



Open Atlas del Sistema de Sanidad Pública de Cataluña

Rafael Méndez Martínez
Máster en Ingeniería de Informática

Anna Muñoz Bolas

11 de Junio de 2015



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Open Atlas del Sistema de Sanidad Pública de Cataluña
Nombre del autor:	Rafael Méndez Martínez
Nombre del consultor:	Anna Muñoz Bollas
Fecha de entrega (mm/aaaa):	06/2015
Área del Trabajo Final:	Sistemas de Información Geográfica
Titulación:	<i>Master en Ingeniería de Informática</i>
Resumen del Trabajo:	
<p>La mayoría de las personas han utilizado alguna vez un mapa geográfico para responder a preguntas relacionadas con la localización de objetos o de otras personas. Responder a este tipo de cuestiones ha sido una necesidad que se remonta a los tiempos en que vivíamos en cuevas y pintábamos sobre la roca objetos y humanos en distintas escenas y lugares.</p> <p>Actualmente responder preguntas ayudados con un mapa sigue siendo una necesidad, que conseguimos satisfacer de manera más precisa y fiable gracias a los avances, en particular, de las tecnologías de la información y la comunicación (TIC). Las TIC han impulsado el desarrollo de los conocimientos aplicados en geodesia.</p> <p>El uso de la geoinformación se ha extendido por todos los sectores de nuestra sociedad: transporte, urbanismo, sanidad, consumo, etc. Internet, los sistemas de posicionamiento global (Global Position System. GPS), los avances en los sistemas de información geográfica (SIG) y los dispositivos móviles inteligentes (smartphones, tablets, PDA, etc.) han popularizado el uso de la información geográfica. En esencia, el concepto de geoinformación de hoy es el mismo que se plasmaba en los mapas temáticos convencionales, pero con todo el valor añadido de los avances tecnológicos mencionados.</p> <p>Este trabajo está orientado al desarrollo de un atlas geo-sanitario que permita visualizar información sanitaria geolocalizada y dé respuestas a cuestiones relacionadas con una serie de indicadores de salud del sistema sanitario de Cataluña.</p>	

Abstract

Most people have ever used a geographical map to answer questions related to the location of objects or other people. Answer to such questions has been a need that goes back to the days when we lived in caves and we painted on the rock objects and humans in different scenes and places.

Currently answer questions helped with a map remains a need, we get to meet more accurately and reliably thanks to advances in particular the information and communications technology (ICT). ICT has boosted the development of the knowledge applied in geodesy.

The use of geoinformation has spread across all sectors of our society: transport, urban planning, health, consumption, etc. Internet, global positioning systems (Global Position System. GPS), advances in geographic information systems (GIS) and smart mobile devices (smartphones, tablets, PDA, etc.) have popularized the use of geographic information. In essence, the concept of geodesic information today is the same as that embodied in conventional thematic maps, but with all the added value of the technological advances.

This work is aimed at developing a geo-health atlas that allows visualizing geographic health information and answers to issues related to a number of health indicators of the health system in Catalonia

Palabras clave:

Geoinformación, mapas, posicionamiento global, atlas geosanitario, geolocalización, Sistemas de Información Geográfica y Geodesia

INDICE DE CONTENIDOS

1. INTRODUCCION.....	4
1.1. CONTEXTO Y JUSTIFICACION.....	4
1.2. OBJETIVOS Y ALCANCE.....	5
1.3. ENFOQUE Y METODO SEGUIDO.....	6
1.4. PLANIFICACION DEL TRABAJO.....	8
1.5. BREVE SUMARIO DE PRODUCTOS OBTENIDOS.....	16
1.6. ORGANIZACIÓN DE CONTENIDOS DE ESTA MEMORIA.....	16
2. ANTECEDENTES.....	20
3. ANALISIS DE SOLUCIONES.....	21
3.1. SERVIDOR DE MAPAS.....	21
3.2. SERVIDOR DE DATOS.....	25
3.3. SERVIDOR WEB.....	27
3.4. VISOR WEB.....	28
4. DISEÑO.....	31
4.1. DATOS.....	31
4.2. VISOR WEB.....	44
4.3. SERVIDOR DE MAPAS.....	54
5. PRUEBAS Y RESULTADOS FINALES.....	58
6. CONCLUSIONES Y MEJORAS.....	61
7. GLOSARIO.....	63
8. BIBLIOGRAFIA Y REFERENCIAS.....	67
9. ANEXO I: PLANIFICACIÓN DE TAREAS DEL PLAN DE TRABAJO.....	68
10. ANEXO II: GUIA DE USUARIO.....	70

INDICE DE FIGURAS

Figura 1 : mapa de actividades y valores en metodologías ágiles, XP.....	7
Figura 2: diagrama de Gantt para las tareas del bloque 1.....	12
Figura 4: diagrama de Gantt para las tareas del bloque 2.....	12
Figura 4: diagrama de Gantt para las tareas del bloque 3.....	13
Figura 5: diagrama de Gantt para las tareas del bloque 4.....	14
Figura 6: diagrama de Gantt para las tareas del bloque 5.....	15
Figura 8: arquitectura básica de una aplicación en MapServer.....	24
Tabla 1: comparativa de características básicas de los servidores web analizados.....	28
Figura 9: interfaz de usuario del visor web.....	45
Figura 10: interfaz de usuario para calcular la cobertura asistencial.....	46

Figura 11: interfaz de usuario para calcular el riesgo de epidemias.....	46
Figura 12: interfaz de usuario para calcular el gasto sanitario	46
Figura 13: interfaz de usuario con información sobre los símbolos y leyendas del mapa	47
Figura 14: representación gráfica de las áreas de influencia de los centros sanitarios mediante diagramas de Voronoi	48
Figura 15: estructura de ficheros del patrón MVC	51
Figura 16: diagrama general del patrón Proxy.....	52
Figura 17: ficheros PHP del patrón Proxy.....	52
Figura 18: ubicación de los ficheros de datos geoespaciales de las regiones sanitarias.	53
Figura 19: visualización del mapa en un navegador.	57
Figura 20: elipsoide y geoide.....	64
Figura 21: latitud y longitud geográfica.....	65
Figura 22: interfaz inicial del visor GIS, capa base topográfica	70
Figura 23: interfaz inicial del visor GIS, capa base ortográfica.....	70
Figura 24: selección del origen de datos geoespaciales, servidor de mapas local o servidor de mapas del ICGC.	71
Figura 25: manejo de capas base y capas de regiones sanitarias.	71
Figura 26: información de usuario para que seleccione una región sanitaria.....	72
Figura 27: ejemplo de obtención de gasto sanitario para la región sanitaria de Barcelona.....	72
Figura 28: ejemplo de obtención de gasto sanitario para la región sanitaria de Barcelona en el año 2013	73
Figura 29: visualización de gráficos estadísticos, gasto medio por habitante y año	73
Figura 30: información de usuario para que seleccione un tipo de enfermedad contagiosa.	74
Figura 31: cálculo del riesgo de epidemia para 'la Clamidia' en la región sanitaria de Barcelona.	74
Figura 32: cálculo del riesgo de epidemia para la 'Clamidia' en la región sanitaria de Barcelona, año 2014.....	75
Figura 33: gráficos estadísticos sobre el riesgo relativo de padecer una epidemia de 'Clamidia' en la región sanitaria de Barcelona.....	75
Figura 34: cálculo de la cobertura asistencia en la región sanitaria de Barcelona.	76
Figura 35: gráficos estadísticos sobre la cobertura asistencial por tipo de hospitalización en la región sanitaria de Barcelona.....	76
Figura 36: ejemplo de geolocalización, búsqueda de la ciudad de 'Mataró'.....	77
Figura 37: ejemplo de geolocalización, resultado de la búsqueda de la ciudad de 'Mataró'	77

Figura 38: búsqueda del centro hospitalario más cercano a localización geográfica elegida por el usuario.....	78
Figura 39: información sobre las leyendas del mapa.....	78

1. INTRODUCCION

La mayoría de las personas han utilizado alguna vez un mapa geográfico para responder a preguntas relacionadas con la localización de objetos o de otras personas. Responder a este tipo de cuestiones ha sido una necesidad que se remonta a los tiempos en que vivíamos en cuevas y pintábamos sobre la roca objetos y humanos en distintas escenas y lugares.

Actualmente responder preguntas ayudados con un mapa sigue siendo una necesidad, que conseguimos satisfacer de manera más precisa y fiable gracias a los avances, en particular, de las tecnologías de la información y la comunicación (TIC). Las TIC han impulsado el desarrollo de los conocimientos aplicados en geodesia.

El uso de la geoinformación se ha extendido por todos los sectores de nuestra sociedad: transporte, urbanismo, sanidad, consumo, etc. Internet, los sistemas de posicionamiento global (Global Position System. GPS), los avances en los sistemas de información geográfica (SIG) y los dispositivos móviles inteligentes (smartphones, tablets, PDA, etc.) han popularizado el uso de la información geográfica. En esencia, el concepto de geoinformación de hoy es el mismo que se plasmaba en los mapas temáticos convencionales, pero con todo el valor añadido de los avances tecnológicos mencionados.

Este trabajo está orientado al desarrollo de un atlas geo-sanitario que permita visualizar información sanitaria geolocalizada y dé respuestas a cuestiones relacionadas con una serie de indicadores de salud del sistema sanitario de Cataluña.

1.1. CONTEXTO Y JUSTIFICACION

El contexto de este trabajo se puede definir desde tres ámbitos distintos:

- Lugar de donde procede la información y donde tendrá aplicación los resultados finales. La información está definida en el contexto sanitario y se obtendrá de indicadores sanitarios sobre los principales temas relacionados con el estado de salud de la población de Cataluña.
- Áreas principales de conocimiento sobre las que se desarrolla. La principal área de conocimiento es la geodesia, dada la extensión de la misma sólo se trabajará con conceptos ligados a los objetivos que se pretenden alcanzar. En el Anexo I se incluyen los conceptos fundamentales de utilidad para realizar este trabajo.
- Tecnologías de la información. Es el ámbito más importante de este trabajo y sobre el que se proyectan los dos ámbitos anteriores. Se reflejarán algunas de las técnicas, metodologías y conceptos específicos aprendidos durante el Master de Ingeniería de Informática, que culmina con la realización satisfactoria de este trabajo.

Como se decía en la introducción la necesidad de ofrecer información y respuestas en un contexto geográfico está cada vez más presente. La comprensión de la información sanitaria dentro de un contexto geográfico facilita la realización de estudios y análisis estadísticos. El atlas geo-sanitario se desarrollará bajo licencia no comercial, para que pueda ser utilizado como una

herramienta de consulta abierta por los ciudadanos y el personal sanitario. Responder todas las preguntas posibles en el ámbito de información indicado supondría un objetivo cuyo alcance queda fuera de este trabajo. Por ello, se ofrecerán respuestas a cuestiones relacionadas con la cobertura asistencial, el gasto público y la vulnerabilidad de la población ante epidemias.

1.2. OBJETIVOS Y ALCANCE

Los objetivos indicados para este trabajo se han clasificado en generales y específicos.

Los objetivos generales son:

1. Diseñar e implantar un servidor de mapas.
2. Diseñar e implementar un visor GIS web.
3. Integrar datos ráster vectoriales, ya sean cartográficos o alfanuméricos, procedentes de diversas fuentes.

