

UNIVERSITAT OBERTA DE CATALUNYA

Ingeniería Técnica en Informática de Gestión (ITIG)

APLICACIÓN WEB PARA EL INVENTARIO DE PRESIONES EN RÍOS CON CARTODB

Alumno/a: Lourdes Sánchez Redondo

Dirigido por: Víctor Velarde Gutiérrez

Co-dirigido por: Antoni Pérez Navarro

CURSO 2012-13 (Septiembre /Enero)

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS.....	3
ÍNDICE DE ILUSTRACIONES	3
RESUMEN.....	5
1. INTRODUCCIÓN.....	6
2. DEFINICIÓN Y OBJETIVO DEL PROYECTO PROPUESTO.....	6
2.1. <i>DESCRIPCIÓN DEL PROBLEMA</i>	<i>7</i>
2.2. <i>DESCRIPCIÓN DEL PROYECTO.....</i>	<i>8</i>
2.3. <i>RESUMEN DE LOS OBJETIVOS.....</i>	<i>8</i>
2.4. <i>ENTREGABLES.....</i>	<i>8</i>
3. PLANIFICACIÓN DEL PROYECTO	9
3.1. <i>MARCO TEMPORAL</i>	<i>9</i>
3.2. <i>CICLO DE VIDA.....</i>	<i>9</i>
3.3. <i>HITOS.....</i>	<i>10</i>
3.4. <i>PLANIFICACIÓN Y DIAGRAMAS DE GANTT</i>	<i>11</i>
3.4.1. <i>PEC1.....</i>	<i>11</i>
3.4.2. <i>PEC2.....</i>	<i>12</i>
3.4.3. <i>PEC 3.....</i>	<i>13</i>
3.4.4. <i>PEC4.....</i>	<i>14</i>
4. LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA.....	15
4.1. <i>LA NATURALEZA DE LOS DATOS GEOGRÁFICOS</i>	<i>15</i>
4.2. <i>SOFTWARE SIG</i>	<i>16</i>
4.3. <i>TENDENCIAS ACTUALES DE LAS APLICACIONES SIG.....</i>	<i>17</i>
4.3.1. <i>OGC: Estandarización de la Información Geográfica.....</i>	<i>18</i>
4.3.2. <i>La Contribución de Google a la popularización de los SIG.....</i>	<i>18</i>
4.3.3. <i>La Neogeografía y los mushups.....</i>	<i>19</i>
4.3.4. <i>WEB 2.0: Los SIG al alcance de todos.....</i>	<i>20</i>
4.4. <i>LA COMPUTACIÓN EN LA NUBE.....</i>	<i>20</i>
5. ANÁLISIS DE LA PLATAFORMA CARTODB.....	21
5.1. <i>ARQUITECTURA DE CARTODB.....</i>	<i>21</i>
5.2. <i>TRABAJAR CON CARTODB.....</i>	<i>22</i>
5.3. <i>DIFERENCIAS ENTRE CARTODB Y LA APLICACIÓN SIG WEB DE LA CHT.....</i>	<i>24</i>
5.4. <i>ESTILOS DE MAPAS EN CARTODB.....</i>	<i>25</i>
5.5. <i>API SQL</i>	<i>25</i>
5.6. <i>MAPS API.....</i>	<i>27</i>
5.7. <i>ANÁLISIS COMPARATIVO DE PETICIONES WEB VECTORIALES Y TILES.....</i>	<i>28</i>
5.8. <i>CARTODB: APLICACIÓN EN LA NUBE. LOS CDNS.....</i>	<i>32</i>
6. DESARROLLO DE LA APLICACIÓN WEB SIG.	33
6.1. <i>ANÁLISIS DE REQUISITOS FUNCIONALES</i>	<i>33</i>
6.1.1. <i>Descripción textual de los casos de uso:.....</i>	<i>34</i>
6.2. <i>ANÁLISIS DE LIBRERÍAS.....</i>	<i>36</i>
6.3. <i>ARQUITECTURA.....</i>	<i>39</i>
6.4. <i>FUNCIONAMIENTO DE LA APLICACIÓN: PROTOTIPO DE LA INTERFAZ GRÁFICA.....</i>	<i>40</i>
6.5. <i>DISEÑO DE LA PERSISTENCIA</i>	<i>46</i>
6.6. <i>IMPLEMENTACIÓN.....</i>	<i>49</i>

7. CONCLUSIONES	53
8. NUEVAS LÍNEAS DE DESARROLLO	53
GLOSARIO	54
ANEXO I. BIBLIOGRAFÍA	57
ANEXO II.PLAN DE GESTIÓN DE RIESGOS	59

ÍNDICE DE TABLAS

Tabla 1. Hitos del Proyecto.....	10
Tabla 2. Descripción de los componentes de los datos geográficos.	15
Tabla 3. Tipos de mapas cartográficos.	16
Tabla 4. Perfiles en CartoDB.	23
Tabla 5. Criterios de valoración de librerías cartográficas.	38
Tabla 6. Contenido de los servidores: localización del código y de las librerías Javascript	40

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Clasificación presiones.	7
Ilustración 2. Diagrama de Gantt de la pec_1.	11
Ilustración 3. Diagrama de Gantt de la pec_2.	12
Ilustración 4. Diagrama de Gantt de la pec_3.	13
Ilustración 5. Diagrama de Gantt de la pec_4.	14
Ilustración 6. Instancias CartoDB.	22
Ilustración 7. Permisos en CartoDB.....	23
Ilustración 8. Arquitectura mostrando uso de CartoDB.	24
Ilustración 9. Nuevo estilo de los mapas en cartoDB versión 2.0.	25
Ilustración 10. Datos obtenidos de CartoDB.	26
Ilustración 11. Visualización del mapa de CartoDB –versión 2.0.	27
Ilustración 12. Ejemplos de url de tile para una sección de CartoDB.	29
Ilustración 13. Acceso al tile de CartoDB mostrando los límites de la CHT	29

Ilustración 14. Vista satélite de la Península Ibérica mostrando CHT con tiles.....	30
Ilustración 15. Tiles de CartoDB.	31
Ilustración 16. Superficie de la CHT en mapa de la Península por librerías de Leaflet.	31
Ilustración 17. Pantalla principal con los límites de la CHT.	40
Ilustración 18. Menú de capas en la aplicación.	41
Ilustración 19. Vista de las presiones en modo tabla.....	41
Ilustración 20. Leyenda con diferente simbología.	42
Ilustración 21. Formulario de edición de presiones.	43
Ilustración 22. Opciones de selección de presiones	43
Ilustración 23. Opciones selección de geometría.	44
Ilustración 24. Dibujo de un polígono con un punto.....	44
Ilustración 25. Dibujo de un polígono con dos puntos.....	45
Ilustración 26. Dibujo de un polígono con tres puntos.....	45
Ilustración 27. Ventana de error por omisión del campo “código”.	46
Ilustración 28. Editar una presión.	46
Ilustración 29. Diseño del modelo de datos.....	48
Ilustración 30. Tabla de las presiones en puntos de CHT en CartoDB.	49
Ilustración 31. Carpetas y ficheros del proyecto.	52

RESUMEN

El presente documento constituye la memoria del Trabajo Final de Carrera (en adelante TFC) titulado “*Aplicación web para el inventario de presiones en ríos con CartoDB*”.

Este TFC se ha realizado como última asignatura de la carrera de Ingeniería Técnica en Informática de Gestión, dentro del área de Sistemas de Información Geográfica, con el fin de poner en práctica los conocimientos adquiridos en los años anteriores, en diferentes asignaturas específicas como: Sistemas de Gestión de Base de Datos, Técnicas de Desarrollo del Software, Competencias Comunicativas para Profesionales de la Informática e Informática aplicada a la Gestión.

El documento actual se ha dividido en varios bloques, comenzando con una breve introducción para resaltar la importancia del acceso a la información y la democratización de los medios que permiten la publicación de contenidos en la Web.

En un segundo lugar, se ha definido el proyecto marcando los objetivos a alcanzar y los entregables a generar. Se ha incluido una breve descripción del proyecto y del problema que le ocupa.

Seguidamente, se ha ofrecido la planificación del trabajo, marcando los hitos del proyecto así como la duración de las tareas principales.

A continuación, se han dedicado varios párrafos a explicar la naturaleza de los Sistemas de Información Geográfica (en adelante SIG, GIS en su versión inglesa). El contenido ha pretendido ser sólo una primera toma de contacto con este tipo de Sistemas y permitir al lector tener una idea general sobre los datos geográficos. Se ha hablado de un nuevo escenario, la Neogeografía y se han comentado las nuevas tendencias de los SIG.

La base del proyecto ha sido la plataforma *CartoDB*, por lo tanto, en el quinto bloque se ha desplegado un análisis sobre esta herramienta. Se ha detallado su arquitectura y se muestran algunos ejemplos de cómo utilizar sus APIs. Se ha visto cómo los tipos de formatos o estándares que usa permiten su integración con la *tecnología Asynchronous JavaScript And XML* (en adelante *AJAX*). Se han detallado las ventajas y desventajas que supone el uso de las librerías cartográficas para *JavaScript* junto con esta Plataforma, como son *OpenLayers*, *GoogleMaps* y *Leaflet*, desde el punto de vista de los criterios elegidos para el desarrollo del proyecto que nos ocupa.

Seguidamente se ha continuado con la fase de desarrollo del software, en la que se ha desglosado un análisis, diseño, implementación del código, pruebas y mantenimiento.

Por último, se han ofrecido al lector unas conclusiones y recomendaciones desde el punto de vista del equipo desarrollador, en las que se han incluido futuras líneas de trabajo.

Los anexos al documento han recogido la bibliografía que hemos empleado para la preparación del contenido, así como glosario de los términos y un Plan de Gestión de Riesgos.

1. INTRODUCCIÓN

El acceso a la información y la democratización de los medios para publicar contenidos (blogs, redes sociales) en la red han permitido a los usuarios pasar de ser meros consumidores a productores de contenidos.

Con el avance de las nuevas tecnologías se ha aprendido a ahorrar tiempo en el desarrollo de nuevas aplicaciones, aprovechando el trabajo ya realizado por otros sin tener que empezar cada vez desde cero. Ejemplo de ello son los *mashup* que es una forma de reciclaje que utiliza los conocimientos previamente desarrollados por otros para integrarlos con un nuevo uso, permitiendo un mayor rendimiento y por tanto dando lugar a nuevas soluciones tanto en la nube como en las aplicaciones móviles.

Una aplicación Web que se vale de la computación en la nube es *CartoDB*. La tendencia de tener siempre la información lo más centralizada posible y disponible en cualquier sitio y desde cualquier dispositivo está cada vez más cerca de cualquier usuario. Sistemas de almacenamiento en la nube como *iCloud* de *Apple* permiten tener en la “nube” la información personal más importante y tener acceso a ella desde cualquier ordenador, teléfono móvil o tableta. Es cuestión de dar rienda suelta a la imaginación para adaptar el gran avance de las nuevas tecnologías a las necesidades de nuestra sociedad.

Este grupo de aplicaciones, accesibles de manera general, han propiciado el desarrollo y el cambio profundo que han experimentado los Sistemas de Información Geográfica, dando lugar a tendencias bautizadas como *Web Mapping* o Neogeografía.

2. DEFINICIÓN Y OBJETIVO DEL PROYECTO PROPUESTO

En este apartado se ha realizado una descripción de la situación de partida de la Confederación Hidrográfica del Tajo (CHT en adelante), en cuanto a sus necesidades para dar cumplimiento a lo establecido en las normativas europeas y españolas.

Posteriormente, se ha tratado de dar solución a las necesidades requeridas por parte de la CHT.

2.1. DESCRIPCIÓN DEL PROBLEMA

La normativa aprobada por la Directiva Marco del Agua (en adelante DMA)¹ emplaza a las administraciones públicas competentes en cada país a realizar un proceso de vigilancia fluvial que incluya, entre otras muchas acciones, un inventario de presiones.

Las necesidades de la CHT consisten en disponer de una aplicación Web completa, que permita gestionar el inventario de presiones de su territorio con delimitación de la parte española del Tajo.

Para la realización de la aplicación Web, la CHT ha proporcionado la información relativa al conjunto de presiones a tener en cuenta, que coinciden con lo establecido en el Anexo II de la Directiva DMA. Se dividen en 7 grupos:

- 1- Fuentes puntuales
- 2- Fuentes difusas
- 3- Extracciones de agua
- 4- Regulación del agua
- 5- Alteraciones morfológicas
- 6- Otras incidencias antropogénicas
- 7- Usos del suelo

Cada grupo comprende los tipos de presiones tal y como se reflejan en la siguiente ilustración:

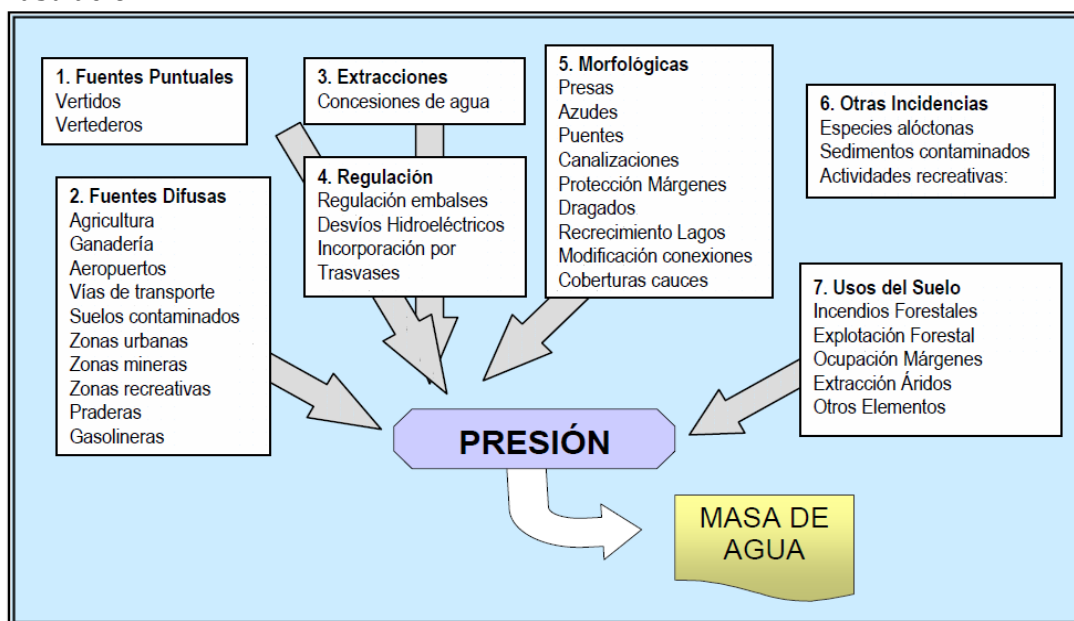


Ilustración 1. Clasificación presiones. Fuente: Manual para la identificación de las presiones y análisis del impacto en aguas superficiales.

http://www.magrama.gob.es/es/agua/publicaciones/impress_tcm7-29158.pdf

¹ La aprobación de la Directiva Marco del Agua 2000/60/CE del Parlamento Europeo y del Consejo, de 23 de octubre de 2000 establece un marco comunitario de actuación en el ámbito de la política de aguas, por el cual es obligatorio el establecimiento de programas de seguimiento del estado de las aguas con objeto de obtener una visión general coherente y completa de la situación de las mismas en la Demarcación geográfica.

2.2. DESCRIPCIÓN DEL PROYECTO

La tarea del Trabajo Final de Carrera ha consistido en la realización de una aplicación de mapas para Internet (SIG Web) que permitiese gestionar el inventario de presiones de la CHT y que se puede encontrar en la url <http://lsanchezre.site44.com/pec4/>.

La aplicación en cuestión debía tener estructura interna y funcionalidad según los parámetros especificados en el enunciado proporcionado por el consultor de la asignatura y que se enumeran en el apartado de requisitos funcionales del software de este trabajo.

1. El sistema debería ser capaz de procesar información ambiental para almacenar datos geospaciales y visualizarlos en mapas.
2. El diseño e implementación de tal sistema se realizaría según las características básicas de un SIG.

2.3. RESUMEN DE LOS OBJETIVOS

- ✓ Conocer qué es un Sistema de Información Geográfica (SIG).
- ✓ Conocer las nuevas herramientas geospaciales en Internet y evaluar sus ventajas y desventajas según su naturaleza y utilización.
- ✓ Familiarizarse con el concepto de computación en la nube, aplicado al mundo de los SIG. Reflexionar sobre los pros y contras del uso de estas herramientas.
- ✓ Diseñar una aplicación (*mashup*) que sea capaz de gestionar la información ambiental de la Confederación Hidrográfica del Tajo, empleando para tal fin los APIs de *CartoDB*, y obtener una herramienta para el manejo del inventario de las presiones de su territorio.

2.4. ENTREGABLES

- ✓ **Plan de trabajo:** planificación y estimación de las tareas necesarias para llevar a cabo los objetivos previstos.
- ✓ **Producto:** desarrollo de la aplicación Web y su correspondiente documentación técnica, que incluye el manual para el usuario (técnico de medio ambiente de nuestra aplicación) y una guía de despliegue y configuración para el administrador
- ✓ **Memoria:** documento que ha sintetizado el trabajo realizado y ha mostrado que se han alcanzado los objetivos propuestos. Ha incorporado toda la información relevante para comprender el problema planteado, la

metodología utilizada para su resolución y ha detallado la solución elaborada.

