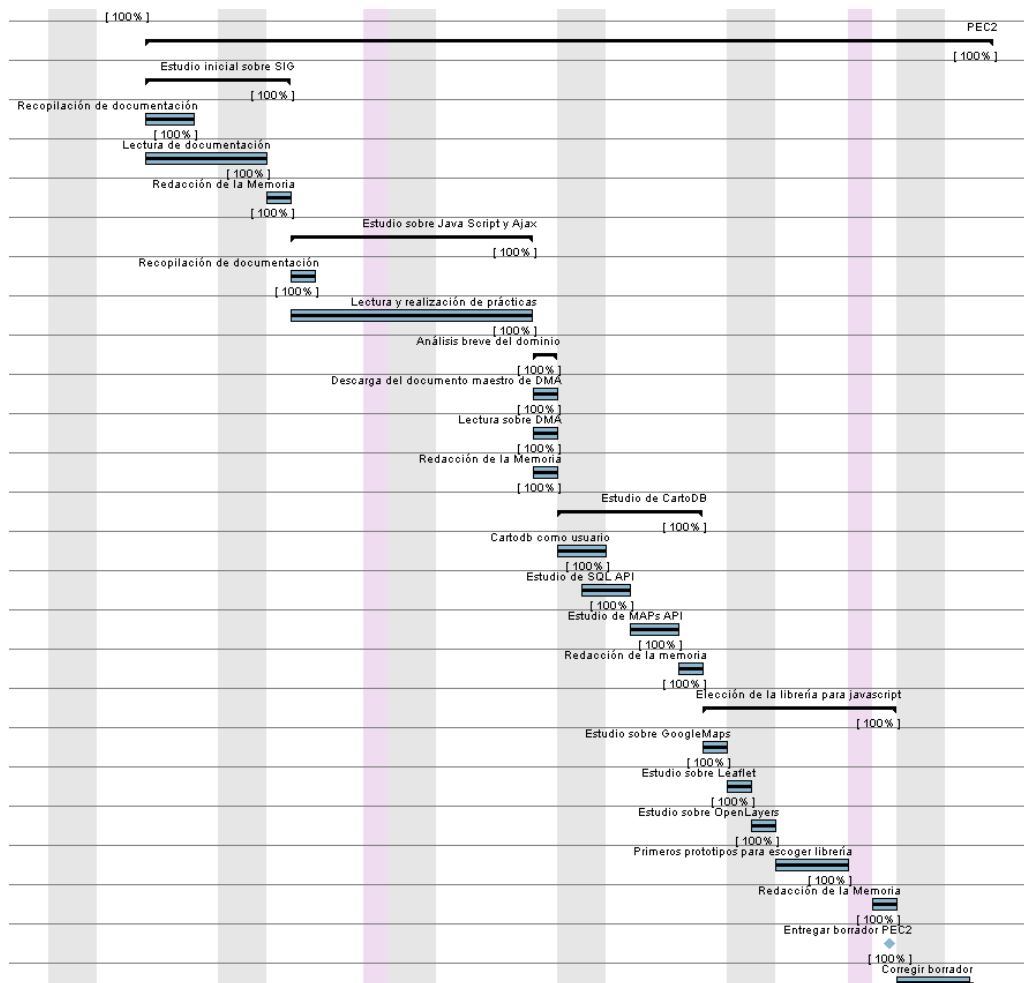
**webplatform.com.** Web Platform Docs. [Online]  
[http://docs.webplatform.org/wiki/Main\\_Page](http://docs.webplatform.org/wiki/Main_Page).

## ANEXOS

### ANEXO A: SEGUIMIENTO DEL PLAN

El seguimiento del plan se encontró con dos incidencias fundamentales. La primera, que era del tipo cubierto por el plan de contingencia, fue una mudanza que restó días de disponibilidad, pero que se suplieron atendiendo al plan de contingencia. La segunda, más importante, tuvo que ver con la estimación del tiempo necesario para empezar a codificar correctamente una aplicación web.