Los objetivos específicos son:

1. Conocer a fondo alguno de los servidores de mapas open source (MapServer, Geoserver, etc.).
2. Analizar las diferentes librerías JavaScript que permiten desarrollar un visor web con componente geográfica (Leaflet, D3, Turf, ...).
3. Visualizar de manera dinámica información estadística de los datos.
4. Seleccionar de manera interactiva las diferentes capas de datos con los indicadores de salud que se quieran visualizar.

El resultado visible, una vez cumplidos los objetivos, desde la perspectiva del usuario final es un visor web que le permite realizar consultas basadas en indicadores sanitarios. Entre otras funciones podrá:

- Conocer el gasto sanitario por zonas.
- Conocer la cobertura asistencial.
- Conocer las zonas de influencia de los distintos centros sanitarios.
- Conocer el riesgo de cada zona geográfica de padecer una epidemia.
- Calcular el tiempo de desplazamiento mínimo al centro sanitario más cercano.

Una vez definido el contexto y los objetivos conviene definir el alcance de este trabajo. La gestión del alcance es uno de los componentes más críticos en la gestión de un proyecto TIC. En términos de requisitos el alcance de este trabajo quedaría definido por:

- Disponer de fuentes de datos relacionadas con indicadores de salud en el sistema sanitario de Cataluña.

- Disponer de fuentes de datos cartográficos open source.
- Integrar la estructura y tipos de acceso de las distintas fuentes de información en un componente software con una interfaz bien definida.
- Desarrollar un componente software especializado en el tratamiento de datos alfanuméricos cuya interfaz permita la realización de consultas (estadísticas, cálculo, selección, etc.). El diseño de este componente estará ligado al componente anterior.
- Implantar un servidor de mapas compatible con las fuentes de datos cartográficos seleccionados.
- Implementar un visor web en el que destaquen la eficacia, la eficiencia, la usabilidad y la calidad de la información, entre otras características.

Las tareas que se llevarán a cabo para cumplir con el alcance están planificadas en el EDT¹ del apartado 5. La validación del alcance se realizará de manera cíclica tras la finalización de cada tarea o bloque de las mismas, aplicando la metodología habitual en el desarrollo de software: análisis, diseño, codificación, pruebas y validación.

1.3. ENFOQUE Y METODO SEGUIDO

En todo momento se ha intentado seguir un enfoque simplista dadas las limitaciones temporales y el conocimiento medio-bajo del tema de este trabajo.

La metodología seguida se puede dividir en dos partes: metodología del trabajo y metodología de desarrollo. La metodología de trabajo ha consistido en la planificación y realización de tareas definidas en base a los objetivos planteados. De manera resumida esta metodología ha consistido en:

- Estudio y recopilación de información relacionada con el tema de este trabajo.
- Análisis y evaluación de contenidos para la toma de decisiones de cara a seleccionar la mejor solución.
- Planificación de trabajos.
- Elaboración de documentación y entregables

Hay que decir que el no disponer de un equipo de proyecto ha condicionado la elección de la metodología de desarrollo. Se ha seguido la metodología de desarrollo ágil denominada XP (eXtreme Programming) por su flexibilidad y adaptabilidad a los cambios. El equipo de proyecto ha estado formado por dos personas: la coordinadora de este trabajo y el autor del mismo. Se ha intentado, en la medida de lo posible, adoptar en cada momento los diferentes roles que intervienen en la metodología elegida (cliente, programador, tester, checker, etc.)

La idea base de esta metodología es plantearse el software como una solución ágil y no como proyectos arquitectónicos. Esta metodología fue creada por Kent

¹ Estructura de Descomposición del Trabajo

Beck, Ward Cunningham y Ron Jeffries a finales de los noventa, la programación extrema ha pasado de ser una simple idea para un único proyecto a inundar todas las "factorías de software".

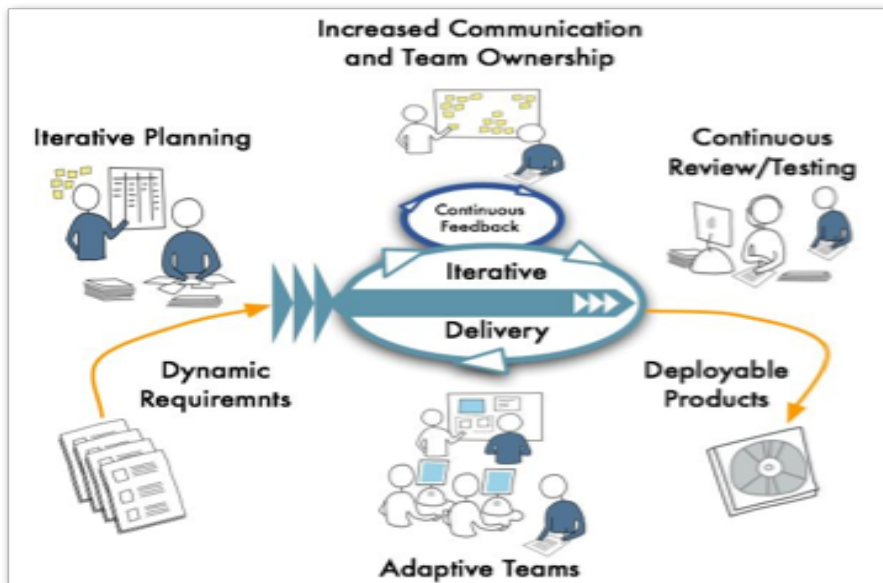


Figura 1 : mapa de actividades y valores en metodologías ágiles, XP

Los 4 valores clave en metodología ágil (figura 1) surgen a partir de sus 12 principios básicos, son los siguientes:

1. Los individuos y sus interacciones (Increased Communication and Team Ownership): los individuos e interacciones son esenciales para los equipos de alto rendimiento. Diversos estudios sobre la comunicación en equipos de trabajo han demostrado que cuando no existe ningún problema de comunicación, los equipos pueden rendir 50 veces más que el promedio. Para facilitar la comunicación, las metodologías ágiles confían en ciclos frecuentes de inspección y adaptación. Estos ciclos pueden variar de cada pocos minutos con programación por parejas, cada pocas horas con integración continua o todos los días con una reunión de puesta en marcha, hasta cada iteración con una revisión y retrospectiva.
2. Entregar software que funciona (Deployable products): es una de las grandes diferencias que aporta el desarrollo ágil. En todas las metodologías ágiles representadas por el *Manifiesto Ágil* destacan los entregables que funcionan y que se van proporcionando periódicamente al cliente, de mayor importancia que la documentación que se genera durante el proyecto. ¿Qué significa que el software funciona? Para equipos ágiles esto supone que la funcionalidad del entregable cumple con todas las características y ha pasado todas las pruebas, por tanto es aceptado por el usuario final y puede ser utilizado. Las entregas parciales del producto permiten evaluar la aceptación del cliente y facilitan la adaptación al cambio. Además las entregas cada vez se realizan en un periodo de tiempo menor, disminuyendo los costes parciales sin reducir el nivel de calidad.

3. Colaboración con el cliente (Interactive Planing): las tasas de éxito de los proyectos han aumentado gracias a las mejoras introducidas por metodologías ágiles como XP. Los problemas se abordan en proyectos de dimensiones manejables y con entregas frecuentes, lo que permite que los clientes se integren en un plan interactivo y aporten comentarios sobre el software. Una colaboración que va más allá de la negociación de un contrato.
4. Responder al cambio (Adaptative teams and Continuous review/testing): más de la mitad de los requisitos de un producto o proyecto cambian durante el desarrollo. Incluso cuando el producto está terminado, el cliente insatisfecho, sigue proponiendo cambios porque no tenía muy claro lo que quería hasta que lo ha visto funcionando (*Ley Humphrey*). Las metodologías ágiles como XP integran al cliente en el proyecto para que pueda incorporar los comentarios y nuevos requisitos a medida que se desarrolla el producto. No se sigue un plan estricto que dificulte o imposibilite el cambio.

El trabajo en metodología ágil se planifica en torno a tareas sencillas que se llevan a cabo en iteraciones. Cada iteración ha generado un resultado que ha sido esencial para trabajar de manera progresiva y acumulativa. Las iteraciones han quedado enmarcadas dentro de la planificación de este trabajo.

1.4. PLANIFICACION DEL TRABAJO

En este apartado se describirá la planificación de las tareas que se han llevado, nos servirá de guía la planificación propuesta en el plan de trabajo inicial (Anexo III). Se han mantenido las tareas incluidas en el plan de trabajo y se han añadido otras nuevas como subtareas de las tareas resumen de cada bloque.

La planificación se realizará con la ayuda de herramientas específicas como Microsoft Project (con licencia de estudiante). Como resultado de la planificación se obtendrá una tabla con la organización de las tareas (EDT) y un diagrama de Gantt. Además se justificarán aquellas tareas que son críticas en el desarrollo de este trabajo.

TAREAS

En la tabla 1 se muestran las tareas previstas para desarrollar este trabajo.

Nótese que las tareas correspondientes a la redacción de la memoria tienen una duración prevista de una hora, se realiza de manera transversal en cada bloque. La duración del resto de tareas se indica en horas o en días con horario de 18:00 a 22:00.

La duración de las tareas 3.1 y 3.2 quizás tengan una estimación pesimista, se debe al menor conocimiento del tema relativo a los datos y el enfoque necesario para responder a preguntas asociadas con los mismos: influencia asistencial de un centro, grado de cobertura sanitaria en una región, gasto sanitario en una región, etc.

Las entregas de cada PEC se han definido como tareas y representan hitos en el desarrollo del proyecto. En cada PEC se realizará una entrega parcial de la

memoria con los contenidos que corresponden según el desarrollo del proyecto. Así mismo, si corresponde, se entregarán los prototipos para revisión de las pruebas y el código fuente de los mismos. Las horas estimadas por cada entrega se emplearán en preparar y revisar la información que se entrega.

Los hitos identificados se corresponden con las entregas periódicas de cada PEC.

- Tarea 1.2. Entrega del plan de trabajo
- Tarea 3.4. Entrega del borrador de la memoria en la que se incluyen los contenidos relacionados con las tareas desarrolladas hasta el momento.
- Tarea 5.2. Entrega del borrador de la memoria en la que se incluyen los contenidos relacionados con las tareas desarrolladas hasta el momento. Prototipo del servidor de mapas para pruebas y código fuente.
- Tarea 5.8. Entrega final. Se entregará la memoria final, el código fuente y la presentación virtual.