- ✓ **Presentación:** ha resumido de forma clara y concisa el trabajo desarrollado y los resultados obtenidos, mediante un vídeo.

3. PLANIFICACIÓN DEL PROYECTO

Para la planificación del proyecto se han tenido en cuenta los plazos de entrega de las diferentes Pruebas de Evaluación Continua (PECs en adelante). Para dar cumplimiento a dichos plazos se han establecido los diferentes hitos, resaltando el tiempo necesario para el cumplimiento de cada uno. Hemos indicado el ciclo de vida que hemos seguido para el desarrollo del proyecto y finalmente hemos representado el correspondiente diagrama de Gantt de cada fase.

3.1. MARCO TEMPORAL

El plazo establecido para la entrega de los elementos resultantes del Trabajo Final de Carrera (Producto, Memoria y Presentación) fue el día 07/01/2013. La planificación elaborada para este proyecto, así como, la estimación del esfuerzo, requirieron un plan de trabajo adecuado para que permitiera alcanzar los objetivos. Se adoptó como fecha inicial del proyecto el día 17/09/2012, es decir, la fecha de entrega del enunciado, y se hizo una previsión para realizar cuatro entregas parciales en las siguientes fechas, con el fin de cumplir con las entregas de la evaluación continua:

- ✓ Entrega del Plan de Trabajo (PEC 1): 02/10/2012
- ✓ Entrega Análisis de la Plataforma *CartoDB* (PEC 2): 06/11/2012
- ✓ Entrega Desarrollo de la aplicación + documentación (PEC 3): 11/12/2012
- ✓ Entrega final (PEC 4): 07/01/2013
- ✓ Debate virtual: del 23/01/2013 al 25/01/2013

3.2. CICLO DE VIDA

Para el desarrollo del proyecto, el ciclo de vida que hemos elegido es el iterativo e incremental ya que es el que mejor se adapta a nuestro trabajo por los siguientes motivos:

- ✓ Ha sido un proyecto de alta incertidumbre y ha comenzado con requisitos de alto nivel.
- ✓ Teniendo en cuenta la necesidad de entregas parciales y regulares de las PECs, el desarrollo se ha planificado en diversos bloques temporales.
- ✓ Se han obtenido resultados importantes después de cada bloque temporal.
- ✓ Se ha permitido gestionar los cambios de forma regular.
- ✓ Se ha conocido el progreso real del proyecto en cada PEC.
- ✓ Ha permitido minimizar la posibilidad de errores (por haber hecho entregas parciales) y ha aumentado la calidad.

3.3. HITOS

En la tabla siguiente se han recogido las fechas HITOS del proyecto que coinciden con las diferentes entregas de la evaluación continua, así como, las horas que se previeron dedicar para cada entrega:

HITO	FECHA	DEDICACIÓN
Inicio	20/09/2012	4 horas
Entrega Plan de Trabajo Borrador	27/09/2012	12 horas
Entrega Plan de Trabajo (PEC 1)	02/10/2012	10 horas
Entrega Análisis de la Plataforma <i>CartoDB</i> Borrador	01/11/2012	58 horas
Entrega Análisis de la Plataforma <i>CartoDB</i> (PEC 2)	06/11/2012	10 horas
Entrega Entregables y Documentación Borrador	06/12/2012	54 horas
Entrega Entregables y Documentación (PEC 3)	11/12/2012	14 horas
Entrega Memoria Borrador	20/12/2012	14 horas
Entrega Memoria y Presentación del trabajo (PEC 4) (ENTREGA FINAL)	07/01/2013	42 horas
Debate virtual	25/01/2013	
TOTAL		218 horas

Tabla 1. Hitos del Proyecto. Fuente: Elaboración propia.

3.4. PLANIFICACIÓN Y DIAGRAMAS DE GANTT

Los hitos de color naranja indican prioridad normal, mientras que los de rojo son de prioridad alta.

3.4.1. PEC1

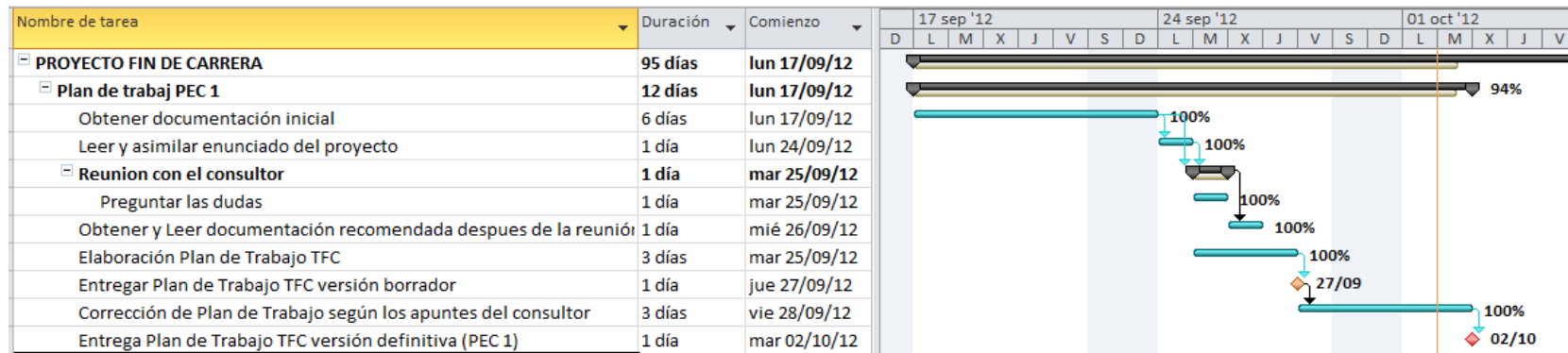


Ilustración 2. Diagrama de Gantt de la pec_1. Fuente: Elaboración propia.

3.4.2. PEC2

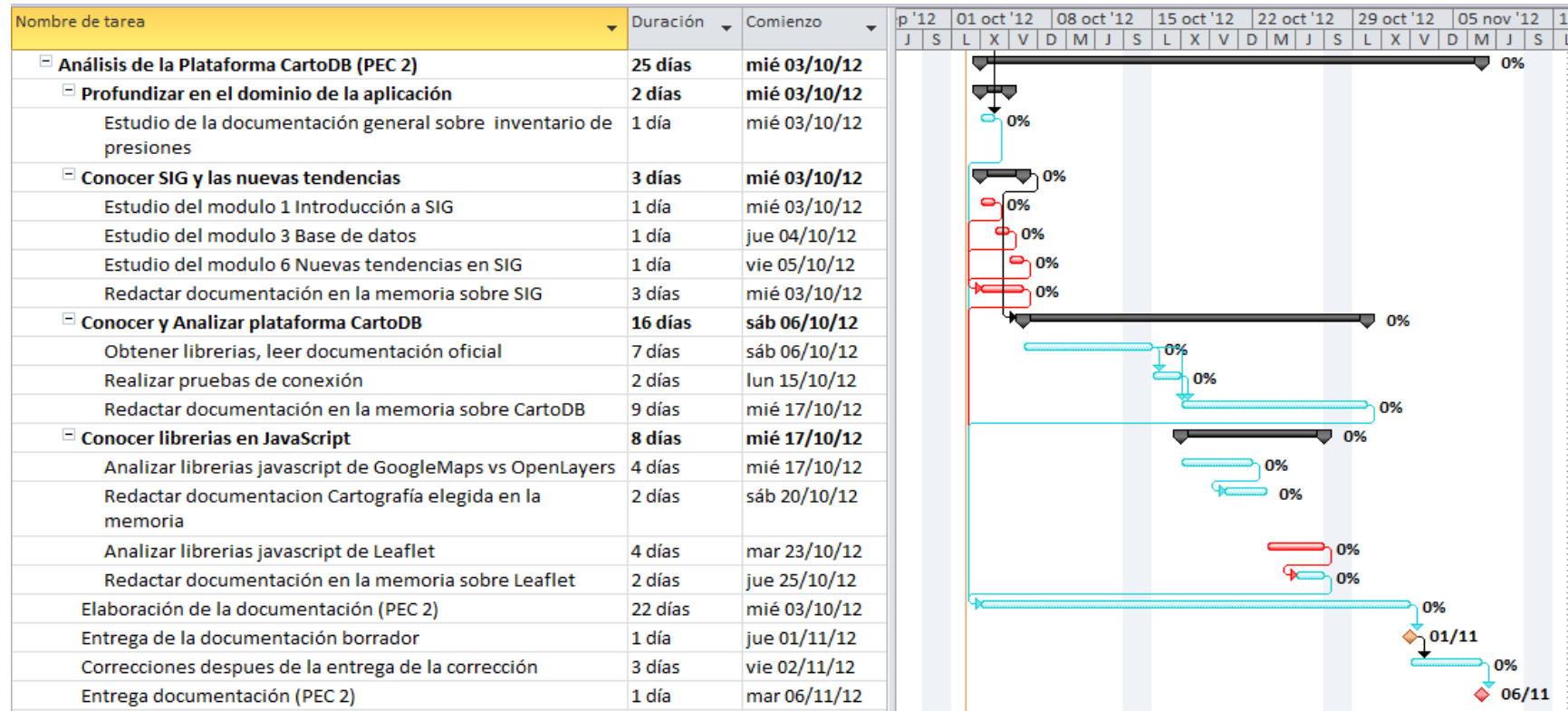


Ilustración 3. Diagrama de Gantt de la pec_2. Fuente Elaboración propia.

3.4.3. PEC 3

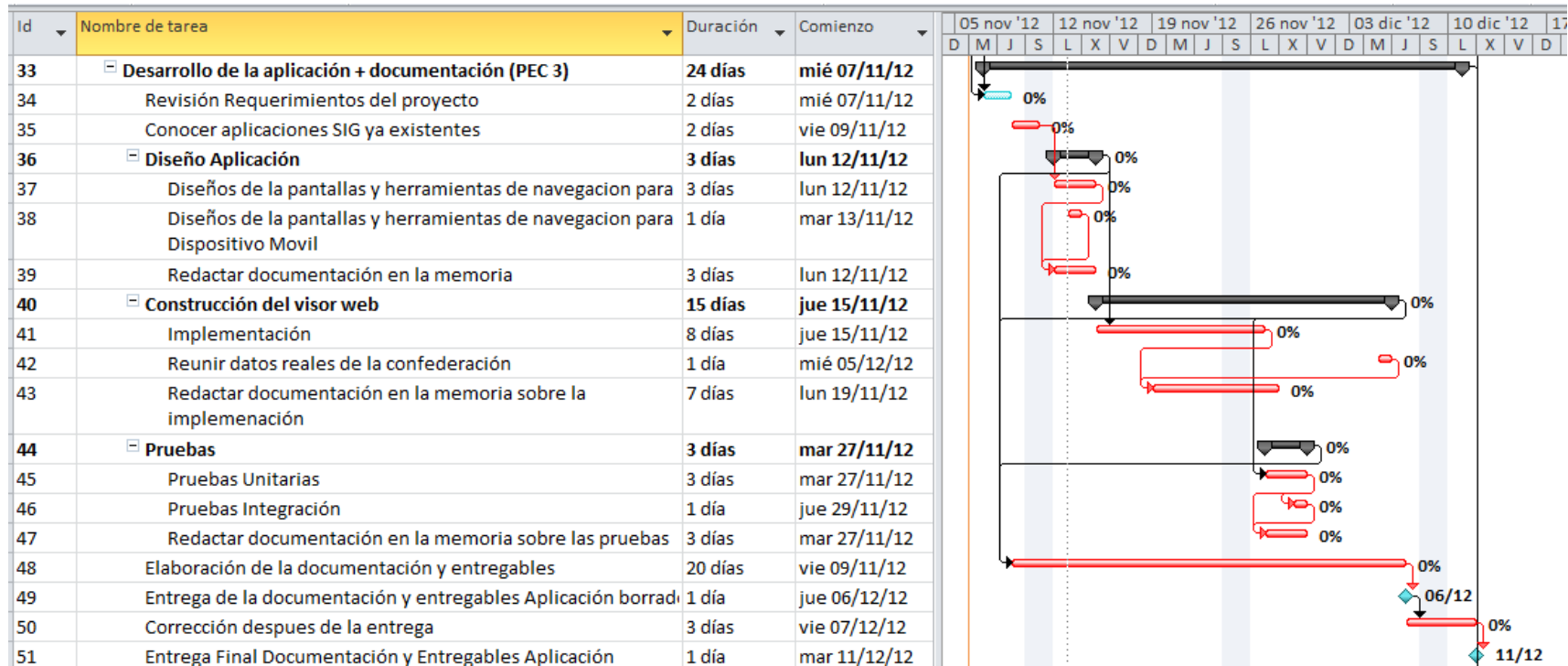


Ilustración 4. Diagrama de Gantt de la pec_3. Fuente: Elaboración propia.

3.4.4. PEC4

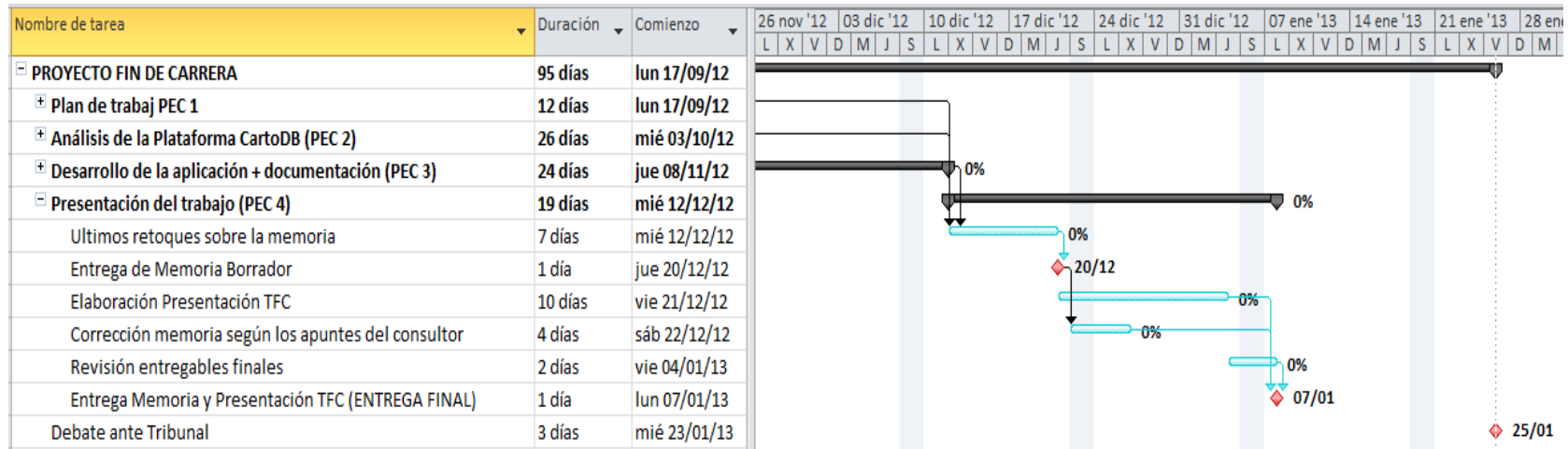


Ilustración 5. Diagrama de Gantt de la pec_4. Fuente: Elaboración propia

4. LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA

Los SIG se caracterizan por ser sistemas informáticos, compuestos por software y hardware, que disponen de las capacidades de una base de datos y trabajan con información espacial georreferenciada.

“Un sistema de información geográfica es una herramienta de cómputo de manejo de bases de datos que codifica, almacena y recupera, transformando y desplegando datos espaciales desde el mundo real, para diferentes propósitos (Burrough, 1986)”.

4.1. LA NATURALEZA DE LOS DATOS GEOGRÁFICOS

En los datos geográficos podemos distinguir tres componentes:

1. Espacial,
2. Temático,
3. Temporal.

Componente	
Espacial	<p>Supone la localización geográfica de un objeto (coordenadas). Detalla la interacción geográfica que un objeto espacial tiene con otros que le rodean (relación espacial) y representa sus propiedades (atributos propios del objeto).</p> <p>Los objetos espaciales se pueden representar por:</p> <ul style="list-style-type: none">✓ Puntos-. Localización en el espacio del objeto por las coordenadas de latitud y longitud.✓ Línea-. Sucesión de puntos.✓ Polígono-. Conjunto de líneas y puntos cerrados para formar una zona delimitada.
Temático	<p>Atributos o variable de una unidad espacial (objeto). Características de los objetos mostrados. Se expresan como variables que pueden ser continuas, discretas, fundamentales y derivadas.</p>
Temporal	<p>Representa los fenómenos geográficos de manera dinámica a lo largo del tiempo. Íntimamente relacionado con el componente espacial y temático porque las distribuciones espaciales varían a lo largo del tiempo, con lo cual, permite para un periodo de tiempo determinado analizar y observar los cambios producidos en la componente espacial y temática representada.</p>

Tabla 2. Descripción de los componentes de los datos geográficos. Fuente elaboración propia

En función de cómo se representan los objetos espaciales existen dos tipos de modelos de datos geográficos:

Raster	Divide el espacio superponiendo una malla de celdilla o píxeles. Semejante a una imagen. Cada píxel que la forma está georreferenciado, es decir, representa una posición geográfica (latitud-longitud). Los mapas tienen un píxel asociado con un valor de atributo. La información se estructura individualmente para cada uno de los atributos y se almacena en capas independientes para cada variable.
Vectorial	El objeto espacial se representa por su geometría (puntos, líneas o polígonos). Cada objeto tiene asignado un conjunto de atributos (posición, color, altura, etc.). La localización de los objetos se realiza en función de un eje de coordenadas. Dos tipos de estructuras: <ul style="list-style-type: none">✓ Cartográficas: Indican la forma y posición de los objetos por medio de sus coordenadas.✓ Topológicas: Detallan la conectividad entre los arcos y la contigüidad entre los polígonos.