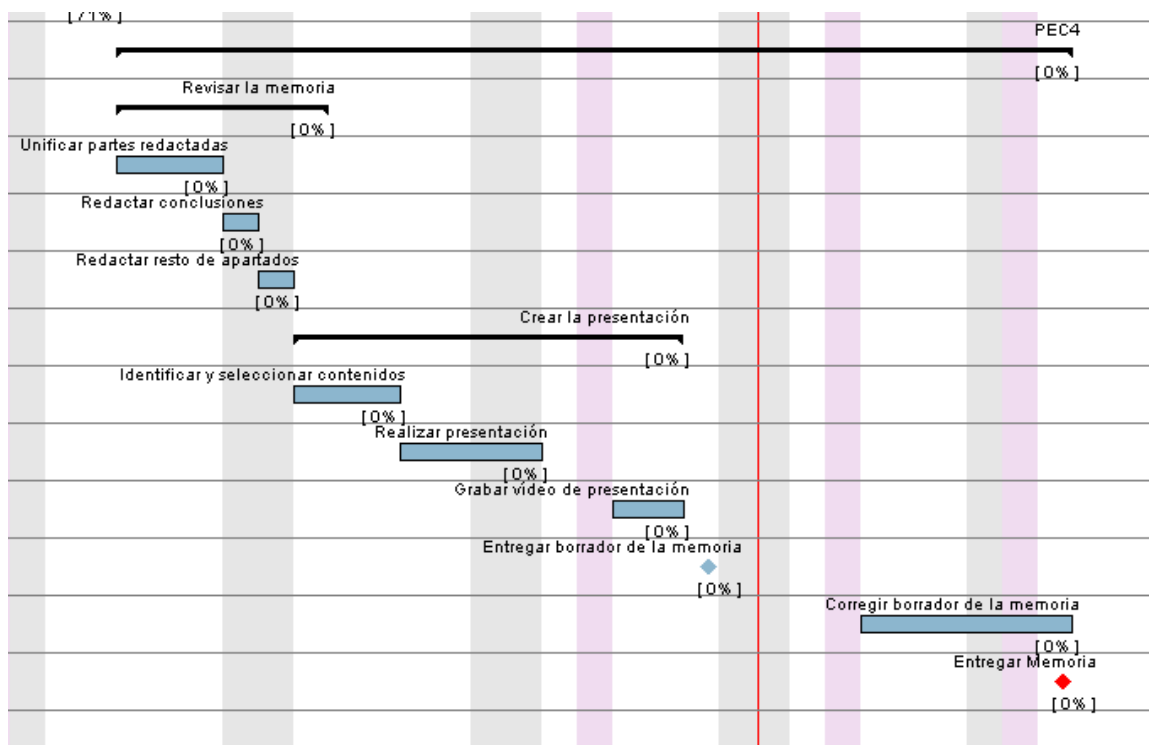
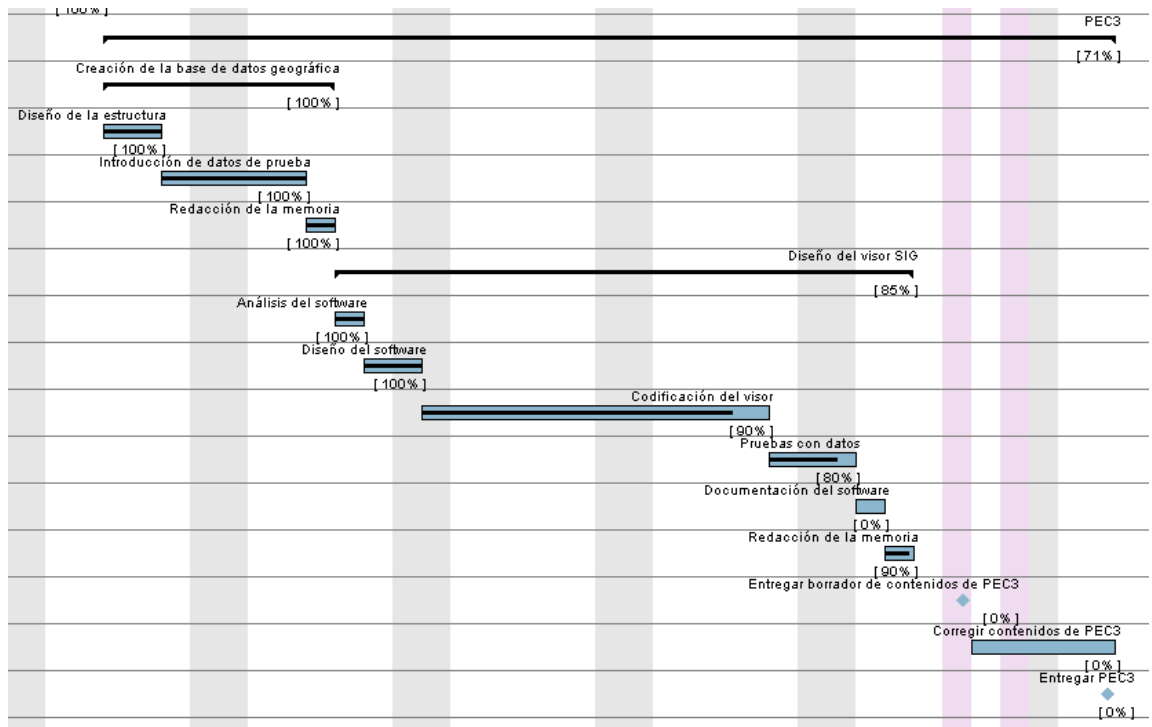
La planificación estaba estructurada alrededor de varias entregas previas del trabajo, que pueden verse a continuación.



# Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1



## **Aplicación web para el inventario de presiones en ríos con CartoDB**

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

### **ANEXO B: RECURSOS**

Durante la realización del trabajo se han utilizado diversos componentes de Hardware y Software. He aquí la lista de los empleados. En realidad, en muchos casos el uso de unos u otros ha tenido más que ver con las costumbres de autor para trabajar con código que con los potenciales beneficios de la herramienta.

#### **Hardware**

PC de escritorio con Windows Vista 64.

Ordenador portátil con OpenSuse 12

#### **Software**

GanttProject

Notepad++

Aptana Studio

Firefox

Firebug

Eclipse

Modelio

Microsoft Word

Microsoft Excel

Microsoft PowerPoint

Camtasia

Dropbox

FileZilla

### **ANEXO C: CARTODB.JS**

Durante el proceso de revisión de la presente memoria, CartoDB reorganizó sus APIs tras el lanzamiento de la versión 2.0. El principal cambio es la supresión de MAPS.API y su reemplazo por CartoDB.js, la nueva librería unificada que permite interactuar con el servicio de CartoDB.

## Aplicación web para el inventario de presiones en ríos con CartoDB

PÍO ALBERTO SIERRA RODRÍGUEZ

UOC – TFC – Curso 2012/2013 Semestre 1

El cambio más importante es la posibilidad de cargar directamente visualizaciones completas creadas en CartoDB, que incluyen el mapa base. La URL para estas visualizaciones se obtiene en la sección “Map View” pulsando el botón “Share” y seleccionando “API” (Ilustración 19).

Otro cambio sustancial, es que con el uso de CartoDB.js la integración de Leaflet es completa, es decir, una vez añadidas las librerías de CartoDB.js, las de Leaflet van incluidas también.

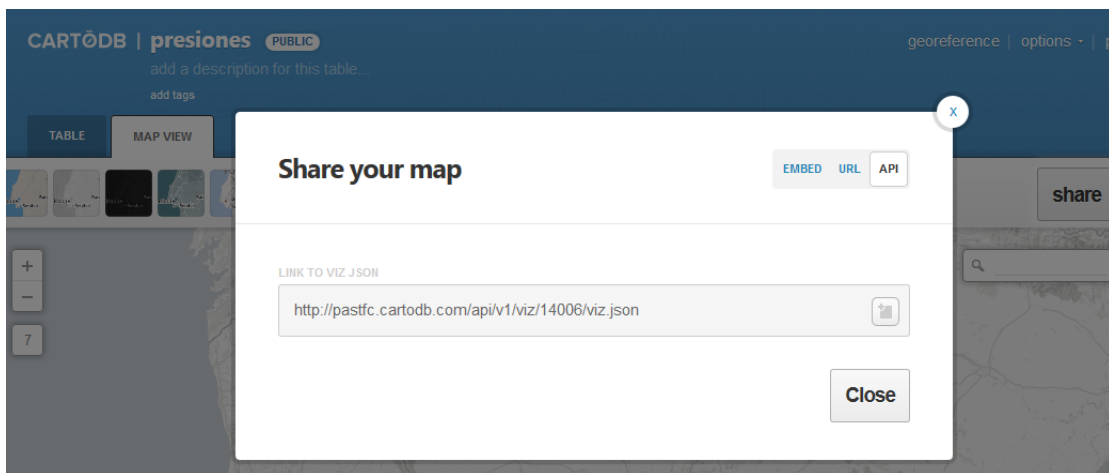


ILUSTRACIÓN 19: USO DE VISUALIZACIONES CON CARTODB.JS