EDT	Nombre de tarea	Duración	Comienzo	Fin
1	Plan de trabajo	2 días	vie 06/03/15	mar 10/03/15
1.1	Elaboración del plan de trabajo	2 días	vie 06/03/15	vie 06/03/15
1.2	PEC1: Entrega del plan de trabajo (hito)	1 día	mar 10/03/15	mar 10/03/15
2	State of the art - Analizar la tecnología existente	7 días	mié 11/03/15	jue 19/03/15
2.1	Analizar las diferentes soluciones tecnológicas disponibles actualmente para resolver el TFM planteado	3 días	mié 11/03/15	vie 13/03/15
2.2	Seleccionar de manera justificada la tecnología a utilizar y describir con detalle las etapas de su implementación	4 días	lun 16/03/15	jue 19/03/15
2.3	Redacción de la memoria	7 días	mié 11/03/15	jue 19/03/15
3	Estructura de la información	15 días	vie 20/03/15	lun 13/04/15
3.1	Descripción de las diferentes fuentes de datos relativas a los indicadores de salud que se usaran en el proyecto	3 días	vie 20/03/15	mar 24/03/15
3.2	Búsqueda de datos cartográficos de tipo open source, como por ejemplo los de OpenStreetMap (OSM), o bien determinar las fuentes cartográficas que se usarán en el proyecto.	3 días	jue 26/03/15	lun 30/03/15
3.3	I. Análisis de la integración de las diferentes fuentes de datos en la tecnología seleccionada para la resolución del TFM	3 días	mié 01/04/15	mar 07/04/15
3.4	PEC2: Entrega del borrador de la	1 día	mar 07/04/15	mié 08/04/15

EDT	Nombre de tarea	Duración	Comienzo	Fin
(hito)	memoria en la que se incluyen los contenidos relacionados con las tareas desarrolladas hasta el momento			
3.5	II. Análisis de la integración de las diferentes fuentes de datos en la tecnología seleccionada para la resolución del TFM	3 días	mié 08/04/15	lun 13/04/15
3.6	Redacción de la memoria	15 días	vie 20/03/15	jue 13/04/15
4	Implementación del servidor de mapas	17 días	vie 10/04/15	mar 05/05/15
4.1	Análisis de los servidores de mapas Open Source y selección del más adecuado.	4 días	vie 10/04/15	mié 15/04/15
4.2	Implementación de máquina virtual	1 día	jue 16/04/15	jue 16/04/15
4.3	Implementación del servidor de datos geospaciales	0.5 días	vie 17/04/15	vie 17/04/15
4.3.1	Instalación y prueba de PostgreSQL	1 hora	vie 17/04/15	vie 17/04/15
4.3.2	Instalación de PostGIS	1 hora	vie 17/04/15	vie 17/04/15
4.4	Implementación del servidor de mapas con la tecnología seleccionada.	7.5 días	vie 17/04/15	mar 28/04/15
4.4.1	Instalación y prueba de Apache Web Server 2	1 hora	vie 17/04/15	vie 17/04/15
4.4.2	Instalación de Python y carga de repositorios de UbuntuGis	2 horas	vie 17/04/15	lun 20/04/15
4.4.3	Instalación y prueba de Mapserver	3 días	lun 20/04/15	jue 23/04/15
4.4.4	Instalación de Imposm	1 hora	jue 23/04/15	jue 23/04/15
4.4.5	Instalación de MapCache	2 horas	jue 23/04/15	jue 23/04/15
4.4.6	Creación de ficheros de scripts con información cartográfica	3 días	vie 24/04/15	mar 28/04/15
4.5	Modelado de datos alfanuméricos	1.75 días	mié 29/04/15	jue 30/04/15
4.5.1	Descargar los datos alfanuméricos relativos a los indicadores sanitarios de interés	1 hora	mié 29/04/15	mié 29/04/15
4.5.2	Seleccionar conjuntos de datos (ficheros en formato CSV)	2 horas	mié 29/04/15	mié 29/04/15
4.5.3	Crear la base de datos en PostgreSQL y las tablas correspondientes, uno por cada conjunto de datos	2 horas	mié 29/04/15	jue 30/04/15
4.5.4	Importar datos procedentes de los ficheros del conjunto de datos en las tablas correspondientes	2 horas	jue 30/04/15	jue 30/04/15
4.6	Pruebas de conjunto	2 días	lun 04/05/15	mar 05/05/15

EDT	Nombre de tarea	Duración	Comienzo	Fin
4.7	Redacción de la memoria	17 días	vie 10/04/15	mar 05/05/15
5	Implementación del visor web GIS	24 días	mié 06/05/15	lun 08/06/15
5.1	Análisis de las diferentes bibliotecas JavaScript que cumplen con los requisitos funcionales del proyecto I	3 días	mié 06/05/15	jue 08/05/15
5.2 (hito)	PEC3: Entrega del borrador de la memoria en la que se incluyen los contenidos relacionados con las tareas desarrolladas hasta el momento	1 hora	vie 08/05/15	vie 08/05/15
5.3	Descripción detallada de las funcionalidades que hay que implementar y diseño del visor web.	2 días	vie 08/05/15	mar 12/05/15
5.4	Implementación de las funcionalidades en el visor web GIS.	8 días	mar 12/05/15	vie 22/05/15
5.4.1	Diseñar la interfaz de usuario	2 días	mar 12/05/15	jue 14/05/15
5.4.2	Desarrollo de funciones en scripts de cliente para la interacción con la interfaz	2 días	jue 14/05/15	lun 18/05/15
5.4.3	Desarrollo de funciones en scripts de cliente para el tratamiento de contenidos cartográficos	2 días	lun 18/05/15	mié 20/05/15
5.4.4	Desarrollo de funciones en scripts de servidor	2 días	mié 20/05/15	vie 22/05/15
5.5	Pruebas de conjunto	3 días	lun 25/05/15	mié 27/05/15
5.6	Redacción de la memoria	19 días	mié 06/05/15	lun 01/06/15
5.7	Elaboración de la presentación virtual	4 días	mar 02/06/15	vie 05/06/15
5.8 (hito)	PEC4: Entrega final	1 día	vie 05/06/15	lun 08/06/15

Tabla 1: tareas programadas

DIAGRAMA DE GANNT

Mostrar un diagrama de Gannt en una hoja de tamaño folio es una tarea compleja a menos que el proyecto sea muy pequeño. Por ello ha decidido dividir el diagrama de Gannt en tantas partes como bloques de tareas existentes. A continuación se muestran los diagramas de Gannt separados por bloques:

1. *Bloque 1:* Plan de trabajo (06/03/2015 – 10/03/2015)

- *Hito:* PEC1, entrega parcial (10/03/2015)



Figura 2: diagrama de Gannt para las tareas del bloque 1

2. *Bloque 2:* State of the art – Analizar la tecnología existente (11/03/2015 – 19/03/2015)

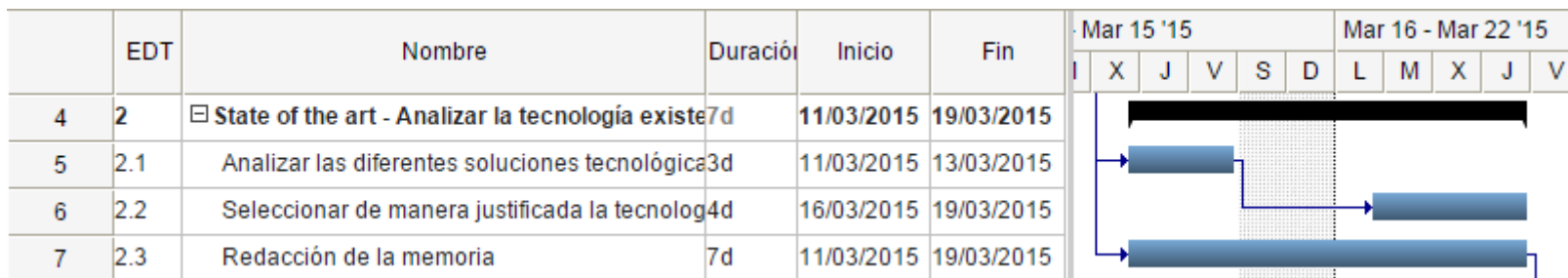


Figura 4: diagrama de Gannt para las tareas del bloque 2

3. *Bloque 3: Estructura de la información (20/03/2015 – 13/04/2015)*

- *Hito: PEC2, entrega parcial (08/04/2015)*

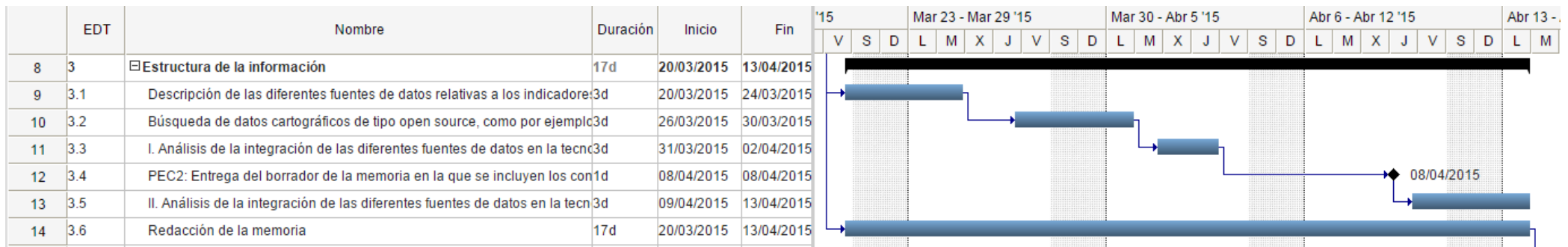


Figura 4: diagrama de Gantt para las tareas del bloque 3

4. *Bloque 4: Implementación del servidor de mapas*

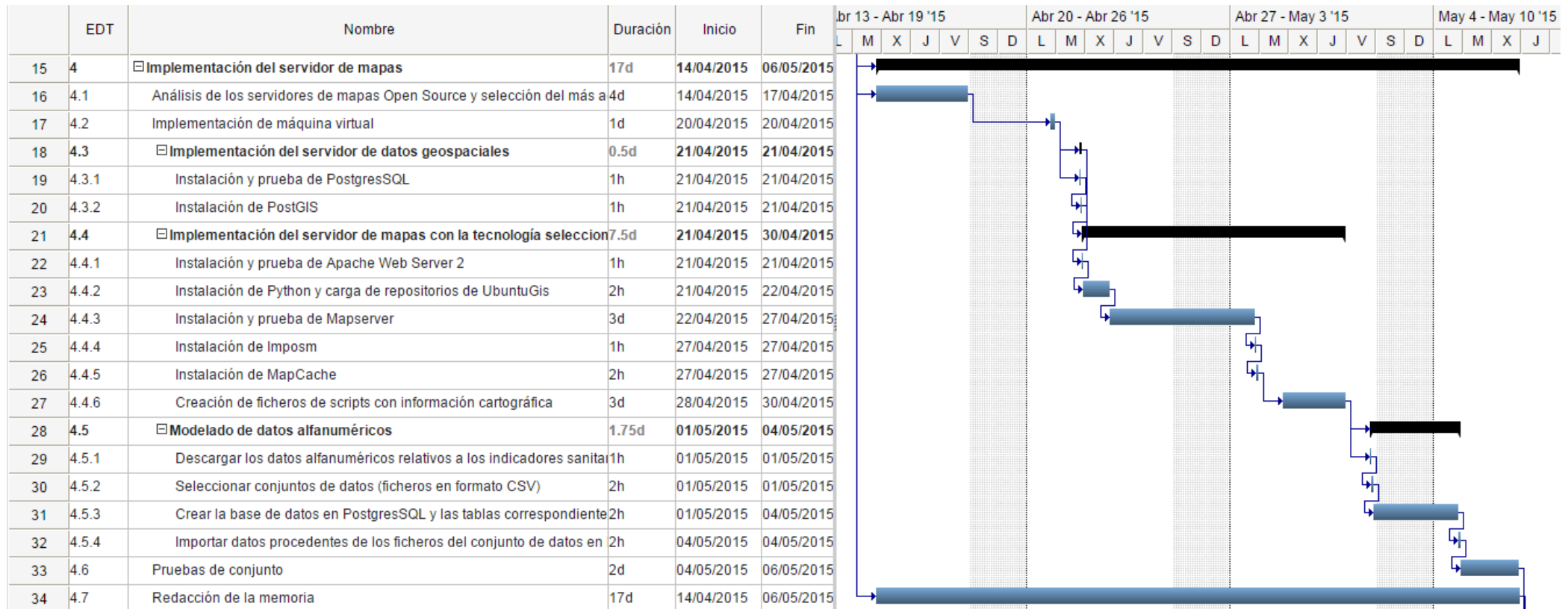


Figura 5: diagrama de Gannt para las tareas del bloque 4

5. *Bloque 5: Implementación del visor web GIS (06/05/2015 – 08/06/2015)*

- *Hito 1: PEC3, entrega parcial (12/05/2015)*
- *Hito 2: PEC4, entrega final (08/06/2015)*