Tabla 3. Tipos de mapas cartográficos. Fuente elaboración propia.

4.2. SOFTWARE SIG

Con software SIG se hace referencia al sistema informático que permite editar, integrar, almacenar, analizar, compartir y visualizar información georreferenciada.

Se destacan cuatro tipos fundamentales de software:

- ✓ **Los SIG de escritorio**, son la categoría más ampliamente utilizada. Se ejecutan en el mismo PC y ofrecen un gran número de herramientas para una gran variedad de usuarios en diferentes áreas. Destacando las siguientes herramientas para ver/editar cartografía (*Open Source*):

-Quantum Gis - www.qgis.org

- gvSIG - www.gvsig.org

- ✓ **Los Sistemas de Información Geográfica Web (SIG Web)** son productos localizados en un servidor al que acceden los usuarios por red (tienen una arquitectura cliente-servidor). Los SIG Web tienen una interfaz de usuario que ofrece funcionalidades de consulta, edición y análisis espacial. El cliente no necesita instalación alguna y la interacción se produce entre el navegador y el servidor web.

Los productos de servidores SIG permiten el acceso de un gran número de usuarios a un bajo coste. Por ello se utiliza ampliamente en entornos corporativos.

Un SIG Web es un Sistema de Información (SI), compuesto por algunos de los componentes básicos de un SI común, como por ejemplo: lenguaje de desarrollo y una base de datos (BBDD). Pero con la diferencia de que la BBDD debe ser capaz de manipular informaciones espaciales.

Un SIG Web debe tener capacidad para generar los mapas dinámicamente conforme a las peticiones de los usuarios.

CartoDB tiene la ventaja de proporcionarnos a través de su servicio de hosting una serie de componentes del SIG Web como la base de datos espacial PostGIS o el motor de renderizado de mapas que ahorra al usuario el tener que descargarlos de su repositorio de código (<https://github.com/CartoDB/cartodb20>). Lo anterior será descrito en el apartado de Análisis de la plataforma CartoDB.

- ✓ **Los componentes de desarrollo SIG**, son paquetes de herramientas de funciones SIG. Para su implementación es preciso dominar la programación. Permiten la creación de aplicaciones SIG adaptadas a las necesidades de los usuarios. Igualmente, posibilitan la programación de funciones de visualización y consulta de información geográfica.

En el entorno web destacan las librerías de *JavaScript* cartográficas: *Leaflet*, *OpenLayers* y *GoogleMaps*.

- ✓ **Los SIG móviles** están diseñados para su utilización en dispositivos móviles. Actualmente, los SIG móviles ofrecen un gran número de funcionalidades similares a los SIG de escritorio (visualización, consulta, edición y análisis simples). Las últimas novedades son los llamados Smartphone (teléfonos inteligentes), capaces de trabajar con grandes volúmenes de datos a pesar de sus reducidas dimensiones.

Los SIG móviles también destacan por disponer de aplicaciones de geolocalización, basadas en GPS u otros métodos, y con fácil acceso a Internet.

Un ejemplo de un SIG Web es la aplicación que hemos desarrollado en este proyecto.

Todo el desarrollo ha sido “en cliente” en la aplicación. El usuario accede a los datos de CartoDB mediante la implementación de un SIG Web en JavaScript que se ejecuta en el navegador del cliente. En la parte servidora no hemos necesitado implementar código en otras tecnologías puesto que se han usado servicios web ofrecidos por terceros (CartoDB).

4.3. TENDENCIAS ACTUALES DE LAS APLICACIONES SIG

La democratización que se ha producido hoy en día respecto a la utilización de las herramientas SIG ha sido debido a la aparición de:

1. **OGC**: Estandarización de la información geográfica.
2. **Google**: La contribución de Google a la popularización de los SIG
3. **Neogeografía y mashups**.
4. **WEB 2.0**: Los SIG al alcance de todos. Plataformas SIG

4.3.1. OGC: Estandarización de la Información Geográfica

Open *Geospatial Consortium* (OGC), es un consorcio de empresas y organismos públicos que tiene como misión lograr, en beneficio de todos los usuarios, un consenso para desarrollar servicios que permitan la interoperabilidad de los sistemas geoespaciales y faciliten el intercambio de la información geográfica.

Para lograr la interoperabilidad lo primero que OGC tenía que conseguir era la utilización de un estándar común para la representación de la información geográfica, así apareció la especificación GML (*Geography Markup Language*). Otras especificaciones logradas para satisfacer las necesidades de las SIG son: los servicios de Mapas (WMS), los de coberturas (WCS), los de *features* (WFS), el lenguaje para representar los datos geográficos en tres dimensiones (KML) y *CSW* (*Web Catalogue Service*) que facilita el acceso a catálogos digitales de metadatos.

En definitiva, el gran logro de OGC es conseguir un consenso para la creación de aplicaciones que permitan intercambiar la información geográfica. Antes del nacimiento de este consorcio, las empresas desarrolladoras creaban formatos comerciales incompatibles entre ellos, con lo cual, sino se utilizaba la misma aplicación y no se podían compartir los datos, hasta que se dieron cuenta que tenían que satisfacer las demanda de mercado, que no eran otras que facilitar el intercambio de información geográfica e instrucciones entre los diferentes sistemas de información. Para facilitar esta tarea, se crea en 1994, OGC.

Para CartoDB no es una prioridad cumplir con el estándar OGC en su desarrollo ya que se centran en las características de los usuarios y desarrolladores que crean aplicaciones con visualizaciones y datos de localización y prefieren cubrirlo a través de otro software como *Geoserver* que es un servidor de código abierto escrito en Java, el cual permite a los usuarios compartir y editar datos geoespaciales. Diseñado para la interoperabilidad, publica datos de las principales fuentes de datos espaciales usando estándares abiertos. Compatible con las especificaciones WMS, WCS y WFS de la OGC.

Ahora bien, la base de datos relacional de CartoDB (PostgreSQL) se apoya de la extensión geoespacial PostGIS, la cual garantiza la interoperabilidad (OGC).

4.3.2. La Contribución de Google a la popularización de los SIG

Google ha contribuido notablemente al desarrollo de los SIG porque, gracias a aplicaciones como *Google Maps* y *Google Earth*, se ha universalizado el interés por la navegación y la cartografía. *Google Maps* permite la búsqueda de cualquier lugar del planeta para poderlo visualizar en dos dimensiones, mientras que *Google Earth* lo hace en tres dimensiones. La importancia de semejantes aplicaciones está en su alto rendimiento y en la gran cantidad de datos, ya que tienen cartografiado todo el mundo.

Las aplicaciones, a las que nos referimos en el apartado anterior, sirven su cartografía base mediante teselas, proyectadas en el sistema Web, esta cartografía puede consumirse gracias a sus APIs, especialmente su librería para JavaScript (así se ha hecho en este proyecto).

Además *Google* se ha integrado activamente en la *Web 2.0*, permitiendo compartir datos, información, mapas, etc., por lo que ha sido muy beneficioso para el desarrollo de los SIG, mediante la colaboración colectiva de los usuarios interesados en dicho tema.

4.3.3 La Neogeografía y los mashups

En los últimos años se ha producido un avance muy notable de los SIG y de las tecnologías geoespaciales en general, porque de ser utilizadas por un reducido grupo de científicos y profesionales de la materia, se ha pasado a un uso generalizado y variado de las mismas. Neogeografía es un nuevo término utilizado para referirse a las aplicaciones sociales derivadas del movimiento web 2.0 que se basan en el uso de mapas, de la que hablaremos seguidamente.

Podemos definir la **Neogeografía**, como un conjunto de *“herramientas y técnicas geográficas empleadas para actividades personales o por un grupo de usuarios no expertos, para uso informal no analítico”*².

Un **mashup** es una aplicación Web que, utilizando recursos de procedencia diversa, implementa un nuevo servicio, es decir, consume servicios de otras aplicaciones a través de la *API* que estos proporcionan para conseguir una funcionalidad determinada.

Existen *mashups* de diferentes servicios, como las fotos (*Flickr*), los mapas (*Google*), los vídeos (*Youtube*), etc.

La importancia de los *mashups* en el desarrollo de los SIG radica en la facilidad de integrar y utilizar, a través de una *API* sencilla, las fuentes originales, así como, ofrecer una cartografía mundial detallada, actualizada e interactiva y otros servicios como *“geocoding”*, *“routing”*, etc. algo que hace unos años suponía un enorme y muy caro esfuerzo.

Los servicios Web son utilizados por aplicaciones de software, diferenciándose de las aplicaciones Web en que estas son utilizadas por usuarios. Por lo tanto, no poseen ninguna interfaz de usuario, siendo esta proporcionada por la aplicación que utiliza el servicio Web.

Como ejemplo de servicio web destacamos “los servicios de mapa”, ya sean implementados mediante servicios WMS, servicios de teselas de imágenes u otros. Son servidores, que al igual que cualquier página web, tienen una dirección URL y necesitan de un software específico que nos permita conectarnos a su URL y así poder mostrarnos la información devuelta en forma de mapas geográficos.

Así nuestra aplicación ha usado librerías en *JavaScript* de *Leaflet* que nos permiten aprovecharnos de servicios de mapas web basados en teselas ofrecidos por *CartoDB* y *GoogleMaps*.

²(2007) A. TURNER: <http://highearthorbit.com/neogeography-towards-a-definition/>

4.3.4. WEB 2.0: Los SIG al alcance de todos. Plataformas SIG

La Web 2.0 o Web Social ha puesto al alcance de todos los Sistemas de Información Geográfica. La Web 2.0 es una tendencia basada en la participación y colaboración entre las personas gracias a las aplicaciones interactivas. Ejemplos de ello son las plataformas como Wikipedia, Facebook, Youtube, etc.

Mediante la Web 2.0 el usuario ha pasado de ser un consumidor de contenidos a generarlos, debido a nuevos procedimientos que permiten crear y compartir la información.

Respecto a los SIG, ha supuesto una revolución en los métodos de creación y difusión de la cartografía, produciéndose una democratización en la creación de mapas, es decir, lo que antes estaba al alcance de especialistas ahora es posible realizarlo por cualquier usuario. En este aspecto, destacan los espacios colaborativos como *Street Map*, *Google My Maps*, *Wikiloc*, *Wikimapia*, etc. En estas plataformas, los usuarios pueden publicar, organizar, compartir y comentar la información geográfica creada colectivamente.

Entre las plataformas más demandadas de publicación de información geográfica en la nube destacan:

-**CartoDB**. Es la utilizada en nuestro proyecto.

-**Geocommons**. Permite a los usuarios no familiarizados con los GIS crear de forma accesible mapas complejos y realizar análisis espaciales.

-**Google Fusion Tables**. Plataforma de Google, diseñada como un gestor de base de datos en el que las operaciones se realizan directamente sobre las tablas. Se pueden combinar tablas de distinta procedencia, aplicar filtros y realizar algunos análisis básicos.

-**IkiMap**. Es una plataforma geoespacial orientada a todo tipo de usuarios. Diseñada para crear y difundir de forma sencilla todo tipo de información geográfica.

-**ArcGis Online**. Gestor colaborativo de contenido geoespacial basado en tecnología cloud de Esri.

Los elementos fundamentales que han permitido semejante desarrollo han sido, principalmente, la facilidad actual para la publicación de mapas y la generalización del uso de los GPS, mediante los cuales, los usuarios capturan los datos que luego comparten con los demás.

4.4. LA COMPUTACIÓN EN LA NUBE

Es un conjunto de computadoras conectadas entre ellas compartiendo recursos de procesamiento, memoria y almacenamiento. Es decir, la computación en lugar de hacerse en un ordenador de escritorio o en un portátil, se hace en remoto. El impulso de la computación en la nube se encuentra en el gran crecimiento del

volumen de datos de las empresas, Administraciones, etc., así como, en los altos costes que estos le generaban (almacenamiento, redes etc.).

Tipos de Servicios en la nube:

- ✓ El software como servicio (“*software-as-a-service*” o SaaS): consiste en que el proveedor pone a disposición de los clientes una aplicación para su utilización, y se encarga de su mantenimiento.
- ✓ La plataforma como servicio (“*platform-as-a-service*” o PaaS): consiste en proporcionar los medios y herramientas para desarrollar software a través de la red.
- ✓ La Infraestructura como servicio (“*infrastructure-as-a-service IaaS*): es la infraestructura del proveedor para procesamiento, almacenamiento, redes, etc., sobre los cuales, los clientes corren sus aplicaciones, sistemas operativos y tienen la posibilidad de controlar ciertas configuraciones (balanceos de carga, firewalls).

5. ANALISIS DE LA PLATAFORMA CARTODB

En el 2007, la empresa madrileña *Vizzuality*, puso en marcha la nueva herramienta para gestionar bases de datos espaciales, llamada *CartoDB*. Este revolucionario producto, además de ser *Open Source*, se creó con la finalidad de facilitar la creación de aplicaciones geolocalizadas y la realización de mapas. *CartoDB* pertenece al grupo de aplicaciones de tipo *SaaS*, aunque también como plataforma es un híbrido con aplicaciones de tipo PaaS.

El corazón de *CartoDB* esta principalmente basado en el Sistema Gestor de Base de Datos (en adelante SGBD) *PostgreSQL* que, junto con el módulo *PostGIS*, convierten esta herramienta en una base de datos geográfica en la “nube”.

Tras la migración a la versión 2.0 *CartoDB* está ahora totalmente integrado con la tecnología CDN, es decir, ofrece una distribución más rápida y escalable de los mapas.

5.1. ARQUITECTURA DE CARTODB

CartoDB se compone de una interfaz de usuario, una base de datos geoespacial basada en *PostgreSQL* y *PostGIS* 2.0, una API de SQL para ejecutar consultas SQL a través de HTTP con resultados formateados usando *GeoJSON*, KML y otros, así como, un generador de mapas que soporta SQL , estilos para la visualización

(*CartoCSS*), y autenticación por medio de *OAuth*. Su funcionamiento interno se basa en *PostGIS 2.0*, *Node.js*, y *Mapnik*.

La arquitectura general de la aplicación es un servidor central que controla la autenticación y te conduce a una instancia personal o a varias en función de la carga que tengamos.

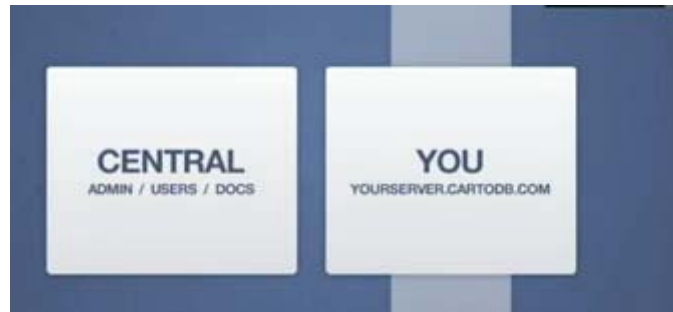


Ilustración 6. Instancias CartoDB. Fuente:
<http://www.youtube.com/watch?feature=endscreen&NR=1&v=yHhUDE68bx0>

A través de NGINX accedemos a la aplicación que nos conecta a WEB APP (implementada en *Rails*). La interfaz de la aplicación WEB APP extrae los datos de *PostgreSQL* (*PostGIS* es su extensión para trabajar con datos geográficos tanto vectoriales como raster) y de *REDIS*. *TILE SERVER* se encarga de atender las peticiones de los *MapAPIs* lanzadas por la aplicación que a su vez las pinta en la web del cliente.

5.2. TRABAJAR CON CARTODB

CartoDB ofrece una interfaz amigable que permite gestionar los datos de forma fácil y cómoda desde cualquier lugar sin importar el tipo de máquina. Lo primero que hay que hacer es crearse una cuenta de usuario (<https://cartodb.com/signup>). Hay una opción gratuita³ y otra de pago (<http://cartodb.com/pricing>).

Para trabajar con *CartoDB* sólo es necesario autenticarse como usuario (<https://cartodb.com/sessions/new>) y empezar a crear. Los usuarios que no tienen experiencia en Sistema de Información Geoespacial, también pueden usar *CartoDB*, apoyándose en la interfaz intuitiva, en los múltiples blogs (<http://blog.cartodb.com/>), en *Twitter* (<https://twitter.com/cartodb>) y en la documentación oficial (<http://cartodb.com/>) proporcionada por los creadores de la plataforma.

Respecto a la privacidad, la plataforma facilita un almacenamiento seguro de los datos. En los paquetes de pago ofrecidos por *CartoDB* existe la opción de alojar los datos en tablas privadas (solamente el propietario leerá sus datos), a diferencia de la cuenta gratuita que son siempre de lectura pública.