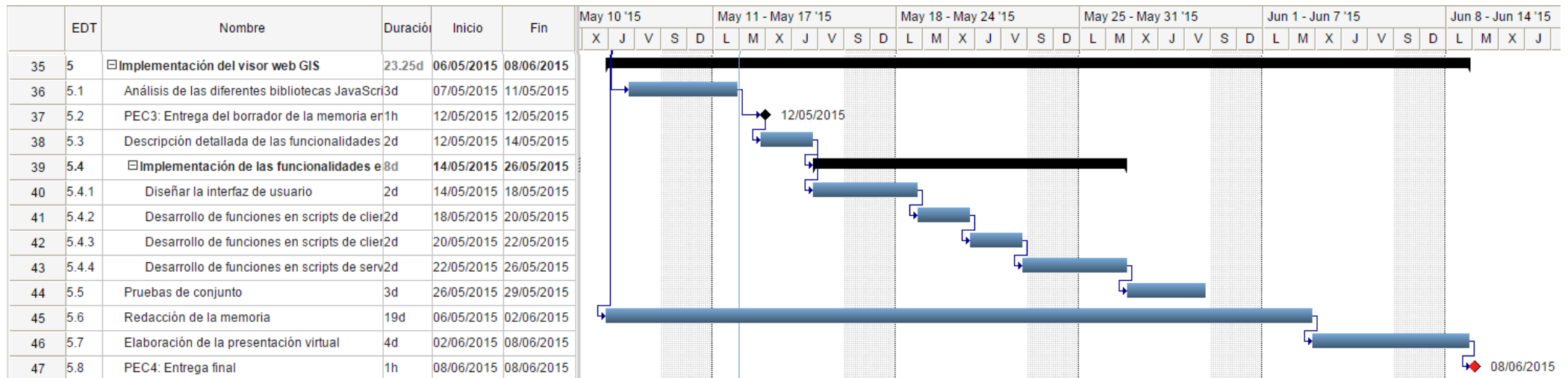


Figura 6: diagrama de Gantt para las tareas del bloque 5

1.5. BREVE SUMARIO DE PRODUCTOS OBTENIDOS

- Servidor de datos alfanuméricos en el contexto de la Sanidad Pública de Cataluña.
- Servidor de datos cartográficos y ficheros de configuración de mapas.
- Visor geográfico para la consulta y visualización de datos sanitarios. El visor puede utilizar como fuente de información cartográfica el servidor anterior o bien el servidor del ICGC.

1.6. ORGANIZACIÓN DE CONTENIDOS DE ESTA MEMORIA

Los contenidos de este trabajo se organizan en distintos apartados, en los que en todo momento se ha priorizado la claridad y sencillez en la redacción. Los apartados se identifican con el texto en mayúscula, negrita y numeración. La numeración utilizada es jerárquica en dos niveles: apartado y subapartado.

A continuación se enumeran y describen los principales apartados de este trabajo:

1. INTRODUCCION

Es el primer apartado, por lo que la intención del mismo es introducir las principales características de este trabajo. Se divide en seis subapartados:

1.1. CONTEXTO Y JUSTIFICACION

Este apartado está orientado a establecer el contexto de definición del presente trabajo. Se fijan los ejes fundamentales del contexto a partir de distintos ámbitos relacionados de información.

Después de introducir el contexto se justifica la realización de este trabajo tomando como idea principal lo descrito en el apartado de introducción.

1.2. OBJETIVOS Y ALCANCE

Una vez se ha fijado el contexto y justificado el porqué de la realización de este trabajo, es más fácil definir los objetivos y el alcance de los mismos. En este apartado se enumeran y describen tanto los objetivos generales como los específicos. Además se explican brevemente los resultados esperados para, de este modo, establecer el alcance dentro del contexto definido.

1.3. ENFOQUE Y METODO SEGUIDO

Antes de planificar el trabajo necesario para cumplir los objetivos definidos y el alcance de los mismos, tendremos que decidir el enfoque general y la metodología a seguir. Esta es la finalidad de este apartado.

El enfoque se establece desde un punto de vista general tomando como base el marcado carácter académico del trabajo. Lo que limita los recursos y el tiempo necesario para completar el trabajo.

Al igual que el enfoque, la metodología a seguir también se ve afectada por la misma característica intrínseca. Por lo que no hay equipo, no hay organización de tareas, pero si se dispone de una planificación temporal de tareas adaptada a estas características.

1.4. PLANIFICACION DEL TRABAJO

Este apartado proporciona una estructura temporal organizada del flujo de trabajo que nos permitirá hacer un seguimiento de los objetivos y el alcance previstos.

1.5. BREVE SUMARIO DE PRODUCTOS OBTENIDOS

Aquí se enumeran brevemente los productos obtenidos como resultado de la ejecución de tareas previstas en la planificación.

1.6. ORGANIZACIÓN DE CONTENIDOS

Se trata de este mismo apartado, en el que describimos de manera resumida como se han organizado los contenidos. Se enumeran y describen brevemente los apartados de esta memoria.

2. ANTECEDENTES

Todo lo que se ha tratado en apartados anteriores es fundamental para introducirnos en el contexto de la información, medir el esfuerzo y prever riesgos. Este apartado supone un paso fundamental antes de buscar soluciones al problema propuesto. Se encarga de estudiar y evaluar los antecedentes al presente trabajo, los resultados del mismo nos acercarán a la mejor solución y nos evitará errores decisión.

3. ANALISIS DE SOLUCIONES

Este apartado presta atención a la búsqueda, estudio y evaluación de soluciones al problema propuesto. Para ello se divide el contexto del problema en 4 partes: el servidor de mapas, el servidor de datos, el servidor web y el visor web.

3.1. SERVIDOR DE MAPAS

Se comparan dos alternativas para el servidor de mapas y se justifica la selección de una de ellas. Después se describe con mayor detalle la arquitectura y funcionamiento del servidor de mapas elegido.

3.2. SERVIDOR DE DATOS

Este apartado incluye información detallada sobre el formato de los datos alfanuméricos que se representarán en la cartografía. Debido al formato origen de los datos se decide y justifica la utilización de una base de datos relacional. Además se define el formato de datos para la comunicación entre los scripts de servidor y los datos.

3.3. SERVIDOR WEB

De manera similar al apartado 3.1, se comparan distintos servidores web, se selecciona uno de ellos y se justifica la elección.

3.4. VISOR WEB

En este apartado se enumeran y describen las principales tecnologías que se utilizarán para la implementación del visor web. Para la elección de las mismas se tienen en cuenta las decisiones tomadas respecto del servidor de mapas, el servidor web y el servidor de datos.

4. DISEÑO

Ha llegado el momento de ir ¡al ataque...!, este apartado trata sobre el desarrollo de 3 de las 4 partes del apartado anterior. No se incluye la parte correspondiente al servidor web porque su implementación no resulta relevante para los objetivos de este trabajo.

4.1. DATOS

Este apartado trata sobre los dos tipos de datos que enmarcan este trabajo: alfanuméricos y geoespaciales. Los primeros se tratan desde el punto de vista estructural en relación con el servidor de base de datos elegido. En el caso de los datos geoespaciales se describe el proceso de implementación de PostGIS, se describen y enumeran los datos geoespaciales relativos a las regiones sanitarias, lo mismo para los datos asociados con los centros de atención hospitalaria, y se introduce brevemente la estructura básica de los ficheros de mapas.

4.2. VISOR WEB

Se describen las principales características de la interfaz, las funciones más importantes y los ficheros de scripts utilizados.

También se describen los scripts de servidor que se comunican con los scripts de cliente mediante objetos de datos serializados.

4.3. SERVIDOR DE MAPAS

En este apartado se describe paso a paso la instalación del software servidor de mapas, incluyendo componentes complementarios para la importación de datos geográficos (caché de mapas, herramientas de importación de datos, etc.).

Además se describen las principales características de los ficheros de mapas utilizados.

5. PRUEBAS Y RESULTADOS FINALES

Este apartado define y describe las pruebas realizadas, y los resultados obtenidos.

6. CONCLUSIONES Y MEJORAS

Una vez se han realizado las pruebas finales y revisada toda la información que se ha generado resulta más fácil llegar a conclusiones y proponer mejoras futuras. Este es el cometido de este apartado.

7. GLOSARIO

Se enumeran y describen brevemente los principales términos de interés en el contexto de este trabajo.

8. BIBLIOGRAFIA

Lista de libros, artículos, blogs, etc. que han sido consultados durante la realización del trabajo.

9. ANEXOS

Este apartado incluye dos anexos, uno dedicado a la planificación inicial del plan de trabajo, que posteriormente ha servido como guía para elaborar la planificación temporal detallada del trabajo completo. El segundo incluye una guía rápida de usuario.

2. ANTECEDENTES

Desde una perspectiva general el 'Atlas Electrònic de Salut de Catalunya'² supone un precedente a este proyecto, destacando como diferencia sobre los objetivos de este trabajo, el acceso restringido a profesionales de la salud. Se trata de un visor de mapas con los principales indicadores de salud que ayudan a los usuarios a la toma de decisiones y la planificación. Las principales componentes de este sistema son:

- Visor geográfico basado en navegador de Internet.
- *Agregador de indicadores generales*: es un módulo que alimenta un repositorio de indicadores del servidor con las bases de datos fuente. Agrega registro y elabora estadísticas.
- *Publicador de contenidos*: es un módulo que permite que los usuarios publiquen directamente en el Atlas sus indicadores:
 - o Especializados: creados con SIG de escritorio.
 - o Personales: provenientes de una tabla Excel.
- Visor PDA de acceso directo a los usuarios, una vez autenticados, con funciones adecuadas a su movilidad.

En lo relativo a análisis estadísticos, el proyecto 'New Zealand Health Quality & Safety Commission Atlas of Healthcare variation'³ representa un antecedente para este trabajo. Uno de los resultados de este proyecto ha sido el desarrollo de un atlas de salud en el que se muestran mapas, gráficos y tablas que destacan las variaciones, por área geográfica, la prestación y utilización de servicios sanitarios específicos. Este sistema impulsa el debate entre profesionales y usuarios para poner de relieve las diferencias en la percepción del sistema sanitario y estimular, de este modo, la mejora del mismo.

El estudio de las zonas de influencia de cada centro sanitario se realizará utilizando diagramas de Voronoi. El proyecto 'Diagramas de Voronoi de alcance limitado'⁴ realizado en la Universidad Politécnica de Cataluña se utilizará como base para dicho estudio. El objetivo principal del citado proyecto fue el desarrollo de una aplicación de visualización para el estudio de diagramas de Voronoi de alcance limitado y los caminos de desviación mínima.

² <http://observatorisalut.gencat.cat/es/demanardades/Dades-sanitaries/dadesobertessalut/index.html>

³ <http://www.hqsc.govt.nz/our-programmes/health-quality-evaluation/projects/atlas-of-healthcare-variation/>

⁴ <http://www-ma2.upc.edu/~geoc/DVALon/MemoriaDVALon.pdf>

3. ANALISIS DE SOLUCIONES

En este apartado se estudiarán y evaluarán los tipos de software que se utilizarán para afrontar la solución. Se seleccionarán aquellos que se distribuyan bajo licencia no comercial, del mismo modo que se distribuirá el atlas geosanitario resultante de este trabajo. En el apartado 6 se describirán con detalle todos los recursos necesarios para dar soporte a la solución propuesta.

Teniendo en cuenta los objetivos propuestos podemos pensar en una solución adecuada compuesta por los siguientes tipos de software (figura 7): servidor de mapas, servidor de datos, servidor web y visor web.

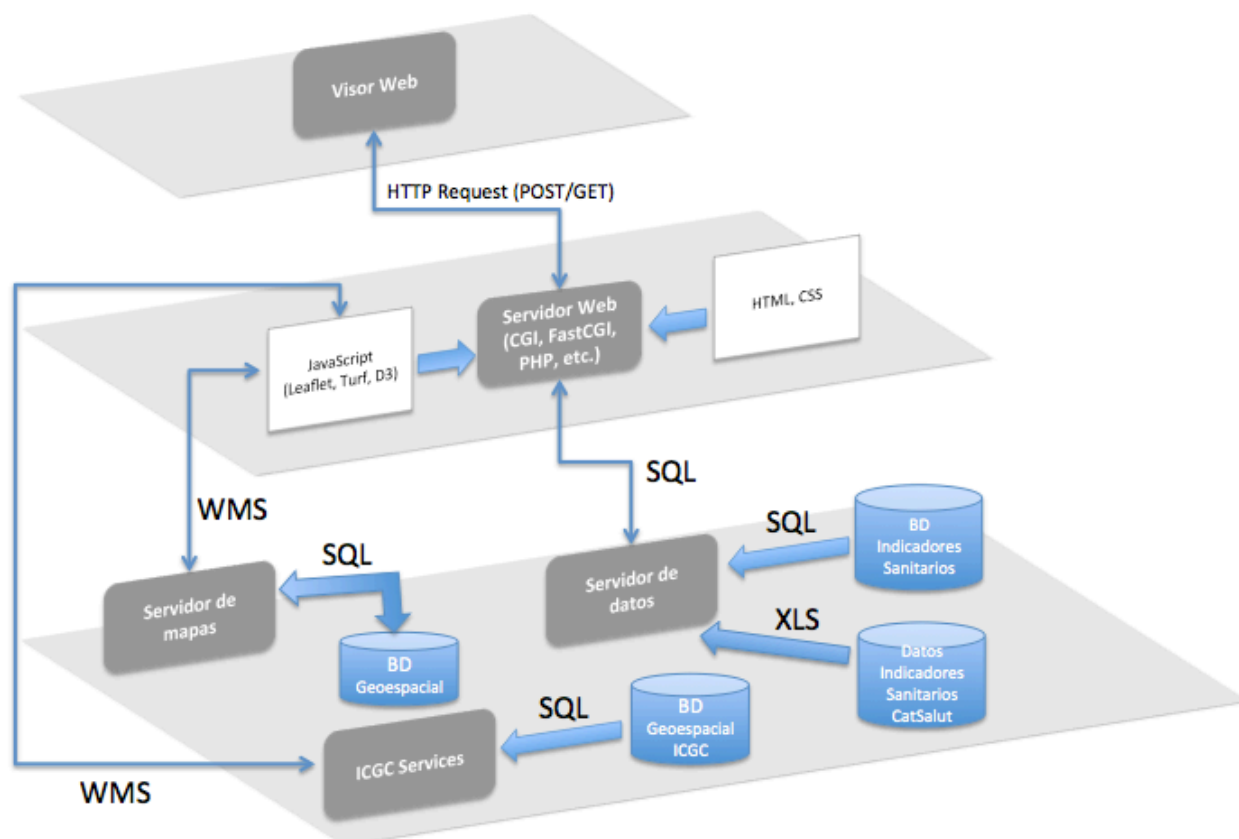


Figura 7: esquema general de los componentes de la solución

3.1. SERVIDOR DE MAPAS

Los dos servidores de mapas, no comerciales, más conocidos son MapServer y GeoServer.

MapServer es uno de los servidores de mapas más conocidos. Es un desarrollo Open Source iniciado por la Universidad de Minnesota y su Departamento de Recursos Naturales en colaboración con la NASA, con el objetivo de servir mapas en Internet, aunque actualmente su principal promotor es Autodesk. Está implementado como una aplicación CGI (aunque dispone de un adaptador Java para Apache), y utiliza numerosas librerías freeware o de código abierto, como Shapelib, FreeType, Proj.4, libTIFF o Perl. Las principales características son:

- Multiplataforma
- Alta velocidad de acceso a datos

- Soporta multitud de formatos raster y vectorial.
- Comunidad muy activa y desarrollada. Un dato de interés, a este respecto, son las estadísticas sobre usuarios de MapServer⁵, según las cuales el número de usuarios adscritos a la lista alcanza los 2344 usuarios, representando a 67 países.
- Tiene un importante número de productos que lo soportan, clientes web (ka-Map, Chamaleon, CartoWeb), utilidades para tratamiento de imágenes, etc.

Por otro lado, GeoServer es un servidor de mapas que forma parte de la nueva generación de aplicaciones desarrolladas sobre la especificación J2EE. Está implementado sobre la plataforma Java, permitiendo el despliegue de la aplicación sobre cualquier servidor de aplicaciones conforme a la especificación J2EE, tanto libres (como Tomcat o Geronimo de Apache o JBoss de RedHat) como propietarios (WebLogic de BEA o WebSphere de IBM, entre otros). Las principales características son:

- Facilidad de manejo, la configuración se realiza a través de una aplicación Web que evita la edición de complejos ficheros de configuración.
- Basado en tecnología J2EE, puede ejecutarse en cualquier contenedor de Servlet.
- Dispone de una potente comunidad, con listas e-mail muy activas que dan soporte al proyecto. Se trata de una plataforma muy dinámica que crece con rapidez.
- Actualmente existen multitud de plataformas que utilizan Geoserver para la publicación de mapas, principalmente mediante el protocolo WFS. Entre ellos el cliente embebido MapBuilder basado en AJAX.
- Permite enviar datos puramente vectoriales utilizando servicios WFS-T (Web Feature Service-Transactional).

Realizaremos una comparativa lo más objetiva posible que nos ayude a analizar las características de cada servidor y decidir que servidor de mapas vamos a implantar.

- MapServer es generalmente mejor con WMS, mientras GeoServer trabaja mejor con WFS.
- Otra gran diferencia es que GeoServer soporta WFS-T, mientras MapServer no.
- MapServer es un proyecto más maduro (1996), mientras que GeoServer es un poco más reciente (2003).
- GeoServer proporciona una web para administración, lo que facilita la configuración. MapServer utiliza ficheros de configuración (Mapfiles).

⁵ Publicadas por el OSGeo Journal en el volumen de mayo del 2007

- MapServer trabaja con CGI, GeoServer con J2EE. Esto puede traducirse como una ventaja para GeoServer, ya que algunas compañías se muestran reacias a trabajar con CGI.
- MapServer dispone de un poderoso sistema cartográfico, proporcionando datos bajo vectores dinámicos con alta calidad.
- Para MapServer, al contar con librerías nativas, el coste de procesado de las imágenes será menor, pero hay que tener en cuenta el acceso a la información y el pool de conexiones.
- Si la información se almacena en base de datos y no se utiliza memoria caché, GeoServer accedería a la información de manera más rápida ya que dispone de pool de conexiones.
- Si se utiliza caché, MapServer sería más rápido, ya que se limita el acceso a base de datos, que es donde MapServer consume más tiempo.

La comparativa nos deja una diferencia destacable: MapServer parece más conveniente para tratar información de tipo raster⁶, mientras que GeoServer tiene mejor rendimiento en datos vectoriales. Una solución híbrida sería muy recomendable en términos de rendimiento. Sin embargo, para la realización de este trabajo no disponemos de holguras de tiempo suficiente que nos permitan optar por la solución híbrida.

Se seleccionará MapServer por tres motivos:

- No necesitamos una gran precisión en los mapas, trabajaremos principalmente con información de tipo ráster.
- Soporta multitud de formatos tanto ráster como vectorial, manteniendo buena velocidad de acceso.
- La comunidad es muy amplia y se dispone de muchas fuentes de información, que pueden ser de utilidad sobre todo cuando uno de los objetivos es conocer en profundidad su funcionamiento.

El servidor de mapas, MapServer, soporta tres tipos de formatos de datos de entrada para la información vectorial:

- *Basado en ficheros:* ESRI Files (SHP, DBF, XHX), MicroStation Design Files (DGN), GeoTIFF images, etc.
- *Basado en directorio:* ESRI ArcInfo Coverages, US Census TIGER, etc.
- *Conexiones a bases de datos geoespaciales:* PostGIS, ESRI ArcSDE, MySQL, etc.

⁶ Es una estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color, denominada matriz, que se puede visualizar en un monitor, papel u otro dispositivo de representación

MapServer trabaja con distintos formatos de ficheros con información de tipo raster. Algunos de los formatos soportados son: JPG, PNG, GIF, TIFF/GeoTIFF, EPPL7 y otros formatos utilizando GDAL⁷ (Geospatial Data Abstraction Library).

La cartografía base, topográfica y ortofotográfica de Cataluña, se obtendrá de Institut Cartogràfic i Geològic de Catalunya (ICGC) mediante conexiones WMS, y de OSM (Open Street Maps) obteniendo los ficheros de datos correspondientes y almacenando los datos geoespaciales en PostGIS. El visor de mapas obtendrá la cartografía base desde este servicio, y desde el servidor de mapas propios de la solución. Se utilizarán conexiones WMS y todos los componentes software especializados en el tratamiento de mapas que se han comentado en el apartado.

El visor web podría obtener los mapas de una única fuente de información, servicios ICGC o bien del servidor de mapas incluido en la solución. La primera fuente de información hubiese sido la menos complicada, sin embargo es objetivo de este trabajo conocer el funcionamiento de alguno de los servidores de mapas más utilizados. Por lo que se utilizarán las dos fuentes de información geoespacial, dando la posibilidad al usuario de seleccionar la fuente de información que quiere utilizar en cada consulta.