³ Las cuentas gratuitas presentan limitación del número de tablas a crear y el volumen de datos a alojar en estas con un máximo de 5 tablas y 5 MB respectivamente. <http://cartodb.com/pricing>



Ilustración 7. Permisos en CartoDB.

Como cualquier SGBD, *CartoDB* también organiza los datos en tablas que pueden ser fácilmente creadas y rellenas con los datos geoespaciales de diferentes modos:

1. Manualmente
2. Con programación: se crea una tabla en blanco con las columnas predeterminadas de *CartoDB* y los índices en su lugar.
3. Desde un archivo: la herramienta reconoce un gran número de formatos de archivos para cargar contenidos en tablas creadas con *CartoDB*. Admite archivos con extensión .CSV, .GZ, .TAR., .GZ, .TGZ, *ESRI Shape File* (.ZIP *), Keyhole Markup Language (.KML y .KMZ), hoja de cálculo Excel (.XLS, .XLSX), GeoJSON (.GEOJSON, .JSON), GPS Exchange Format (.GPX), Open Street Map (.OSM, .BZ2), hoja de cálculo *OpenDocument* (.ODS) y *GeoTIFF* (.ZIP)
4. Desde URL: los datos se importan de la dirección indicada.

La plataforma ofrece dos perfiles de alcance diferentes que se detallan en la Tabla 4.

<i>CartoDB</i>	
Usuario (Administrador de la aplicación)	Desarrollador
<ul style="list-style-type: none"> ✓ Crea, borra, modifica tablas ✓ Añade, borra, edita registros ✓ Genera Clave API para modificaciones remotas (INSERT, UPDATE, DELETE) 	Usa las API de <i>CartoDB</i> para: <ul style="list-style-type: none"> ✓ Añadir capas a mapas de <i>GMaps</i>, <i>Leaflet</i> u <i>OpenLayers</i> ✓ Pedir información geográfica a <i>CartoDB</i> ✓ Modificar datos en la Base de datos de <i>CartoDB</i>

Tabla 4. Perfiles en *CartoDB*. Fuente: Elaboración propia.

A continuación a modo de ejemplo se muestra una ilustración que muestra la arquitectura y uso de *CartoDB*:

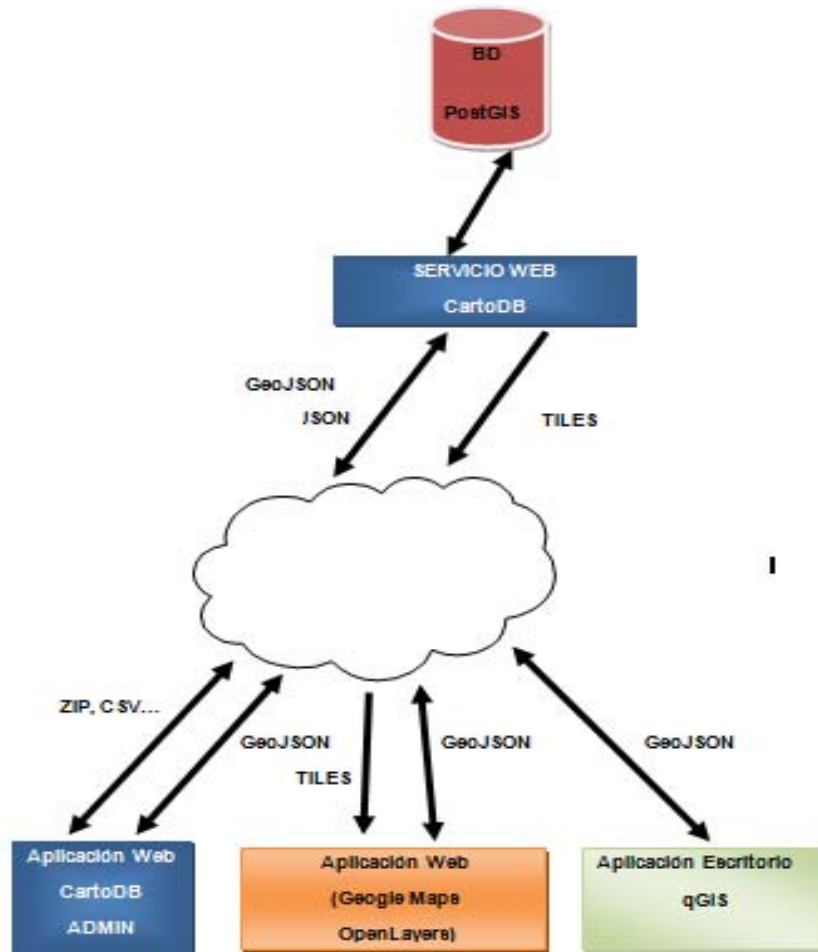


Ilustración 8. Arquitectura mostrando uso de *CartoDB*. Fuente: Elaboración propia.

5.3. DIFERENCIAS ENTRE CARTODB Y LA APLICACIÓN SIG WEB DE LA CHT

CartoDB es al mismo tiempo una aplicación web y una plataforma. La aplicación construida encima consume los servicios de dicha plataforma al igual que la aplicación hecha para la CHT.

CartoDB está diseñada para ser utilizada por personas con conocimientos de base de datos, geolocalizaciones, mientras que la aplicación para la CHT ha sido diseñada para su utilización por "técnicos medioambientales".

El administrador de base de datos de la CHT se hará cargo de la migración inicial de datos a la plataforma. Una vez que accede y se loga, puede añadir, borrar y modificar tablas y columnas.

En la aplicación para la CHT, el usuario final, es decir el "técnico de medio ambiente" sólo podrá ver, añadir, modificar o borrar registros.

5.4. ESTILOS DE MAPAS EN CARTODB

CartoDB permite utilizar el lenguaje de estilo de *MapBox* llamado *CartoCSS*, similar al *CSS*. Con él es posible aplicar diferentes estilos a los elementos del conjunto de datos según sus valores. Combinándolo con las sentencias *SQL*, se consiguen visualizaciones de datos avanzadas.

Con la versión 2.0 de *CartoDB*⁴, se han conseguido más visualizaciones, así con *CartoDB* 1.0 se introdujeron los mapas de burbujas (cambio de tamaño), y coropletas (cambio de color), ahora con la versión 2.0 se han presentado mapas de densidad renderizable tanto en hexágonos como en cuadrados.

También se ha conseguido que las ventanas de información (*Infowindows*) sean personalizables, ahora se puede elegir entre una serie de estilos para que estas encajen mejor con los mapas. Así podemos elegir entre los tonos de claro u oscuro, y entre una amplia gama de colores.

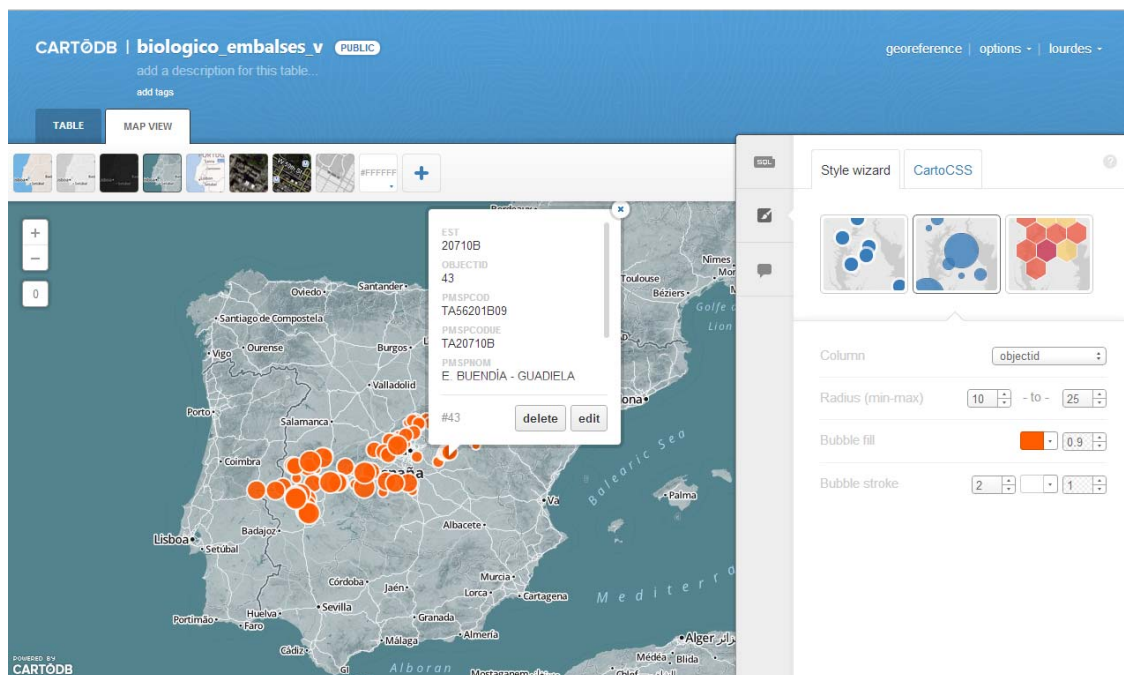


Ilustración 9. Nuevo estilo de los mapas en *cartoDB* versión 2.0. Fuente: Elaboración propia.

5.5. API SQL

Como un buen SGBD, *CartoDb* permite a sus usuarios realizar consultas sobre los datos previamente almacenados. Para tal fin, la herramienta pone a su disposición *APIs* que pueden ser usadas por una conexión *HTTP*, mediante los métodos *GET* o *POST*, simplemente introduciendo la dirección del servicio seguido por la sintaxis de la consulta.

⁴ Información expuesta en el blog oficial de *CartoDB* (<http://blog.cartodb.com/>)

El formato de la url para realizar consultas en la base de datos creado con *CartoDB* es el siguiente:

| 1. <http://{{account}}.cartodb.com/api/v2/sql?q={SQL statement}>

{*account*} es el nombre de la cuenta de usuario mediante la cual se ha dado de alta en el servicio.

A continuación, mostramos un ejemplo de url utilizada en nuestro código del proyecto para mostrar los datos de una de las tablas de *CartoDB*, en este caso, la de las presiones (representadas como vectores lineales). http://lourdes.cartodb.com/api/v2/sql?q=SELECT * FROM presiones lineas limit 2

Después de ejecutar la consulta anterior se muestran los datos recuperados de la base de datos. Los resultados se han acotado en un registro con el fin de no sobrecargar el contenido del documento. Devuelve un máximo de 2 registros de la tabla *presiones_lineales*. El número de campos es el total declarado en la tabla.

```
{
  "time":0.004,
  "total_rows":2,
  "rows":
  [
    {
      "cartodb_id":1,
      "tipo":"6",
      "description":"zona de pesca",
      "created_at":"2012-11-06T00:32:13.240Z",
      "updated_at":"2012-11-06T00:37:51.565Z",
      "the_geom":"0105000020E610000001000000010200000006000000002176A618A91
1C0BFE84585743344400059F96578D011C089BEBD9AFD3344400059F96530EB11C0268
04FB70F3544400059F965500712C096A4BC04BB354440802FD978D82412C01BAE2B074
4364440802FD978D04A12C009AC74B10F354440",
      "the_geom_webmercator":"0105000020110F00000100000001020000000600
00006185D47889FF1DC16A8208FF1CC9524120E43DF06A421EC184BE27E3B5C95241A6
EAE25DCD6F1EC14416ACA2E7CA5241DBA0904A939F1EC11ACD5EBAA6CB52417F6356D6
BCD11EC1A00D99933FCC524125E6403C3B121FC1E82A249CE7CA5241",
      "pmspcod":"6656",
      "pmspnom":"Pesca"
    },...
  ]
}
```

Ilustración 10. Datos obtenidos de *CartoDB*. Fuente: Elaboración propia.

En la Ilustración 10 mostramos lo que nos aparece en el navegador al hacer la petición al servidor de *CartoDB*. La columna llamada *the_geom* contiene la geometría en WGS 84 (EPSG: 4326) proyección. La columna *the_geom_webmercator* es en la que se almacenan todas las geometrías en una nueva versión con proyección (EPSG: 3857), conocida como “*Pseudo-Mercator*” y compatible con los sistemas de teselas más habituales como el *GoogleMaps* u *OpenStreetMap*.

A diferencia de un SGBD convencional, basado simplemente en el modelo relacional, *CartoDB* devuelve los resultados en formato *JSON*, con el fin de poder manejar esta información en combinación con *Ajax*, *JQuery*, etc...

Otra ventaja de la plataforma, es que permite que los datos obtenidos de la consulta se codifiquen directamente en formato *GeoJSON*, permitiendo representar una gran variedad de estructuras de datos geográficos. Los tipos de geometría soportados por *GeoJSON* son punto, polígono, línea, multilínea, multipolígono, colecciones de geometrías, etc....

Para que *CartoDB* presente los datos extraídos de la tabla mediante una consulta *SQL*, sólo es necesario indicar el formato *GeoJSON* en la url, como se muestra en el ejemplo siguiente:

|1. https://lourdes.cartodb.com/api/v2/sql?format=GeoJSON&q=select * from |presiones_lineas LIMIT 2000

5.6. MAPS API

CartoDB se encarga, no sólo de la gestión de los datos, sino también de la visualización, ofreciendo soluciones muy ligeras.

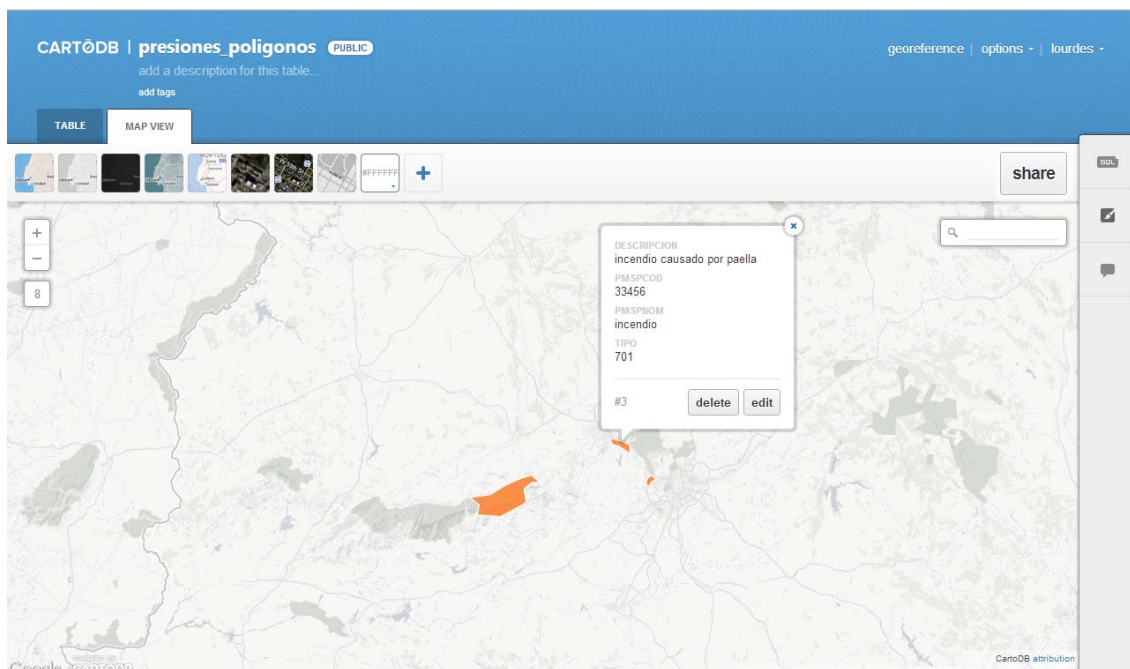


Ilustración 11. Visualización del mapa de *CartoDB* –versión 2.0. Fuente elaboración propia.

La información se puede representar mediante *tiles* o vectores.

Los vectores, en algunos casos especiales, requieren mayor cantidad de datos para guardar las coordenadas de una geometría. Para un polígono, como por ejemplo, el de los límites de la CHT, la cantidad de información a guardar y a tratar en las consultas provocaría cierta lentitud, además de generar un abundante tráfico.

Para solucionar este inconveniente, *CartoDB* facilita el acceso a una *API* que genera los *tiles* para una sección concreta del mapa. Así, en lugar de tener que descargar la información completa de los polígonos, simplemente se descarga una capa transparente que se superpone en el mapa. El almacenamiento de información es menor y por tanto, el tráfico de datos se reduce significativamente.

5.7. ANÁLISIS COMPARATIVO DE PETICIONES WEB VECTORIALES Y TILES

A continuación se ofrece una comparación entre la versión Vector y la versión *Tile*:

La petición vía web del polígono que representa los límites de la CHT es:

http://lourdes.cartodb.com/api/v2/sql?q=SELECT%20*%20FROM%20CHT_V

Y el resultado es un text de casi 12 MB (11.887.151 bytes).

```
{ "time":0.549, "total_rows":1, "rows": [{"area_km2":55780.846887, "the_geom": "0106000020E610000000100000000103000000001000000084D502004324B395E79109C0EDE003CADBA644409B7D3C9EC99109C07FDD8C83DBA644403301C5DDAA9109C0E5E11F88DBA6444091710BD18C9109C092B157D7DBA644407BEA274F709109C005137F6CDC A64440AB7BF295539109C02D059D26DDA644402FEC8930349109C075FCC6DEDDA6444057AAF4A .... "1"}]}
**** Total 11.887.151 bytes
```

En cambio la petición web mediante teselas consiste en una petición “casi continua” según vamos navegando por el mapa pero con un tráfico de datos por la red mucho menor.