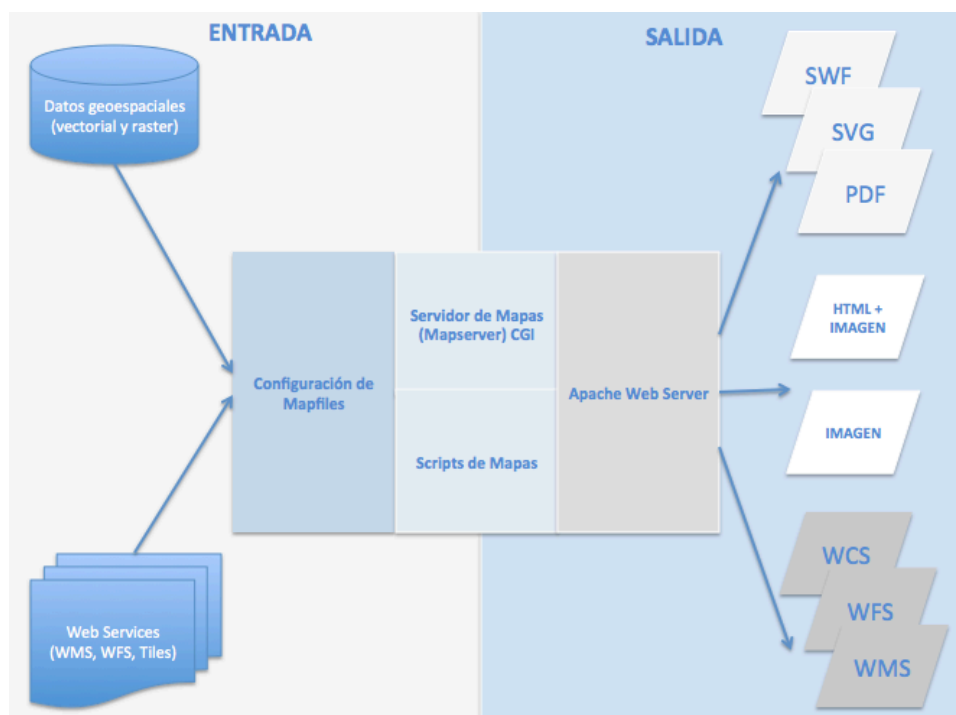


Figura 8: arquitectura básica de una aplicación en MapServer

Si nos introducimos un poco más en el funcionamiento de MapServer, podemos ver que arquitectura básica de aplicación (figura 8) es la siguiente:

- **Datos geoespaciales:** MapServer soporta diferentes fuentes y formatos de datos, el formato por defecto es ESRI Shape. Además puede incorporarse las librerías GDAL para compilar otros tipos de datos y ampliar el rango de formatos admitidos. Como por ejemplo los datos almacenados en

⁷ Puede ampliar información sobre esta librería en <http://es.wikipedia.org/wiki/GDAL>

PostGIS mediante la importación de ficheros obtenidos desde OSM (spain-latest.osm.bz2).

- *Web Services (WMS, WFS, Tiles)*: una fuente de datos alternativa a los que se dispone en la base de datos geoespacial (por ejemplo la que proporciona ICGC). Son servicios dinámicos, definidos por OGC, que proporcionan mapas de datos referenciados espacialmente. Los mapas generados se suelen presentar en formato de imagen digital (PNG, GIF o JPEG) y también como gráficos vectoriales (SVG, PDF o SWF).
- *MapFile*: se trata de un fichero de configuración estructurado. Define el área del mapa, los datos de entrada del mapa y el formato de la salida. Además se puede utilizar para definir capas (incluyendo orígenes de datos específicos), proyecciones y simbología. El fichero debe tener la extensión .map para que MapServer pueda reconocerlo.
- *Scripts de mapas*: proporcionan una interfaz para aplicaciones Web, es un módulo de MapServer puede ser invocado de manera independiente al módulo CGI. Soporta diferentes tipos de lenguajes de script: PHP, Perl, Python, Java, .NET flavors, etc.
- *Servidor de mapas CGI*: no hay memoria de estado, cada petición es nueva y no se almacena información relativa a la petición anterior. Cada petición envía información de contexto almacenada en variables ocultas de formularios Web o a través de variables incluidas en la propia URL (capas que se quieren visualizar, localización de un punto concreto en el mapa, símbolos, etc.). En una aplicación web simple se utilizarían dos páginas HTML, una a modo de inicialización donde se envía la información de contexto de la petición. Y otra a modo de plantilla con la respuesta de MapServer, que será la que se visualizará en el navegador web a modo de respuesta. Esta última, incluye información de referencia de los objetos que incluye (nombre de la imagen del mapa, referencia de la imagen del mapa, dimensión del mapa, etc.) y además información que determina las posibilidades de interacción del usuario con el visor.
- *Servidor Web (Apache)*: se encarga de servir las páginas HTML que se utilizarán la visualización de mapas. Se comunica con el módulo CGI del servidor de mapas para enviar las peticiones y recibir las respuestas correspondientes, que posteriormente enviará al navegador del usuario que realizó la petición.
- *Información de respuesta*: mapas en formato gráfico y vectorial, HTML con el mapa en formato imagen, o bien la publicación de un servicio de información geográfica que puede ser utilizado del mismo modo que hemos indicado cuando se han tratado los Web Services. Por lo que, podemos utilizar como fuente de entrada y salida un Web Services como por ejemplo WMS.

3.2. SERVIDOR DE DATOS

Necesitamos combinar datos geoespaciales (información vectorial y de tipo raster) con datos alfanuméricos procedentes de los indicadores sanitarios. El

visor web debe responder consultas del usuario que relacionen por ejemplo las zonas sanitarias con el gasto sanitario, o las zonas sanitarias y la probabilidad de la población, que vive en la misma, de padecer una epidemia. También otras consultas en las que sólo se utilizarán datos geoespaciales, como la que nos permite conocer los centros sanitarios de acceso más rápido a una localidad determinada.

Los datos alfanuméricos se distribuyen en formato XLS y se organizan en diferentes ficheros (uno por año) y dimensiones de análisis: edad, evolución, territorio, etc. El formato de los ficheros de datos incluye características de diseño (color, comentarios, líneas, etc.) y los datos aparecen, en algunos casos, en distintas hojas. Por lo que automatizar la conversión de dicho formato a otro, como por ejemplo tablas de una base de datos relacional, no se considera oportuno. Se anotará como mejora el desarrollo de un proceso que automatice la conversión de datos en la medida que el formato que se distribuyen actualmente cambie y lo haga posible.

Se utilizará un proceso manual de carga de los datos en una base de datos relacional como PostgreSQL, a la que se ha denominado 'openatlasdb'. Después, los datos podrán ser consultados utilizando el lenguaje de consultas SQL desde scripts PHP encapsulándolos en objetos JSON. Finalmente los objetos JSON serán utilizados por JavaScript para obtener la información correspondiente al tipo de consulta que haya solicitado el usuario.

El modelo de la base de datos para los datos relativos a los indicadores sanitarios es relacional. No existen relaciones entre las tablas y tampoco ha sido necesario definir componentes de programación como funciones o procedimientos almacenados. Los principales componentes del modelo son:

- *Tablas:*
 - *ahospitalaria:* almacena las localizaciones geográficas de los centros de atención hospitalaria.
 - *cobertura:* almacena los datos relativos a los distintos tipos de hospitalizaciones por centro, temporada y región sanitaria.
 - *gasto:* almacena los datos relativos a los distintos tipos de gasto por cada región sanitaria, temporada, sexo y tramo de edad (sólo hay disponibles datos del 2013).
 - *epidemia:* almacena los datos relacionados con los distintos tipos de enfermedades contagiosas por región y temporada.
 - *poblacion:* almacena la población de cada municipio de Cataluña por año (de 2008 a 2014).

- *Vistas:*
 - *vwCobertura:* esta consulta relaciona las tablas 'ahospitalaria' y 'cobertura' por el municipio. Como resultado se obtiene una tabla con la información de cobertura, la localización geográfica de los municipios y los datos relativos a las hospitalizaciones.

- *vwEpidemia*: muestra los datos agrupados por región, temporada y tipo de enfermedad el número de casos registrados.
- *vwGasto*: muestra los datos agrupados por región, indicador de gastos y temporada.

Se ha seleccionado la base de datos PostgreSQL porque es la recomendada por MapServer. Además esta base de datos se adapta correctamente tanto a modelos de datos relacionales como geoespaciales.

3.3. SERVIDOR WEB

Los ficheros de script PHP se alojan y ejecutan en un servidor web. Además en el servidor web se alojarán el resto de ficheros que se utilizarán desde el visor web (HTML, CSS, JavaScript, etc.).

La aplicación web estará ubicada dentro de un contenedor o sitio web configurado para permitir el acceso de los usuarios a través de una dirección de Internet (URL). La interacción del usuario con el visor web generará peticiones al servidor (GET/POST) que serán resueltas en el mismo, y los resultados serán devueltos en formato HTML. No todas las acciones de interacción del usuario representarán peticiones al servidor, otras acciones se resolverán en el lado del navegador desde el que se invoque el visor web. En general las acciones que se resuelvan en el lado del navegador serán más rápidas y se implementarán con JavaScript. Las acciones en el lado del servidor implicarán un mayor tiempo de respuesta, puede que se necesite hacer cálculos, cargar datos geoespaciales, datos alfanuméricos o ambos. Estas acciones se implementarán utilizando PHP.

Actualmente existen muchos servidores web, no comerciales, cuyas características serían suficientes para el objeto de este trabajo (FastCGI, SCGI, PHP, CGI, SSI, SSL/TLS, etc.). Algunos de ellos son:

- *Apache*: es un servidor web de código abierto multiplataforma (Unix, GNU/Linux, Windows y Mac OS X). Es uno de los servidores web más veteranos del mercado, permite el uso de distintos módulos para ampliar su funcionalidad. A partir de la versión 2.0 se distribuye bajo licencia ASL (Apache Software License), compatible con GPLv3.
- *NGINX*: es un servidor web y proxy inverso para protocolos de correo. Se distribuye bajo licencia BSD simplificada y es multiplataforma (GNU/Linux, Solaris, Mac OS X, Windows, etc.). Se utiliza en muchos sitios web, como: WordPress, Hulu, GitHub y SourceForge.
- *Lighttpd*: es un servidor web rápido, seguro y flexible. Es recomendable en entornos donde la velocidad es una cuestión importante. Tiene un bajo consumo de CPU y memoria RAM. Se distribuye bajo licencia BSD y se puede utilizar en GNU/Linux, UNIX y Windows.
- *Cherokee*: es un servidor web rápido escrito completamente en C. Soporta multitud de complementos que permiten ampliar su funcionalidad. Se distribuye bajo licencia LGPL de GNU.

El servidor web Apache es uno de los más utilizados actualmente, su madurez en el mercado ha facilitado la aparición de multitud de fuentes de información que serán de utilidad para el desarrollo de este trabajo.

Finalmente, en la tabla 1, se muestra una comparativa de las principales características de los servidores web analizados.