- ✓ 334 bytes para los tiles vacíos (sólo color transparente) de 256x256 pixeles
- ✓ 6500 bytes para los tiles con más información (por ejemplo el tiles/cht_v/7/61/48.png)

Generalmente se suelen pedir entre 16 y 24 tiles para completar una vista del mapa. Una petición de los límites de la CHT puede suponer unos 150 KB como mucho, frente a los 11887 KB que se necesita si se pide el polígono completo en formato *GeoJSON*.

Un caso concreto es la vista del mapa conteniendo la península completa en la pantalla. En este caso el servidor de tiles de *CartoDB* nos devolverá solamente 6 tiles de unos 3 a 7 KB de tamaño y el resto serán tiles completamente transparentes de un tamaño de 334 bytes.

Para poder acceder al servicio de tiles de *CartoDB* usando las librerías de *Leaflet* crearemos mediante código *JavaScript* un objeto, una capa de manera similar a la siguiente:

```
var capa = L.tileLayer(URL_TILES, {}).addTo(map);
```

Donde el formato de la url debe ser:

| 1. http://{account}.cartodb.com/tiles/{table_name}/{z}/{x}/{y}.png

En nuestro caso, para la CHT usaremos:

```
var FRONTERA_URL = http://lourdes.cartodb.com/tiles/cht\_v/;  
var fronteraLayer = L.tileLayer(FRONTERA_URL + '{z}/{x}/{y}.png', {});  
fronteraLayer.addTo(map);
```

{z} es el zoom, mientras que {x},{y} son las coordenadas relativas del Tile dentro del mapa, la mayoría de los proveedores de mapas generan estos valores automáticamente.

A continuación mostramos 6 ejemplos de peticiones web al servicio de *CartoDB*, con su correspondiente url de tile:

| 1. http://lourdes.cartodb.com/tiles/cht_v/7/61/47.png

| 2. http://lourdes.cartodb.com/tiles/cht_v/7/62/47.png

| 3. http://lourdes.cartodb.com/tiles/cht_v/7/63/47.png

| 4. http://lourdes.cartodb.com/tiles/cht_v/7/61/48.png

| 5. http://lourdes.cartodb.com/tiles/cht_v/7/62/48.png

| 6. http://lourdes.cartodb.com/tiles/cht_v/7/63/48.png

Ilustración 12. Ejemplos de url de tile para una sección de CartoDB. Fuente: Elaboración propia.

El resultado de cada petición es un fichero .png. Acceder al tile solo tiene sentido cuando se aplica sobre un mapa.

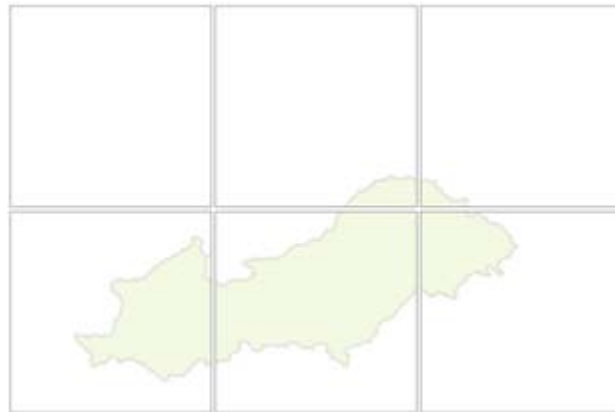


Ilustración 13. Acceso al tile de CartoDB mostrando los límites de la CHT.

El tamaño en pixeles es de 256x256 pixeles. El tamaño en bytes oscila desde los 300 bytes hasta algo más de 6 KB.

Según sean las coordenadas de la vista del mapa que queremos representar, variarán los parámetros en la peticiones que haga *LeafLet* a *CartoDB*.

Estos parámetros son los números finales de la URL.

Ejemplo: /7/61/47.

A continuación mostramos un ejemplo de cómo *LeafLeft* usa dos servicios web de mapas diferentes para mostrarnos un fondo tras los límites de la CHT respecto a la Península Ibérica.

Estos dos servicios de mapa que mostramos como ejemplo nos “dibujan” un fondo: la península ibérica.

La imagen que se muestra seguidamente es una captura del elemento HTML con id ‘*map*’ que contiene un conjunto de tiles que forman una vista de tipo satélite de la Península Ibérica.

En esta vista no hemos resaltado las divisiones de los *tiles*.

Es lo que se vería si *LeafLet* solo hubiera hecho las peticiones a *GoogleMaps*.



Ilustración 14. Vista satélite de la península mostrando CHT con tiles. Fuente elaboración propia.

En la siguiente imagen hemos representado cómo se verían los tiles devueltos por *CartoDB*. Lo que aparece en color blanco en realidad es transparente. Lo que aparece naranja en realidad sería un naranja con una opacidad de un 25% (o como lo hayamos configurado en *CartoDB*).

Es lo que se vería si *LeafLet* no hubiera recibido todavía (o no lo hubiera hecho) información de Tiles de fondo como lo son las fotos por satélite.

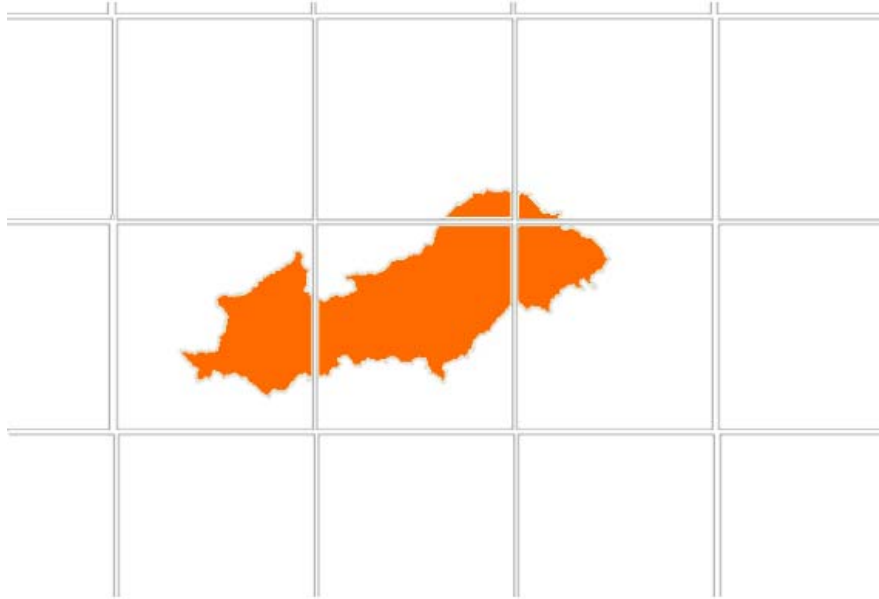


Ilustración 15. Tiles de CartoDB. Fuente elaboración propia.

El resultado mostrado por las librerías de *Leaflet* es una única imagen donde aparece la superficie ocupada por la CHT superpuesta en un mapa fotográfico de la Península Ibérica.



Ilustración 16. Superficie de la CHT en mapa de la Península por librerías de Leaflet. Fuente elaboración propia.

En este caso concreto, de representación de los límites de la CHT, por los motivos explicados anteriormente es más conveniente utilizar la técnica de Tiles frente a la de los vectores. Pero es preciso indicar que en el caso de las peticiones de presiones a la base de datos (como puntos, polígonos, rectas pequeñas) es mejor utilizar la técnica de vectores. Además tratando las presiones como vectores tenemos la posibilidad de interactuar con los puntos, líneas, áreas usando

JavaScript. Por ejemplo hemos podido hacer que un punto que representa una presión cambie de color cuando lo seleccionemos. También hemos podido pintar en una nueva capa una presión como un área de manera que hemos podido permitir su modificación.

5.8. *CARTODB: APLICACIÓN EN LA NUBE. LOS CDNS*

CartoDB es una aplicación en la nube, con lo cual ofrece las ventajas de la nube. Destacan la alta prestación de servicios tecnológicos e informáticos de muy alto rendimiento en régimen de externalización, permitiendo una disminución de costes y una mayor flexibilidad dado que, no necesitan hacer grandes inversiones en tecnología, sino que, utilizan solamente la que necesitan pagando por ella. Además permite olvidarse del mantenimiento y la actualización de la infraestructura. Beneficios que no se obtienen con los sistemas SIG más tradicionales.

Pero la nube también tiene sus inconvenientes como fallo de los servidores, alto número de visitas etc. puede dejar fuera de servicio a los usuarios. Otro riesgo es la fuga de información, dado que la computación en la nube se está consolidando en una herramienta empresarial fundamental, los programas maliciosos y los piratas cibernéticos pueden tratar de apuntar a ella para acceder a la información. También hay que señalar el gran impacto ambiental que produce la computación en la nube porque actualmente consume el 1,3% de la demanda eléctrica mundial⁵

CartoDB ha tratado de hacer frente a algunos de los inconvenientes de la nube con la integración en la versión 2.0 de un CDN (Distribución escalable de contenidos en inglés Content Delivery Network), permitiendo la escalabilidad, es decir, ahora todas las llamadas que se hacen a CartoDb pasan a través de un CDN externo, debido a una necesidad del tipo de cliente que solicitaba 3 millones de visitas al día y por eso la escalabilidad es un factor fundamental en la plataforma y es lo más importante en la solidez de la solución aportada.

El CDN de CartoDB, permite dar una respuesta rápida a los usuarios al disponer de una red de servidores distribuidos buscando la proximidad con el cliente.

Ahora bien los CDNs se enfrentan a múltiples desafíos tecnológicos:

-Escalabilidad del sistema:

Los CDNs monitorizan las condiciones de la red y son capaces de hacer frente a la información incompleta y desactualizada. Además tienen que responder a los cambios de la red. También aíslan a los clientes que son perjudiciales para la red, capturan los logs y los informes de auditoría.

-Fiabilidad del sistema

Los CDNs deben ser tolerantes ante posibles fallos.

-Visibilidad y control de contenidos

⁵ Pablo Pardo. Diario El Mundo. Edición impresa de Madrid del 10/12/2012.

Deben mantener la consistencia de la información almacenada en cache, siendo capaces de administrar el tiempo de vida de los contenidos. Igualmente tienen que velar por la integridad, a fin de que las peticiones de los clientes sean completas y poder dar respuestas correctas.

Si bien los CDNs aportan grandes ventajas en la distribución de contenidos, no menos cierto es que actualmente su costo de explotación es muy elevado.

6. DESARROLLO DE LA APLICACIÓN WEB SIG

En este apartado se realiza el análisis de requisitos funcionales para elaborar un documento de especificaciones técnicas, y de las librerías que se pueden utilizar para la realización de un proyecto de este tipo, justificando aquellas que finalmente han sido utilizadas. Posteriormente hacemos el diseño de la interfaz gráfica y el de la persistencia, destacamos el apartado de implementación del software y finalmente tratamos las pruebas.

6.1. ANÁLISIS DE REQUISITOS FUNCIONALES

Tal como se ha dicho en el apartado 2.2 de este trabajo, en descripción del proyecto, a continuación se han descrito sus requisitos funcionales:

- ✓ Pantalla/s para la gestión del inventario de presiones: alta, baja, modificación y consulta básica. Se podría definir la componente alfanumérica y espacial de las entidades, mediante formulario/s y el dibujado sobre un mapa.
- ✓ Controles básicos de navegación (cambio de mapa base, zoom, desplazamiento...).
- ✓ Representación de las entidades con una simbología clara que permitiera diferenciar su tipo.
- ✓ Identificación interactiva de entidades sobre el mapa, que mostrara una ventana emergente con sus datos (popup).
- ✓ Visualización de los límites de la demarcación hidrográfica.

6.1.1. Descripción textual de los casos de uso:

Nombre:	Insertar presión
Descripción:	Añade una presión en forma de punto, línea o polígono.
Actores:	Técnico de medioambiente de la CHT (usuario de nuestra aplicación)
Precondiciones:	El usuario tiene que conectarse a la aplicación web.
Flujo Normal:	<ol style="list-style-type: none">1. El usuario pulsa sobre el botón nueva presión2. La aplicación le muestra un formulario para cumplimentar: código en formato numérico y nombre (obligatorios), combobox para elegir un tipo (o grupo) de presión, <i>option button</i> para seleccionar una, entre tres tipos de geometrías, botón fijar tipo que deshabilita otras opciones de geometría y permite comenzar a dibujar y botón cancelar o aceptar.3. El usuario introduce el código, nombre, elige el tipo (o grupo) de presión, marca un tipo de geometría, pulsa en fijar tipo y comienza a dibujar en el mapa, cancela o acepta.4. El sistema comprueba los datos, valida la geometría y hace la inserción en la base de datos.
Flujo Alternativo:	<ol style="list-style-type: none">4. El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al usuario para corregirlo: campo código no es numérico, falta el nombre, no se ha seleccionado en el mapa ningún punto (para geometría punto), o al menos no hay dos puntos (geometría línea) o no hay seleccionados tres (polígono), también avisará si se intenta validar fuera de los límites de la CHT.
Poscondiciones:	El dato de la nueva presión se ha insertado en la base de datos.

Nombre:	Modificar presión
Descripción:	Permite cambiar la posición geoespacial o dimensiones de una presión.
Actores:	Técnico de medioambiente de la CHT (usuario de nuestra aplicación)
Precondiciones:	El usuario tiene que conectarse a la aplicación web.
Flujo Normal:	<ol style="list-style-type: none">5. El usuario hace clic sobre una figura geométrica del mapa.6. Se despliega una ventana emergente con la opción de editar.7. El usuario edita la presión.8. La aplicación le despliega un formulario con los datos cumplimentados relativos a esa presión y un botón activo de fijar tipo antes de poder dibujar en el mapa.9. El usuario pulsa sobre fijar tipo y comienza a dibujar la figura sobre el mapa.10. El sistema comprueba los datos, y hace la modificación en la base de datos.
Flujo Alternativo:	<ol style="list-style-type: none">10. El sistema comprueba los datos, y hace la modificación en la base de datos.
Poscondiciones:	El dato ha sido modificado en la base de datos.

Nombre:	Baja presión
Descripción:	Permite borrar un dato de una presión.
Actores:	Técnico de medioambiente de la CHT (usuario de nuestra aplicación)
Precondiciones:	El usuario tiene que conectarse a la aplicación web.
Flujo Normal:	<ol style="list-style-type: none">11. El usuario hace clic sobre una figura geométrica del mapa.12. Se despliega una ventana emergente con la opción de borrar.13. El usuario da al botón de borrar.14. El sistema comprueba los datos, y lo borra en la base de datos.
Flujo Alternativo:	<ol style="list-style-type: none">11. El sistema advierte de esta acción al usuario para que de su conformidad, y lo borra en la base de datos.
Poscondiciones:	El dato ha sido eliminado en la base de datos.

El caso básico de una consulta sería:

- 1) Pedir a base de datos presiones puntuales.
- 2) Pedir a base de datos presiones lineales.
- 3) Pedir a base de datos presiones superficiales.

- 4) Recibidos datos de presiones puntuales, dibujar solo las presiones cuyo grupo esté activo.
- 5) Recibidos datos de presiones lineales, dibujar solo las presiones cuyo grupo esté activo.
- 6) Recibidos datos de presiones superficiales, dibujar solo las presiones cuyo grupo esté activo.

6.2. ANÁLISIS DE LIBRERÍAS

Para realizar este proyecto ha sido necesario utilizar librerías de mapas porque facilitan el trabajo y suponen un importante ahorro en la escritura del código de *JavaScript*.

Con *JavaScript* hemos accedido a la plataforma *CartoDB* a través de peticiones web a sus dos *APIs*.

Para la visualización de la información geográfica hemos usado librerías de mapas para *JavaScript* como *GoogleMaps* y *Leaflet*.

A continuación se hace una breve referencia a las librerías cartográficas:

- *Leaflet* (versión 0.4.4): Es una librería en *JavaScript* que se ejecuta en el navegador (cliente) y es de libre distribución. El código fuente es libre, continuamente desarrollado y corregido por voluntarios. Se usa para representación de datos geográficos en mapas y para manipulación de los mismos. Utiliza *HTML5* y *CSS3*. Tiene un aspecto moderno y amigable.
- *GoogleMaps JavaScript API V3*: Es una librería de marca comercial, que sirve, al igual que la anterior, para la manipulación de datos geométricos y mapas. Además ofrece un servicio de Tiles que no se pueden modificar.
- *OpenLayers*, es una biblioteca de código abierto de las más populares en la red. Debido a su vida de desarrollo larga (antigüedad), es una de las más complejas y mejor documentada.

Es preciso indicar que también se ha utilizado una librería de uso general, *jQuery*, porque nos ha permitido manipular los elementos de la página web de forma sencilla, realizar peticiones *AJAX*, la mayoría de estas han sido a *CartoDB* pero hemos hecho una (de tiles) a *Google* utilizando un plugin de *Leaflet*.

Para decidir las librerías que se han utilizado en el proyecto se ha realizado un estudio comparativo entre ellas tal como se ha reflejado en la tabla 5. Además han sido evaluadas mediante el desarrollo de un prototipo.

Para el desarrollo se ha utilizado tecnología Web: *HTML, CSS y JavaScript*, librerías de JavaScript como *GoogleMaps JavaScript API V3* y *Leaflet*, mecanismos de intercambio de datos basados en *AJAX* y *CartoDb* (especialmente diseñado para su uso desde *JavaScript*). Leaflet es usada para mostrar en pantalla los datos obtenidos de los servidores de CartoDB. También usamos la librería de *GoogleMapsV3* para la edición de las geometrías y de los marcadores, (no fuimos capaces de hacerlo con *leaflet* ni con *cartodb.js*)

Características	<i>Leaflet</i>	<i>GoogleMaps JavaScript API V3</i>	<i>OpenLayers</i>
Facilidad de uso	Similar en todas		
	Fácilmente integrable con dispositivos móviles		Sencillez
Documentación	Escasa. Compensada con foros y blogs muy útiles	Abundante. Tiene muchos foros, etc.	Compleja. Cuesta encontrar lo que se busca. Existen libros.
Ejemplos	Limitados. Muestra dos formas de uso de la API: uso como objetos y uso con métodos como <i>jQuery</i>	Es el que tiene más ejemplos. Incluso hay <i>plugings</i> de <i>jQuery</i> para <i>Google Maps</i> . Sencillos.	Abundantes. No parecen claros
Ejemplos de uso/integración con CartoDB	Tiene	Tiene	No tiene aunque su integración con CartoDb es viable.
Licencia	Gratuita, sin limitaciones Código fuente legible	La versión 3 de Google Maps JavaScript API es un servicio gratuito disponible para cualquier sitio web que sea gratuito para el consumidor. ⁶	Gratuita, sin limitaciones Código fuente legible

⁶ <https://developers.google.com/maps/terms?hl=es>

		Código fuente ilegible.	
API respecto a html5 y css3	Nueva. Tiene en cuenta html5, css3 y librerías como <i>jQuery</i> .	2 APIs, una obsoleta y la nueva que tiene en cuenta html5 y css3	Antigua, no tiene en cuenta html5 y css3. En proceso de migración.
Plugins	Página principal contiene lista de plugins con enlaces al código fuente. Su API acepta nuevos plugins. Estamos utilizando un plugin que permite mostrar tiles de <i>GoogleMaps</i> en nuestra aplicación web.	Existen plugins de terceros no incluidos en la documentación de Google. Hay algunos plugins implementados con <i>jQuery</i>	EL API parece muy completo. No se he encontrado información sobre plugins para <i>Openlayers</i>
Soporte para desarrolladores	Tiene un blog ⁷	Realizado por Google	Está reconocida como proyecto OSGeo e implementa numerosos estándares OGC. ⁸

Tabla 5. Criterios de valoración de librerías cartográficas. Fuente elaboración propia.

Para una persona que no esté acostumbrado ni a utilizar *JavaScript* ni a tratar con geo información, todas estas APIs son igual de complejas. Aun sabiendo JavaScript, el API de cada una de estas librerías es complicado para una persona novata en GIS por la terminología usada. En base a las características definidas en la anterior tabla y después de probar las diferentes librerías en el desarrollo de un prototipo se ha considerado que las librerías que mejor se adaptaban al desarrollo del proyecto eran *GoogleMaps* y *Leaflet* porque *CartoDB* ofrece ejemplos para su uso, se dispone de *plugins* en internet. Además el desarrollador disponía previamente de ejemplos para su utilización con *GoogleMaps*. Las dos que hemos elegido están mejor documentadas y los ejemplos son más sencillos.

⁷ Por soporte entendemos que hay un intercambio de mensajería entre un técnico y un usuario para ayudarse en el uso de *Leaflet*.

⁸ Existe una Fundación <http://www.osgeo.org/> cuya misión es el desarrollo de software libre relacionado con el tratamiento de la información geoespacial. Ejemplos de ello, son la aplicación de escritorio *gvSIG* (en proceso de incubación), el *SGBD PostGIS* en el que se basa *CartoDB*. *OpenLayers* también está incluido como proyecto bajo esta Fundación.

6.3. ARQUITECTURA

La aplicación Web del proyecto será una aplicación en la “nube”, es decir, que se distribuirá de la siguiente forma:

✓ **Aplicación**⁹

Por una parte su interfaz se encontrará en el servidor web del cliente. Hay varias librerías de *JavaScript*. La mayoría las tenemos en nuestro servidor:

Hemos desplegado la aplicación de *dropbox* a través de *site44*. La url para acceso es : <http://lsanchezre.site44.com/pec4/>

Solamente la de *Google* es descargada al navegador cliente cuando se necesita.

Por otra parte, la cartografía base y la librería *JavaScript* estarán en el servidor *Google*. ya que se ha optado por la tecnología *GoogleMaps* para mostrar como capas base los mapas de *Google*.

La librería de *Leaflet* se ejecuta en el navegador del cliente y se encuentra en nuestro servidor web al igual que la librería de *jQuery*, aunque esta está también disponible en los servidores de *Google* y de otras compañías. Podríamos tener una copia en el servidor del cliente, pero es costumbre utilizar la que tengan los de *Google* en el momento de usarlas por un mejor rendimiento y por disponer de mejor memoria caché.

El origen de los tiles que usamos en nuestra aplicación de mapas de base proviene de tres fuentes distintas:

- Fondo del mapa por *CloudMade*.
- Fondo de mapa *Google*. Las fotos de vista de satélite.
- Fondo de los límites de la CHT. Se generan por *CartoDB*, y son personalizados por nosotros mismos a partir de un fichero *SHP* y de unos estilos *CartoCSS*. Para la importación de los límites de la CHT se ha descargado de la página oficial de la CHT al disco duro del desarrollador y dentro de una sesión, sólo se ha tenido que arrastrar el fichero *shp* a la ventana de *CartoDB*. El “fichero *shp*” era un *.zip* que contenía *shp, dbf, prj*, etc.

✓ **Datos:**

Se almacenarán en los servidores de *CartoDB* que a su vez, son los servidores de *Amazon*.

⁹ La parte principal de la aplicación se realizará con *HTML, CSS, JavaScript*. Las librerías *JavaScript* de apoyo serán obtenidas de servidores externos. Un usuario en una red interna sin acceso a Internet, pero con acceso a un servidor web interno con la parte principal de la aplicación, no podrá usarla.

A continuación se muestra un esquema de lo anteriormente expuesto:




Localización	Contenido
Servidor web del Cliente 	Interfaz de la aplicación, librería js de <i>Leaflet</i> (generación de mapas).
 Google	Cartografía base, API principal de acceso a mapas de Google, <i>jQuery</i> , librería <i>JavaScript (HTML)</i> .
CartoDB 	Datos de la aplicación, <i>API</i> de acceso.

Tabla6. Contenido de los servidores: localización del código y de las librerías JavaScript Fuente: elaboración propia.

6.4. FUNCIONAMIENTO DE LA APLICACIÓN: PROTOTIPO DE LA INTERFAZ GRÁFICA.

✓ Pantalla principal:

Muestra sobre el mapa de la Península Ibérica, los límites de la Confederación Hidrográfica, resaltando las diferentes presiones.

En la siguiente ilustración se muestran las capas de presiones con el fondo de la capa de cartografía de CloudMade.

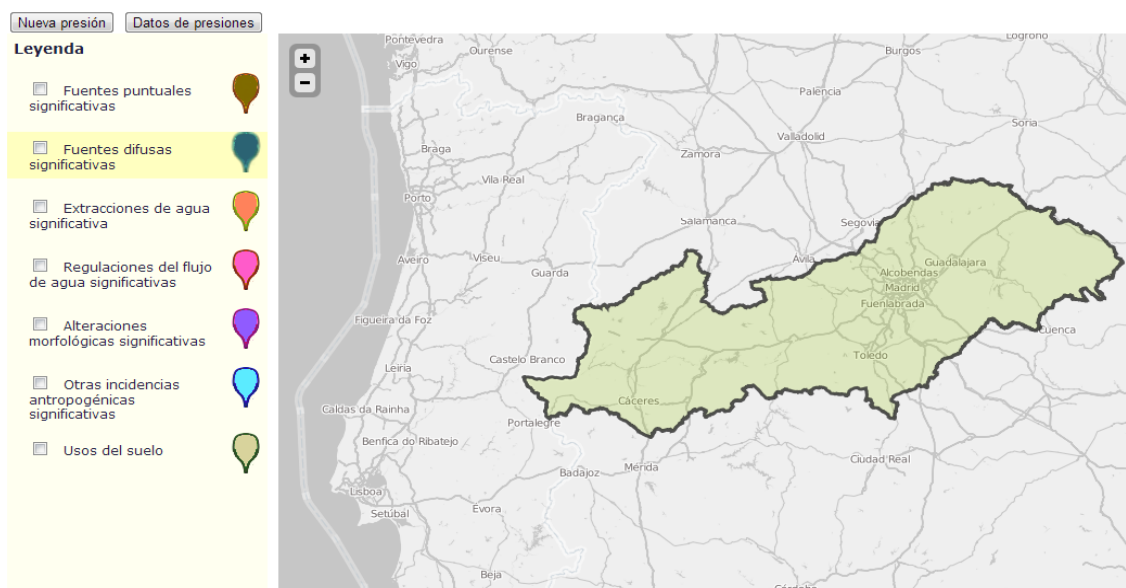


Ilustración 17. Pantalla principal con los límites de la CHT. Fuente: Elaboración propia.

✓ **Pantalla con las opciones de filtros que se pueden seleccionar:**

A continuación mostramos la Ilustración 18 con el menú de capas desplegado.

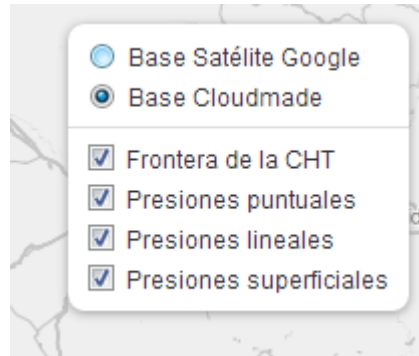


Ilustración 18. Menú de capas en la aplicación. Fuente: Elaboración propia.

En esta pantalla podemos seleccionar las diferentes capas del mapa. También podemos elegir la fuente de procedencia del mapa base, bien sea de Google o de CloudMade.

✓ **Pantalla con vista de presiones:**

Desde la página principal si pulsamos sobre el botón de datos de presiones obtenemos una vista de las presiones en modo de tabla:

Nueva presión Datos de presiones

Leyenda Listado de presiones

	COD	Nombre	Tipo
<input type="checkbox"/> Fuentes puntuales significativas	1 55	Provincia de Guadalajara	Incendios Forestales
<input type="checkbox"/> Fuentes difusas significativas	2 111	Arganda del Rey	Regulación embalses
<input type="checkbox"/> Extracciones de agua significativa	3 543653	Talavera de la Reina	Vertidos
<input type="checkbox"/> Regulaciones del flujo de agua significativas	4 25	Rivas-Vaciamadrid	Vertidos
<input type="checkbox"/> Alteraciones morfológicas significativas	5 37	Aldesanueva del Camino	Agricultura
<input type="checkbox"/> Otras incidencias antropogénicas significativas	6 88	Talavera de la Reina	GRUPO 3. Extracciones de agua significativas
<input type="checkbox"/>	7 88	Provincia de Cáceres	GRUPO 4. Regulaciones del flujo de agua significativas

Centrar en mapa

Ilustración 19. Vista de las presiones en modo tabla.

En esta vista tabular, cuando acercamos el cursor a la izquierda de la fila, vemos que se pone el fondo de color verde. En este caso al pulsar con el ratón se tiene que centrar el mapa en esa presión. En la versión 1.3 entregada no ha sido posible implementar esta funcionalidad, y es por ello que aparece una ventana avisándonos de que está pendiente centrar presión en el mapa.

✓ **Pantalla con la Leyenda:**

Hemos aplicado una simbología a las presiones según el Grupo de origen al que pertenecen. También hemos implementado la posibilidad de filtrar las presiones por Subgrupos. A diferencia de las capas, donde el usuario no ve a primera vista las que están activas, con los grupos, no pasa lo mismo ya que se pueden ver en cualquier momento en la leyenda. Se muestra una lista con todos los grupos y un checkbox asociado a cada uno de ellos, de forma que sólo se pinten en el mapa los grupos que estén previamente seleccionados.



Ilustración 20. Leyenda con diferentes simbologías.

✓ **Pantalla con formulario de edición de presiones y de añadir nueva presión:**

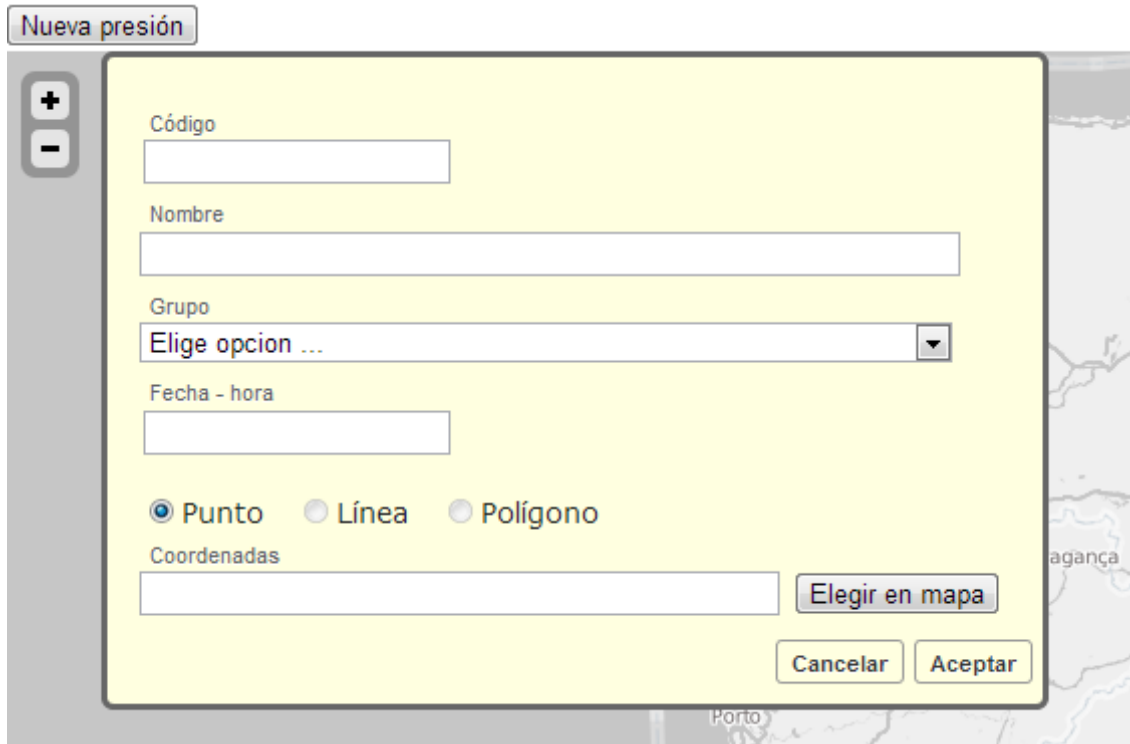


Ilustración 21. Formulario de edición de presiones. Fuente: Elaboración propia.

A través de esta pantalla se insertan las presiones. En primer lugar, pulsaremos el botón de Nueva presión que nos conducirá al formulario a través del cual procederemos a insertar el código, darle un nombre, elegir tipo de presión entre las opciones mostradas en la pantalla siguiente:

- Trasvases
- GRUPO 5. Alteraciones morfológicas significativas
- Presas
- Azudes
- Puentes
- Canalizaciones
- Protección Márgenes
- Dragados
- /Recrecimiento Lagos
- Modificación conexiones
- Coberturas cauces
- GRUPO 6. Otras incidencias antropogénicas significativas
- Sedimentos contaminados
- Actividades recreativas
- GRUPO 7. Usos del suelo
- Incendios Forestales
- Explotación Forestal
- Ocupación Márgenes
- Extracción Áridos
- Otros Elementos
- Elige opcion ...

Ilustración 22. Opciones de selección de presiones. Fuente: Elaboración propia.

El usuario tiene la opción de seleccionar el origen (Grupo1, grupo2, ...) o el tipo, que sería una presión concreta dentro del grupo. Cada grupo dispone de varios tipos de presiones. Por ejemplo el Grupo 1, Fuentes Puntuales significativas, contiene los tipos Vertidos y Vertederos. Evidentemente, si se selecciona el Grupo se muestran el conjunto total de las presiones de dicho grupo sin hacer distinción por tipo de las mismas (te muestra todas). En el formulario podremos indicar la fecha y hora en que se ha tomado la presión. Si bien para la versión 1.3 no es funcional. También existe la opción de indicar la forma geométrica (punto, línea y polígono) para mostrar la presión en el mapa, una vez seleccionada una de ellas y pulsando sobre el botón de fijar tipo, quedarán deshabilitadas las otras opciones de geometrías posibles. A partir de este momento ya podemos dibujar en el mapa los puntos.