	Característica	Apache	NGINX	Lighttpd	Cherokee
Seguridad	Autenticación básica de acceso				
	Autenticación avanzada de acceso				
	SSL/TLS				
Contenido dinámico	CGI (*)				
	FCGI				
	SCGI				
	Java Servlets (*)				
	ISAPI				
Otras características	Consola de administración (*)				
	IPV6				

Tabla 1: comparativa de características básicas de los servidores web analizados

Las características marcadas con * son a priori las más significativas para la elección del servidor web. El módulo CGI es necesario para la compatibilidad con el servidor de mapas, el soporte para servlets es muy conveniente para desarrollar servicios de datos y la consola de administración es recomendable, ya que facilita el proceso de desarrollo en especial durante la fase de depuración.

3.4. VISOR WEB

El visor web es una aplicación para navegadores de Internet. Actualmente existen multitud de frameworks, plugins y librerías JavaScript que facilitan el tratamiento de mapas. Por otro lado la serialización de datos en clases de objetos (JSON, GeoJSON, etc.) ha reducido considerablemente el esfuerzo de programación para convertir datos, almacenados en algún tipo de fichero o base de datos, en objetos en memoria. El visor web es una aplicación cliente que realiza peticiones de geoinformación y de datos para combinarlas en un mapa. El lenguaje de marcas HTML5 y las hojas de estilo CSS3 nos servirán para ofrecer al usuario una interfaz sencilla y usable.

A continuación se analizan algunas librerías y plugins orientados a mapas, seleccionando los que se consideren más adecuados para este trabajo.

- *LeafLet*: es una librería JavaScript, de código abierto, para crear mapas interactivos en un entorno móvil. Se utiliza en muchos sitios web conocidos como: Flickr, Wikimedia, Foursquare, OSM, cartoDB, GIS Cloud, Washington Post, etc. Una alternativa a esta librería es OpenLayers que era la API principal de OpenStreetMap antes de aparecer Leaflet. Analicemos algunas de las principales diferencias entre OpenLayers 3 y LeafLet para decidir cuál de las dos utilizaremos en este trabajo.

Las principales características de Leaflet son:

- o Rapidez de aprendizaje y facilidad de uso
- o Estilos más modernos

- o Es una librería muy ligera
- o Compatible con dispositivos móviles
- o Buen rendimiento
- o Utiliza plugins adicionales para extender su funcionalidad lo que permite una programación modular según las necesidades.

Las principales características de OpenLayers 3 son:

- o Soporta todos los protocolos WebGIS
- o Mucha documentación y existencia de una comunidad amplia
- o Se pueden utilizar herramientas de edición
- o El proyecto es más maduro y está más probado
- o Tiene más controles de mapas (edición, medición, etc.)
- o Soporte WMTS
- o En general no es necesario incorporar plugins adicionales.
- o El tamaño de la librería sigue siendo grande sobre todo para dispositivos móviles.

Se utilizará Leaflet principalmente porque es una librería muy ligera, fácil de usar, flexible y eficiente. Nos será muy útil pensando en futuras mejoras o ampliaciones.

- *Turf*: es una librería JavaScript que permite realizar operaciones geoespaciales desde el navegador. Nos permite realizar la mayoría de las operaciones clásicas en un GIS: buffering, within, merge, unión, contouring, TINs, etc. Además permite usar GeoJSON directamente y se integra perfectamente con LeafletJS.

Con esta librería podemos calcular las rutas más próximas desde una localidad al centro sanitario más próximo. Para ello crearemos un área de influencia (buffering) de una distancia mínima determinada a la ruta y luego obtendremos la localización de los centros sanitarios próximos dentro de dicha área (within).

- *D3*: Es una librería JavaScript que permite enlazar y transformar datos convirtiéndolos en objetos DOM (Document Object Model). Se pueden obtener tablas HTML a partir de un vector de datos, o bien utilizar dichos datos para crear un gráfico SVG interactivo. También se integra correctamente con Leaflet, por ejemplo se puede utilizar para conseguir visualizar mapas canvas en formato vectorial (SVG), manteniendo la nitidez en diferentes pantallas.

Esta librería nos permite además crear animaciones visuales, se puede aprovechar esta función para incrementar la interactividad con el mapa. Por ejemplo, cuando el usuario realice una consulta sobre el gasto sanitario superior a una cantidad dada, se podría crear una animación visual intermitente para los polígonos que representan las regiones

sanitarias que queden por encima de la cantidad indicada. También se podría utilizar para crear una animación visual con las líneas que representan la ruta más rápida a un centro sanitario desde una localidad determinada.

4. DISEÑO

Ya en los apartados anteriores se ha dejado claro que en este trabajo hay tres partes diferenciadas: datos, visor web y servidor de mapas. En este apartado se tratarán todas las especificaciones de diseño.

Para la implantación del servidor de mapas y el servidor de datos alfanuméricos se ha creado una máquina virtual con el software de virtualización VirtualBox, que se distribuye bajo licencia GNU. Se ha instalado Ubuntu Server 14.04 sobre la máquina virtual, asignándole los recursos mínimos para asegurar el funcionamiento del servidor de mapas sin perjudicar el rendimiento de la máquina física. El nombre de la máquina virtual es *ubuntu-server*.

El software de virtualización, VirtualBox, se instala con ayuda de un asistente sobre el sistema operativo de la máquina anfitrión con OSX Yosemite versión 10.10.2.

4.1. DATOS

Los datos que se utilizarán serán del tipo alfanumérico y de tipo geoespacial. Los primeros provienen de los indicadores sanitarios del CatSalut, utilizados para estimar la influencia de los centros sanitarios y cobertura asistencial, el gasto sanitario y el riesgo de epidemias. Algunos de estos datos, como los nombres de los centros sanitarios serán utilizados para la geolocalización de los mismos. Estos datos se distribuyen de forma abierta y se pueden descargar desde el Observatorio del Sistema de Salud de Cataluña. Se pueden obtener en distintos formatos, principalmente en formato texto y en formato CSV⁸. Se ha seleccionado el último formato porque facilita la importación de los mismos en una base de datos.

Datos alfanuméricos

Una vez se han descargado los datos en el formato indicado se ha realizado el proceso de importación. Para ello se ha creado una base de datos dedicada (sanidad) en el servidor de PostgreSQL, el gestor de la base de datos dispone de proceso de importación y es compatible con el formato CSV. La creación de la base de datos se ha realizado utilizando la consola y los comandos correspondientes:

```
sudo -u postgres createdb --encoding=UTF8 sanidad
```

Los conjuntos de datos representativos de cada contexto de información se han almacenado en tablas distintas. Se han creado tres tablas distintas: cobertura, epidemias y gasto. La estructura de dichas tablas es la siguiente:

- Cobertura
 - *idrow* (bigserial): campo clave autoincremental. El contexto de información para esta tabla no permite definir otro tipo de clave primaria de la tabla, los datos no se pueden identificar unívocamente.

⁸ Comma Separator Values

- *region* (text): región sanitaria.
- *sector* (text): sector sanitario.
- *año* (integer): año del estudio.
- *indicador* (text): denominación del indicador.
- *valor* (double): valor del indicador.
- *total* (double): valor total (todas las regiones sanitarias) para el indicador correspondiente.
- *Epidemias*
 - *temporada* (text): temporada del estudio (bianual)
 - *semana* (integer): semana de la temporada.
 - *region* (text): región sanitaria.
 - *valor* (integer): número de casos de la enfermedad infecciosa.
 - *origen* (text): denominación de la enfermedad infecciosa.

La clave principal de esta tabla lo componen los siguientes campos: temporada, semana, región y origen.

- *Gasto*
 - *idrow* (bigserial): campo clave autoincremental. El contexto de información para esta tabla no permite definir otro tipo de clave primaria de la tabla, los datos no se pueden identificar unívocamente.
 - *region* (text): región sanitaria.
 - *sector* (text): sector sanitario.
 - *año* (integer): año del estudio.
 - *indicador* (text): denominación del indicador.
 - *valor* (double): valor del indicador.
 - *total* (double): valor total del indicador en toda Cataluña.

En las tablas no se han definido relaciones, restricciones, vistas, etc. No es necesario para el tratamiento de datos previsto, sólo se necesitan índices bien definidos para acelerar la carga de datos desde el visor web.

Llegados a este punto ya disponemos de los datos alfanuméricos en un formato adecuado para consultas desde un lenguaje de script como PHP. Para llegar hasta aquí ha sido necesario:

- Descargar los datos alfanuméricos relativos a los indicadores sanitarios de interés.

- Seleccionar conjuntos de datos (ficheros en formato CSV):
 - *Cobertura*: hospitalizaciones de los distintos centros sanitarios.
 - *Epidemias*: casos identificados de distintas enfermedades contagiosas.
 - *Gasto*: importe de gasto farmacéutico por región y sector sanitario, importe de facturación al CatSalut.
- Crear la base de datos en PostgreSQL y las tablas correspondientes, uno por cada conjunto de datos.
- Importar datos procedentes de los ficheros del conjunto de datos en las tablas correspondientes.

No se han encontrado datos de los centros de atención primaria que pudieran ser de utilidad para estimar la cobertura asistencial. Es por este motivo por el que en el modelo de datos no aparece ninguna tabla para almacenar dicha información. La cobertura asistencial se estimará a partir de los distintos tipos de hospitalizaciones en los centros de atención hospitalaria.

Datos geoespaciales

Los datos geoespaciales, para el servidor de mapas local, provienen de OSM. Por motivos de rendimiento y contexto del tratamiento de la información se utilizarán sólo los datos geoespaciales relativos a España⁹. Esta fuente de datos distribuye los mismos en diferentes formatos, se ha seleccionado un formato compatible para importación en una base de datos geoespaciales tipo PostGIS. La base de datos se ha creado en PostgreSQL, formato PostGIS, y después se ha realizado el proceso de importación desde el fichero con datos geoespaciales descargado (spain-latest.osm.bz2) a la base de datos.

Antes de importar los datos geoespaciales desde el fichero OSM se crea la base de datos geoespacial. Para ello se instala el complemento de bases de datos geoespaciales PostGIS (v.2.1), el proceso se ha llevado a cabo utilizando líneas de comando en la consola del sistema:

- Se crea la base de datos geoespacial denominada 'osm':

```
sudo -u postgres createdb --encoding=UTF8 osm
```

- Se ejecutan los scripts de PostGIS para ajustar el formato de la base de datos creada a un formato geoespacial.

```
sudo -u postgres psql -d osm -f
/usr/share/postgresql/9.4/contrib/postgis-2.1/postgis.sql

sudo -u postgres psql -d osm -f
/usr/share/postgresql/9.4/contrib/postgis-2.1/spatial_ref_sys.sql
```

⁹ <http://download.geofabrik.de>

```
sudo -u postgres psql -d osm -f
/usr/share/postgresql/9.4/contrib/postgis-
2.1/postgis_comments.sql

sudo -u postgres psql -d osm -c "GRANT SELECT ON spatial_ref_sys
TO PUBLIC;"

sudo -u postgres psql -d osm -c "GRANT ALL ON geometry_columns TO
rafa;"
```

- Se importan los datos en formato .osm a la base de datos 'osm'
 - Se accede al directorio donde se encuentra el fichero .osm y se leen los datos geoespaciales del fichero. El sistema de referencia espacial es el EPSG:3857, uno de los más utilizados en aplicaciones web con mapas (Google, Bing, OpenStreetMaps, etc.), también conocido como EPSG:900913.

```
cd /home/rafa/gis

imposm --proj=EPSG:3857 --read spain-latest.osm.bz2
```

- Se escriben los datos leídos en la base de datos y se optimiza

```
imposm --proj=EPSG:3857 --write --database osm --host localhost -
-user postgres

imposm --optimize -d osm
```

Para poder invocar al servicio de PostgreSQL desde la máquina anfitrión es necesario modificar el fichero de configuración del servicio y reiniciarlo. Esto se hace modificando el fichero 'pg_hba.conf':

```
sudo gedit /etc/postgresql/9.4/main/pg_hba.conf
```

Se habilita un total de 30 direcciones IP locales para el acceso desde la máquina anfitrión al servidor PostgreSQL.