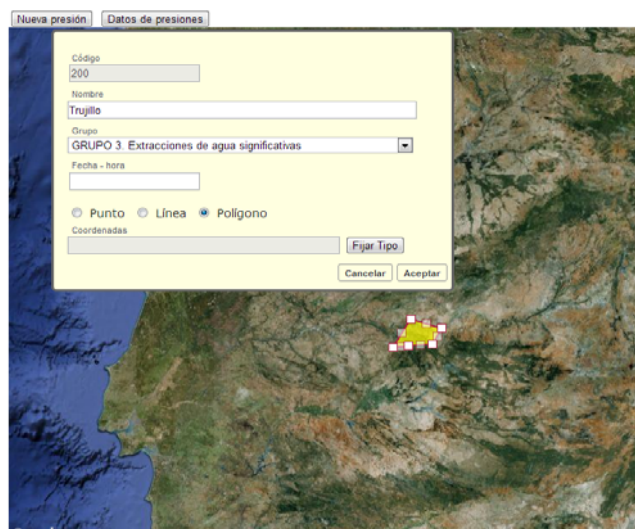


Ilustración 23. Opciones de selección de geometrías

-Si hemos elegido polígono y hemos fijado el tipo:

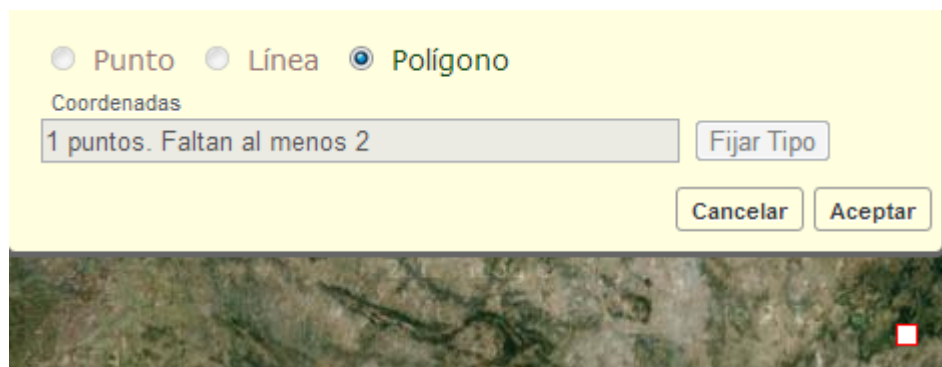


Ilustración 24. Dibujado de un polígono, con un punto



Ilustración 25. Dibujado de un polígono, con dos puntos

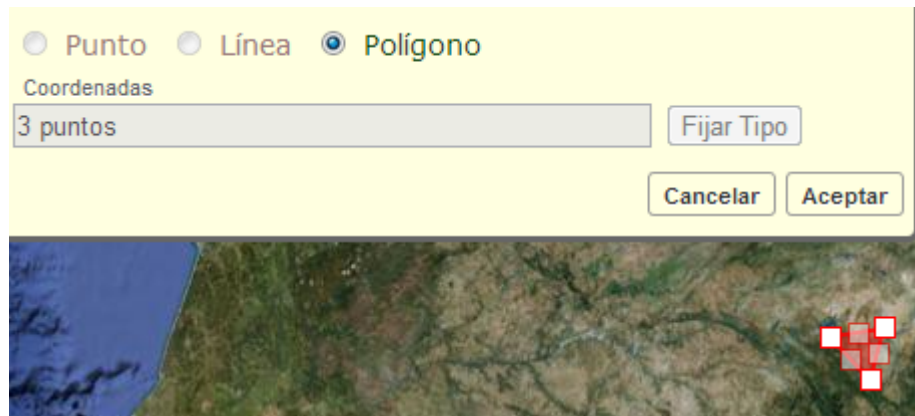
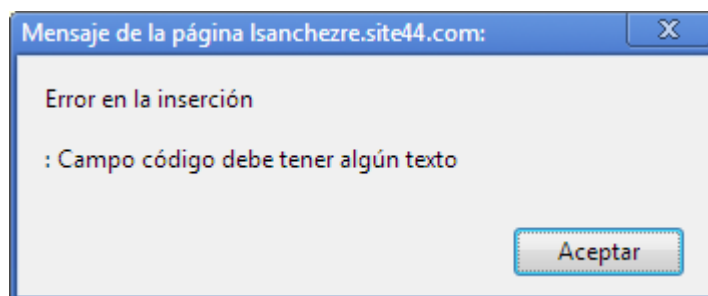


Ilustración 26. Dibujado de un polígono, con tres puntos

En las ilustraciones 7, 8 y 9 vemos como la aplicación va avisando al usuario de que para pintar un polígono se necesitan mínimo tres puntos. Esto es extensible a una línea que le pide como mínimo dos puntos, si no introduce el número mínimo de puntos exigido no validará la figura y no será insertada o modificada.

Una vez cumplimentado el formulario procedemos a aceptar, quedando insertada en el mapa la figura geométrica. Para que la aplicación nos permita completar la opción de aceptar es imprescindible que se hayan rellenado los campos de código (formato numérico) y nombre (no puede estar vacío), de no hacerlo saldrán las siguientes ventanas de advertencia:



Si procedemos a insertar texto:

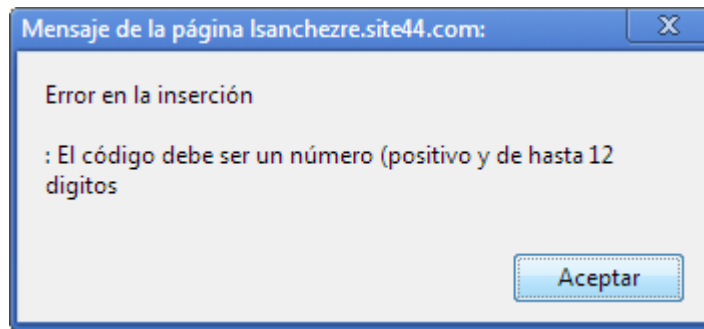


Ilustración 27. Ventana de error por omisión del campo "código".

Existe la posibilidad de cancelar la edición de la presión mediante el botón correspondiente situado junto al botón "Aceptar" del formulario de "edición de presiones".

✓ **Pantalla de edición y borrado de las presiones:**

Pantalla principal con un "popup" de una presión. La ventana emergente tiene dos botones para las acciones de "editar" y "eliminar" la presión.

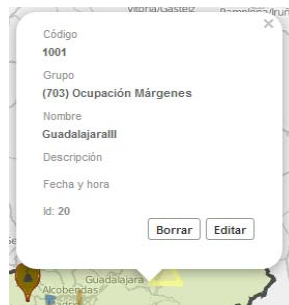


Ilustración 28. Edición de una presión. Fuente: Elaboración propia.

Una vez que se hace clic sobre una presión del mapa se muestra un bocadillo que nos ofrece la identidad y características de la presión. Nos da la posibilidad de editarlas o borrarlas.

Si elegimos la opción editar se mostrará el formulario de inserción de presiones anteriormente descrito, a través del cual realizaremos las diferentes modificaciones.

6.5. DISEÑO DE LA PERSISTENCIA

La persistencia se ha realizado mediante el uso de una base de datos geoespacial, concretamente se ha hecho uso del servicio web ofrecido por una empresa externa: CartoDB.

Esta empresa nos permite guardar los datos en tablas geoespaciales ubicadas en sus servidores, nos ofrece una aplicación web para la administración de la base de datos y unos servicios web para acceso a las tablas vía web. Aunque se ha usado JavaScript también se podía haber consumido el API desde otros lenguajes.

Además de un servicio especial de “tiles” al estilo GoogleMaps que se ha usado para la representación de los límites de la CHT.

Las presiones han sido almacenadas en tablas geoespaciales que tienen las siguientes características:

- ✓ Tener un campo de tipo especial “Geometría”. En realidad el tipo del campo será: o punto, o línea o polígono. Cada tabla solo puede contener uno de estos tipos de geometría. Lo que significa que debemos tener al menos 3 tablas para guardar los datos de las presiones.
- ✓ Poder crear un índice geoespacial usando el campo geometría.
- ✓ Aceptar consultas SQL especiales donde la condición es una “condición geoespacial”: `SELECT * FROM PRESIONES_POLIGONOS WHERE ST_Intersects (the_geom, ST_Buffer (ST_SetSRID('POINT(40,585-4,132) '::geometry,4326),1)) AND TIPO = 701;` Devolviendo todos los incendios forestales que tengan alguna zona que intersecte con un círculo de 1 grado (unos 110 km) de radio y con centro en la estación de cercanías del Escorial. El ejemplo está tomado de <http://developers.cartodb.com/examples/query-by-distance.html>, en nuestro proyecto no se ha hecho ningún tipo de consulta de este tipo.

Han sido utilizadas tres tablas para las presiones.

Las tres tablas tendrían los mismos campos excepto el campo Geometría y algún campo más:

- ✓ Tabla para puntos
- ✓ Tabla para líneas (o polilíneas). Tendrá un campo extra: longitud
- ✓ Tabla para polígonos (o multi polígonos). Tendrá dos campos extra: área y perímetro

A continuación mostramos Diseño del modelo de datos adaptado a CartoDB:

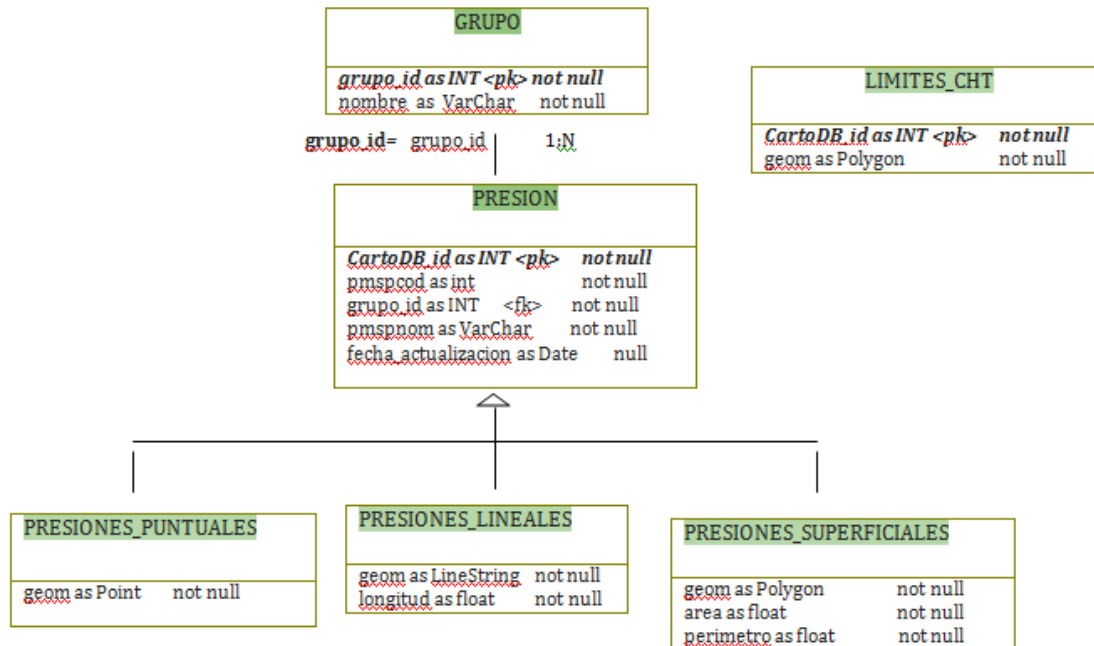


Ilustración 29. Diseño del modelo de datos. Fuente: Elaboración propia.

Presión tiene una relación de herencia con las tablas PRESIONES_PUNTUALES, PRESIONES_LINEALES y PRESIONES_SUPERFICIALES, en realidad no existe la tabla PRESIONES ya que de esta heredan tres clases.

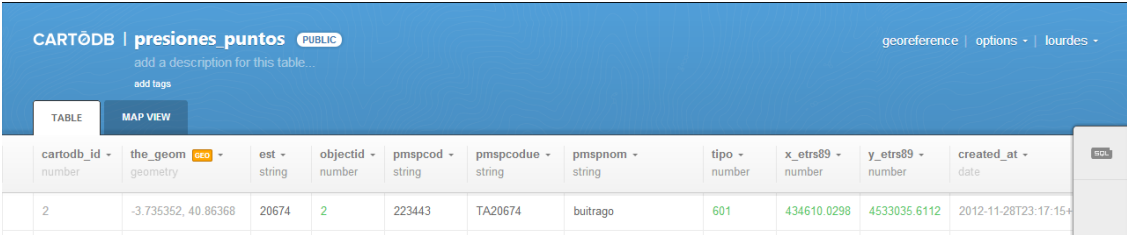
Grupo tiene una relación de 1 a muchos con la tabla PRESION mediante el campo "grupo_id".

LIMITES_CHT solo contiene un polígono que representa los límites de la CHT. Este polígono es muy detallado. Contiene muchos vértices (aprox 185732).

En detalle en la tabla LIMITES_CHT sólo tiene un registro. El campo THE_GEOM de este registro es de tipo "MultiPolygon". Este multipolígono consta de un solo polígono. Este polígono consta de un exteriorRing (no tiene huecos). Este exteriorRing está formado por 185732 vértices.

En CartoDB tenemos las tablas presiones_puntuales, presiones_lineales y presiones_superficiales como tres tablas independientes.

✓ **Captura de la aplicación de *CartoDB* mostrando una tabla de puntos:**



cartodb_id - number	the_geom - geometry	est - string	objectid - number	pmspcod - string	pmspcodue - string	pmspnom - string	tipo - number	x_ets89 - number	y_ets89 - number	created_at - date
2	-3.735352, 40.86368	20674	2	223443	TA20674	buitrago	601	434610.0298	4533035.6112	2012-11-28T23:17:15+

Ilustración 30. Tabla de las presiones en puntos de CHT en CartoDB. Fuente: Elaboración propia

6.6. IMPLEMENTACIÓN

Esta aplicación web se ha realizado implementado el código en un único lenguaje: *JavaScript*.

Este lenguaje es el que ha sido usado en la parte cliente (el navegador).

En la parte servidora no se ha necesitado implementar código en otras tecnologías como ASP, ASP.NET, JSP, etc., puesto que se han usado servicios web ofrecidos por terceros (CartoDB).

El código fuente ha estado compuesto de:

- ✓ Un único fichero HTML: `index.htm`.
- ✓ Una serie de ficheros js, donde hemos tenido el código.
- ✓ Una serie de ficheros .css donde se han escrito los estilos asociados a los elementos HTML que se tenían en la página.

Organización del código fuente:

En la carpeta raíz se tiene el único fichero HTML: `index HTML`.

En la carpeta js se tienen los ficheros *JavaScript*. Se hace la siguiente diferenciación:

- ✓ Código fuente libre copiado en nuestro servidor:
 - **leaflet.js** (Versión 0.4.4): Código libre que se ha bajado de <http://leafletjs.com/>. Es el fichero principal de *Leaflet*.
 - **l-google.js**: Código libre de un plugin de *Leaflet* que permite usar como fondo tiles de Google Maps. (Requiere Leaflet).
- ✓ Código fuente propio. Se divide en varios ficheros atendiendo a su funcionalidad:

- **cht-l.js:** Es el fichero *JavaScript* principal que contiene el código JavaScript de la aplicación y que controla cómo se representan las presiones, cómo se editan, borran, etc.
- **bocadillo.js:** Implementa el funcionamiento del popup o ventana emergente. Mostrado de datos. Botones de acciones (borrar y editar)
- **formularios.js:** Implementa el funcionamiento del formulario, validación de datos, rellenado del combo “grupo”, selección de tipo de geometría y activación del modo de edición de geometrías.
- **grupos.js:** Contiene información sobre los grupos en los que pueden clasificarse las presiones. Hace también de “tabla de grupos” puesto que contiene tanto los códigos como las descripciones de cada grupo o subgrupo. Contiene el código que da “estilo” a las presiones puntuales (por estilos css), a las presiones lineales y a las superficiales. En este fichero se establecen los colores a usar por cada tipo de presión.

En la carpeta *css* se tienen los ficheros de estilo y algunas imágenes para determinados botones.

* *leaflet.css:* Contiene los estilos usados por los elementos html creados por *leaflet.js*

* *style.css:* Contiene los estilos básicos de la aplicación.

* *bocadillo.css.* Contiene los estilos de los elementos del bocadillo.

* *formulario.css.* Contiene los estilos de los elementos del bocadillo.

Dentro de *css/images* se han guardado tres imágenes que han sido usadas para los estilos de *leaflet.css: zoom-in, zoom-out y layers.*

Es necesario indicar que han sido utilizados otros ficheros *JavaScript* pero que no han estado alojados en el servidor del cliente, sino que se han descargado de un servidor externo.

jQuery: <http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js>

Código fuente libre alojado en Google. Nos hemos descargado la versión “comprimida”, pero también está disponible la versión “legible”.

GoogleMaps: <http://maps.google.com/maps/api/js?v=3.2&sensor=false>

Código fuente de uso libre. Sirve para cargar otros servicios de Google como el de “tiles” para los mapas. Se ha cargado porque lo requiere el plugin de *leaflet*: l-google del que se tiene copia en nuestro servidor.

Plugins propios son Formulario(formulario.js), Bocadillo (bocadillo.js) y Tabla.

Tabla (tabla.js): utiliza código de otros *plugins*. Implementada como un *widget* de *JavaScript*

A continuación se muestra una captura desde el proyecto web.

La captura se he hecho abriendo el proyecto desde *WebStorm*.¹⁰

¹⁰ Al abrir el proyecto (o al abrir una carpeta donde se tiene código html, js etc) con *Webstorm*, aparece una nueva carpeta “.idea”. *WebStorm* es un ide para aplicaciones web.

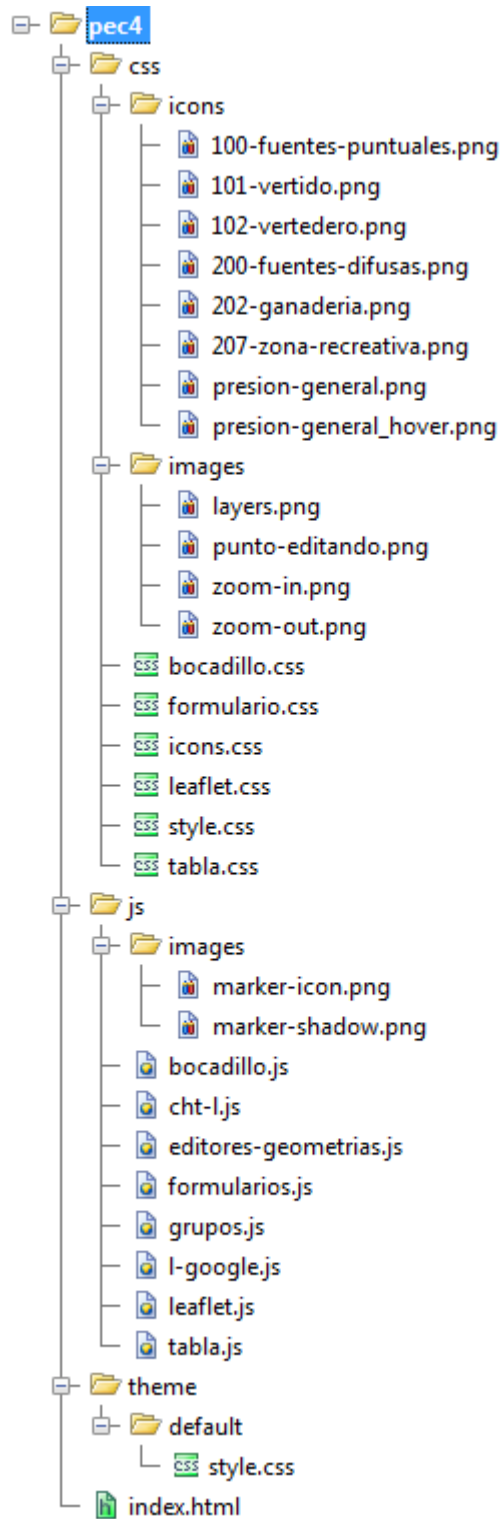


Ilustración 31. Carpetas y ficheros del proyecto. Fuente: Elaboración propia.

7. CONCLUSIONES

Una vez finalizado el proyecto final de carrera, me gustaría indicar que se ha tratado de un trabajo de tipo "investigación". A diferencia de lo que sucede normalmente con los proyectos de desarrollo de software, por falta de tiempo, no hemos diseñado un plan de pruebas para realizar las pruebas unitarias y de integración. Los prototipos realizados al comienzo del trabajo es lo más parecido a pruebas que hemos realizado. Se ha dedicado más tiempo a probar las diferentes técnicas de desarrollo web: *JavaScript*, *Ajax*, estilos *css*, geoconsultas, manejo de mapas, etc.

Entre los inconvenientes encontrados, el principal se centra en que *JavaScript*, como lenguaje interpretado es más costoso en tiempo e inseguro. Lo cual supone que los errores producidos con este lenguaje son más difíciles de detectar. Para poder detectarlos es necesario realizar pruebas exhaustivas que sustituyan el proceso de compilación de los lenguajes como *Java* o *C#*.

Entre las ventajas destacan la existencia de librerías de uso gratuito como *GoogleMaps* que disponen de multitud de ejemplos y ofrece foros donde explican el uso de sus *APIs*. También existen librerías de uso gratuito cuyo código fuente es accesible, es el caso de *jQuery* y *Leaflet*.

También como ventaja hay que reseñar que la tecnología de *plugins* tanto para *Leaflet* como para *jQuery* facilita el desarrollo de aplicaciones. Además al disponer de código fuente, es posible mejorarlos o corregirlos.

8. NUEVAS LÍNEAS DE DESARROLLO

Sería interesante una valoración global de *CartoDB* como alternativa a otros sistemas 'tradicionales' y especialmente para su utilización por las Administraciones Públicas como es el objeto de nuestro proyecto.

Igualmente, la integración con otras aplicaciones de la CHT para continuar desarrollándola conforme a las necesidades que puedan ir surgiendo.

En una próxima versión deberíamos incluir nuevas funcionalidades como ampliación de capas conectándose a orígenes de datos externos, hacer funcional campos fecha-hora, incluir los requisitos opcionales del proyecto.

Desarrollo adaptado a dispositivos móviles

GLOSARIO

AJAX. Acrónimo de *Asynchronous JavaScript And XML* (*JavaScript* asíncrono y *XML*). Es una técnica de desarrollo web para crear aplicaciones interactivas o *RIA* (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

API. *Application Programming Interface* o Interfaz de Programación de Aplicaciones. Conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca (o librería), para ser utilizado por otro software como una capa de abstracción.

CCAA: Comunidades Autónomas.

CCHH: Confederaciones Hidrográficas.

CDN. (Tecnología- *CDN*) es una red de servidores que replican el contenido de una web entre varias máquinas, hacen que esté siempre disponible y sea accesible de una forma más rápida.

Si tenemos un blog alojado en un hosting todo su contenido será servido por éste, usando su propio ancho de banda. Si utilizamos un *CDN* al estar el contenido repartido entre varios servidores la carga y el ancho de banda se reparten, por lo que es menos probable que el hosting se sature. Así podremos asegurar que no se caiga el servidor o que el tiempo de respuesta sea más rápido.

Los servicios proporcionados por un *CDN* mejoran el acceso a contenido especializado al abordar tres áreas básicas en el ámbito del networking: velocidad, fiabilidad y escalabilidad.

Normalmente las *CDN* son de pago (Akamai, Amazon Cloud Front) pero hay una *CDN* gratuita CoralCDN.

CGI (Common Gateway Interface). El sistema más antiguo para servir páginas dinámicas que permite la ejecución en el servidor de programas hechos en *C++*, *Visual Basic*, *Perl*, *Python*... Un servidor de mapas muy extendido como es *Mapserver* trabaja con esta tecnología.

CHT: Confederación Hidrográfica del Tajo.

CSV. *Comma Separated Values*. Fichero de texto con valores separados por comas.

DMA: Directiva Marco de Agua.

GML. *Geography Markup Language* o Lenguaje de Marcado Geográfico, descrito como una gramática en XML *Schema* para el modelaje, transporte y almacenamiento de la información geográfica.

GeoJSON. Es un formato para la codificación de una variedad de estructuras de datos geográficos. Un objeto GeoJSON puede representar una geometría, una característica o un conjunto de características. GeoJSON soporta los siguientes tipos de geometría: punto, *LineString*, *Polygon*, *MultiPoint*, *MultiLineString*, *MultiPolygon* y *GeometryCollection*. Características de GeoJSON contener un objeto de geometría y propiedades adicionales, y una colección de fenómenos representa una lista de características. Implementa el funcionamiento del *popup* que aparece al hacer clic en alguna presión (punto, línea, o área).

GPX. *GPS eXchange format*, formato XML para el intercambio de datos GPS (*waypoints*, *tracks*, *routes*) entre aplicaciones. Hoy en día es el formato más empleado para el intercambio de información entre dispositivos GPS.

GvSIG es un proyecto de desarrollo de Sistemas de Información Geográfica en software libre. Herramienta de escritorio para ver y editar cartografía.

jQuery es una biblioteca de *JavaScript*, , que permite simplificar la manera de interactuar con los documentos *HTML*, manipular el árbol *DOM*, manejar eventos, desarrollar animaciones y agregar interacción con la técnica *AJAX* a páginas web.

JSON es un formato ligero para el intercambio de datos, objetos de *JavaScript*.

JSP. La tecnología de páginas dinámicas de Java, dentro de la arquitectura general J2EE.

IMPRESS: Análisis de Presiones e Impactos y Evaluación del Riesgo.

INSPIRE. La Directiva 2/2007/EC para el establecimiento de una Infraestructura de Datos Espaciales en Europa.

KML. *Keyhole Markup Language*, Lenguaje XML utilizado inicialmente por *Google Earth* para representar datos geográficos en tres dimensiones empleado masivamente en la red para compartir información cartográfica, y que ya en 2008 se convirtió definitivamente en nuevo estándar aprobado por la *OGC*.

LISIGE. La Ley 14/2010, de 5 de julio, sobre las Infraestructuras y Servicios de Información Geográfica en España.

MAS: Masa de agua superficial.

NGINX. Es un servidor web / *proxy* inverso ligero de alto rendimiento y un *proxy* para protocolos de correo electrónico (*IMAP/POP3*).

Es software libre y de código abierto, licenciado bajo la Licencia *BSD* simplificada. Es multiplataforma, por lo que corre en sistemas tipo *Unix* (*GNU/Linux*, *Solaris*, *Mac OS X*, etc.) y *Windows*.

AUTH. *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.

OGC. (*Open Geospatial Consortium*). Organización internacional participada por empresas e instituciones del sector de la información geográfica enfocada a la definición de estándares de interoperabilidad (datos y procesos).

OOCC: Organismos de Cuenca.

OMA: Objetivo medioambiental

Perl. Lenguaje interpretado, con muchas funciones para el procesamiento de texto.

Python. Un lenguaje de script multipropósito, utilizado en ámbitos muy diversos (computación científica, geoprocésamiento, administración de sistemas...).

PHP. Lenguaje originalmente diseñado para el desarrollo web para producir páginas web dinámicas libre y gratuito, muy extendido en páginas de foros y comunidades sociales.

PRESIÓN: cualquier actividad humana que incida sobre el estado de las aguas.

Quantum GIS (o QGIS) es un Sistema de Información Geográfica (SIG) de código libre. Herramienta para ver y editar cartografía.

REST API. *Representational State Transfer API*, es una API, o librería de funciones, a la que se accede por el protocolo HTTP, es decir, a través de direcciones web o URLs en las que enviamos los datos de nuestra consulta. Como respuesta a la consulta sobre el REST API se obtienen datos en diferentes formatos, como pueden ser texto plano, XML, JSON, etc.

REDIS. Es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes pero que opcionalmente puede ser usada como una base de datos durable o persistente. Está escrito en ANSI C y esta liberado bajo licencia BSD por lo que es considerado software de código abierto.

SHAPEFILE. Formato de datos geográficos vectoriales de ESRI, formado por varios ficheros, siendo los principales los siguientes: *.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*.

SSL. *Secure Socket Layer*. Protocolo que proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, y cifrando la clave de sesión mediante un algoritmo de cifrado de clave pública.

WCS. *Web Coverage Service* (Servicio de Cobertura Web) proporciona una interfaz para realizar peticiones de cobertura geográfica a través de la web utilizando llamadas independientes de la plataforma.

WFS. *Web Feature Service*, proporciona la información relativa a la entidad almacenada en una capa vectorial (*Features*) que reúnen las características formuladas en la consulta.

WMS. *Web Map Service*, produce mapas en formato imagen a la demanda para ser visualizados por un navegador web o en un cliente SIG.

Se suele contraponer “cliente ligero” (navegador web) vs. “cliente pesado” (aplicación SIG de escritorio).

WPS. *Web Processing Service*. Servicio del OGC para la ejecución remota de geoprocetos.

ANEXO I. BIBLIOGRAFÍA

ARROYO, L. A. La tecnología de los sistemas de información geográfica en el uso de la tierra. Departamento de Suelos, Ministerio de Agricultura y Ganadería. http://www.mag.go.cr/congreso_agronomico_xi/a50-6907-I_237.pdf

AULA TI. BASES DE DATOS GEOGRÁFICAS. Sistemas de Información Geográfica, una realidad para todos. 2009. <http://www.aulati.net/tag/bases-de-datos-geograficas/>

BATALLER DÍAZ, A. Gestión y Desarrollo de Proyectos. (P08/89018/00444).

BOTELLA PLANA, A. Bases de Datos Geográficos. UOC (P07/89036/02929).

BOTELLA PLANA, A. Nuevas Tendencias en SIG. UOC (P07/89036/02932).

BURROUGH, P.A., 1986. Principles of Geographical Information Systems for Land Resources Assessment. Monographs on soils and resources survey, N°12, Oxford University, New York, U.S.A. 194 p.

CLARISÓ, R. Orientaciones Generales para el Trabajo Final. (UOC).

DE LA TORRE ALONSO, J., ÁLVAREZ LEIVA, S. y SANTANA, J. *CartoDB*: Base de datos geoespacial para el desarrollo fácil de aplicaciones web. <http://www.youtube.com/watch?feature=endscreen&NR=1&v=yHhUDE68bx0>

FERNÁNDEZ LÓPEZ, G. Y MÉNDEZ PÉREZ, X. (2011). Curso de Introducción a OpenLayers. ETS de Camiños, Canais e Ports. Universidade de A Coruña, Campus de Elviña. <http://www.slideshare.net/sonxurxo/curso-openlayers-xeoinquedos>

GUARDADO, I. Tutoriales (2012): Aplicaciones de mapas con CartoDB. <http://web.ontuts.com/tutoriales/aplicaciones-de-mapas-con-cartodb/>

LONGLEY, P.A.; GOODCHILD, M.F.; MAGUIRE, D.J., y otros (2005). *Geographic Information Systems and Science* (2ª. Ed. –ed-original, 2001-). Chichester (Reino Unido): John Wiley & Sons.

MINISTERIO AGRICULTURA, ALIMENTACIÓN Y MEDIO AMBIENTE.
Confederación Hidrográfica del Tajo: Memoria SAICA 2009-2011.

MINISTERIO DE MEDIO AMBIENTE. DIRECCIÓN GENERAL DEL AGUA. SECRETARÍA GENERAL PARA EL TERRITORIO Y LA BIODIVERSIDAD. Manual para la identificación de las presiones y análisis del impacto en aguas superficiales.http://www.magrama.gob.es/es/agua/publicaciones/impress_tcm7-29158.pdf

MONTAGUT, R. B. Presentación de documentos y elaboración de presentaciones. (P08/89018/00446).

MUÑOZ BOLLAS, A. Geodesia y Cartografía.(P07/89036/02928).

RODRIGUEZ LLORET, J. y OLIVELLA, R. Introducción a SIG y Geotelemática. UOC (P07/89036/02927). Introducción a los Sistemas de Información Geográfica. UOC (P07/89036/02930).

SÁENZ HIGUERAS, N. y VIDAL OLTRA, R. Redacción de textos científico técnicos (P08/89018/)

UOC (2012). Aula virtual. Tablón y foro del TFC. Enunciado del TFC “Aplicación Web para el inventario de presiones en ríos con CartoDB.”

UOC. “Trabajos y Proyectos Finales de Carrera en Sistemas de Información Geográfica”. Repositorio O2 de la Biblioteca Virtual de la UOC, <http://openaccess.uoc.edu/webapps/o2/handle/10609/36>

VI Jornadas de SIG Libre. Girona. (2012). [en línea]
<http://www.sigte.udg.edu/jornadassiglibre2012/programa/jornadas>

VELARDE GUTIÉRREZ, V. (2011). Curso de experto en desarrollo y gestión de Sistemas de información geográfica (31-EX-75). Desarrollo de aplicaciones geográficas web: OpenLayers (11-M4.2).Universidad de Cantabria.

WM LELER. (2012).The top seven alternatives to the Google Maps API.
<http://www.netmagazine.com/features/top-seven-alternatives-google-maps-api>

ANEXO II.PLAN DE GESTIÓN DE RIESGOS

Riesgos	Medidas preventivas
Avería en la estación del trabajo	Se intentará conseguir otro PC El código fuente: desplegada la aplicación de <i>dropbox</i> , a través de site 44, carpeta principal compartida. Copia de seguridad en dispositivo externo.
Baja por enfermedad y/o problemas familiares	Incorporarse lo antes posible Recuperar el tiempo perdido trabajando en los días que no está previsto dedicar horas al proyecto
Efectuar horas extra en el trabajo	Recuperar el tiempo perdido trabajando en los días que no está previsto dedicar horas al proyecto.
No alcanzar los objetivos mínimos	Pedir ayuda al consultor Corregir los errores
Caída del servidor principal(sin acceso a la aplicación)	Se tratará en el foro del aula de la asignatura
Caída de los servidores de Google (no visualización fondos de mapas, ni dibujar gráficos)	Se tratará en el foro del aula de la asignatura
Caída de los servidores de <i>CartoDB</i> (se muestra mapa pero sin información hidrográfica)	Se tratará en el foro del aula de la asignatura
Uso de navegadores antiguos	La versión mínima del navegador se mostrará en la interfaz, para que los usuarios de la aplicación puedan navegar sin problemas.