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only local all
#all trust # IPv4/v6
#connections:

host all all 192.168.1.90/120 trust

host all all ::1/128 trust
```

Las regiones y sectores sanitarios son delimitaciones geográficas que pueden ser representadas mediante polígonos. La delimitación de las mismas se ha

obtenido desde mapas publicados en InstaMaps¹⁰, sin embargo las capas donde se representa dicha información no se pueden exportar. Para llevar a cabo esta delimitación y otros tratamientos de datos geoespaciales se ha utilizado la herramienta QGIS¹¹ que se distribuye bajo licencia GNU. Utilizando las conexiones a los servidores de datos geoespaciales, servidor local y servidor del ICGC, se han obtenido las cartografías base de tipo ortográfica y topográfica. Sobre la cartografía base, y utilizando la información geoespacial de los límites administrativos de Cataluña¹², se han realizado las delimitaciones de las regiones sanitarias. Las capas resultantes se han exportado en formato GeoJSON para utilizarlas posteriormente desde el visor web. La delimitación de los sectores sanitarios se ha realizado construyendo polígonos que unen las localizaciones geográficas de los centros que componen cada sector. Las capas resultantes también se han exportado a formato GeoJSON. Finalmente, los datos geoespaciales de los centros sanitarios, atención primaria y hospitalaria, se han obtenido desde recursos propios de InstaMaps. Esta herramienta web permite exportar capas de datos geoespaciales en diferentes formatos: GeoJSON, ESRI, DXF, KML y GPX. Se ha utilizado de nuevo el formato GeoJSON.

Se ha obtenido como resultado los ficheros con información geoespacial en formato GeoJSON. Uno por cada región y sector sanitario, uno para los centros de atención hospitalaria y otro para los de atención primaria. Los ficheros GeoJSON son los siguientes:

- *altopir.geojson*: región sanitaria del Alto Pirineo.
- *barcelona.geojson*: región sanitaria de Barcelona.
- *catacentral.geojson*: región sanitaria de Cataluña Central.
- *gerona.geojson*: región sanitaria de Gerona.
- *lleida.geojson*: región sanitaria de Lleida.
- *tarragona.geojson*: región sanitaria de Tarragona.
- *terresebre.geojson*: región sanitaria de Terres del Ebre.

Los ficheros correspondientes a las regiones sanitarias tienen dos características (features):

- *Propiedades (properties)*:
 - *id*: identificador de la región sanitaria.
 - *region*: nombre de la región sanitaria.
- *Geometría (geometry)*: de tipo polígono, contiene las localizaciones geográficas de los puntos que definen el polígono.

¹⁰ <http://www.instamaps.cat/>

¹¹ QGIS versión 2.8.1-wien, disponible en <http://qgis.org>

¹² Geoservicios del ICGC, <http://www.icc.cat/esl/Home-ICC/Geoinformacio-digital/Serveis-en-linia-Geoserveis/WMS-de-cartografia-vector/WMS-Base-municipal-1-50.000-limits-administratiu>

Por ejemplo, el fichero GeoJSON de la región del Alto Pirineo muestra un aspecto similar al siguiente:

```
{ "type": "FeatureCollection", "crs": { "type": "name",  
  "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },  
  "features": [ { "type": "Feature", "properties": { "id": 1,  
    "region": "Alto Pirineo" }, "geometry": { "type": "Polygon",  
    "coordinates": [ [ [ 0.677935221533511, 42.067773035152726 ],  
    [ ..., ... ], ... ] ] }
```

El fichero correspondiente a los centros de atención hospitalaria tiene las siguientes características:

- *athospitalaria.geojson*: centros de atención hospitalaria. Contiene las siguientes características:
 - *properties*: sólo se enumerarán las que tengan algún tratamiento en este trabajo. Por ejemplo: *name*, *description* e *icon*, que son el nombre, la descripción e icono del marcador que representa a un centro de atención hospitalaria.
 - *geometry*: de tipo punto, contiene la localización geográfica de cada de centro.

Toda esta información se utilizará desde el visor web con la ayuda de las librerías especializadas en tratamiento de mapas que se han seleccionado: LeafLet, Turf y D3.

A continuación se enumeran los pasos que se han seguido para obtener la información geoespacial en un formato compatible con el visor web:

- Implantar y configurar MapServer.
- Obtener la cartografía base OSM.
- Instalar PostgreSQL con el complemento para datos geoespaciales PostGIS.
- Utilizar herramientas de importación de datos geoespaciales como *imposm*¹³ v.2. Con esta herramienta se han pasado los datos geoespaciales a la base de datos PostGIS.
- Obtener las capas vectoriales con información sobre los centros de atención hospitalaria y primaria.
- Obtener y publicar la cartografía base, ortográfica y topográfica, de interés.
- Delimitar las regiones y sectores sanitarios, almacenando las capas vectoriales en formato GeoJSON.

¹³ <http://imposm.org>

La fuente predeterminada de cartografía (WMS) proviene del ICGC y como fuente alternativa de cartografía se utilizará la base de datos geoespacial que se ha creado antes. Para ello, se han generado ficheros de configuración de mapas (Mapfiles) que se ejecutarán en el servidor de mapas local (MapServer) y que se conectarán a la base de datos.

Los datos correspondientes a las regiones sanitarias y centros de atención hospitalaria se importarán desde los ficheros en formato GeoJSON. De este modo podremos obtener información dinámica de estos elementos en distintas capas.

La base de datos geoespaciales denominada 'osm' está compuesta por una serie de tablas cuya información se utilizará para crear capas vectoriales. A continuación se enumeran las tablas utilizadas y se describen brevemente los elementos que contiene cada una de ellas:

- *osm_new_places*: contiene información geoespacial relativa a lugares. Localidades o municipios (locality), estados (state), países (country), condado (county), pueblos o municipios (town), aldeas (hamlet), ciudades (city), villas y pedanías (village), suburbios (suburb) y comunidades autónomas (region).
- *osm_new_waterareas*: almacena información geoespacial sobre embalsamientos de agua. Depósitos (reservoir), cuencas hidrográficas (basin) y orillas de ríos (riverbank).
- *osm_new_landusages*: información geoespacial sobre usos de la tierra. acera (footway), jardín (garden), área peatonal (pedestrian), zonas comerciales (commercial), cines (cinema), escuelas (school), bibliotecas (library), etc.
- *osm_new_mainroads*: información geoespacial sobre carreteras primarias (primary), secundarias (secondary), terciarias (terciary), nodos de enlace primarios (primary_link) y nodos de enlace secundarios (secondary_link).
- *osm_new_waterways*: información geoespacial sobre corrientes de agua. Drenajes (drain), canales (canal), ríos (river) y arroyos (stream).
- *osm_new_motorways*: información geoespacial sobre autopistas (motorway), vías troncales (trunk), enlace troncal (trunk_link) y enlace de autopista (motorway_link).
- *osm_new_railways*: información geoespacial sobre tranvías. Metro (subway), monorail (monorail), tranvía (tram), funicular (funicular), etc.
- *osm_new_aeroways*: información geoespacial sobre aeropuertos. Pista de aterrizaje (runway) y pista de rodaje (taxiway).
- *osm_new_minorroads*: información geoespacial sobre vías secundarias. Caminos (path), zonas residenciales (residential), carreteras (road), etc.

Los datos geoespaciales generales, los que nos servirán para mostrar el mapa, se obtendrán de ficheros estáticos 'shapefiles' y de conexiones a la base de datos

'osm'. De este último origen de datos se obtendrán las capas vectoriales con información relativa a carreteras, autopistas, usos de la tierra, ciudades, etc.

Por cuestiones de rendimiento no se cargarán todas las capas del mapa, se excluirán capas poco representativas en el contexto de este trabajo. Por ejemplo no se incluirán las siguientes capas: embalsamientos de agua, ferrocarriles, aeropuertos y carreteras secundarias. Todas las capas mencionadas se generan a partir de conexión a la base de datos geoespacial. La carga de capas desde la base de datos suele ser un 10% más lenta que la carga de capas desde ficheros 'shape', requieren un procesamiento adicional entre el servidor de la base de datos y el servidor de mapas.

Además de la reducción del número de capas se llevan a cabo otras dos tareas de optimización del rendimiento: generación de índices en los campos de consulta y creación de una conexión a la base de datos reutilizable por las distintas capas. La primera tarea se realizará sobre las tablas que se usan como origen de las capas, teniendo en cuenta los campos de consulta. La segunda tarea se realizará en el fichero de configuración del mapa, incluyendo la directiva PROCESSING "CLOSE_CONNECTION=DEFER" en las conexiones a la base de datos 'gis'.

Los ficheros de script para mapas (Mapfiles) son un componente esencial para MapServer. En ellos se definen las relaciones entre objetos, los punto de localización de datos y qué elementos deben dibujarse en los mismos.

Los Mapfiles se construyen utilizando un conjunto de etiquetas y atributos determinados (MapServer no es sensible a mayúsculas), que sirven para definir objetos con sus características.

A modo de introducción a los ficheros de script de mapas mostraremos un sencillo ejemplo¹⁴ sobre el que nos centraremos para describir la estructura básica.

```
MAP
NAME "AtlasSanitario"
FONTSET "fonts.lst"
IMAGETYPE png
MAXSIZE 4000
SIZE 800 800
EXTENT -556126 4532723 469622 5292984
UNITS meters
IMAGECOLOR 153 179 204
SHAPEPATH "/home/rafa/gis/basemaps/data"
WEB
  METADATA
    "ows_enable_request" "*"
    "wms_srs" "EPSG:900913 EPSG:4326 EPSG:3857 EPSG:2154
EPSG:310642901 EPSG:4171 EPSG:310024802 EPSG:310915814
EPSG:310486805 EPSG:310702807 EPSG:310700806 EPSG:310547809
EPSG:310706808 EPSG:310642810 EPSG:310642801 EPSG:310642812
EPSG:310032811 EPSG:310642813 EPSG:2986"
    "labelcache_map_edge_buffer" "-10"
    "wms_title" "WMS Atlas Sanitario Server"
  END
```

¹⁴ Puede encontrar el ejemplo en <http://www.mapserver.org/introduction.html#introduction>

```

IMAGEPATH "/var/www/html/tmp/ms_tmp/"
IMAGEURL "/tmp/ms_tmp/"
END
LAYER
STATUS ON
GROUP "default"
TYPE POINT
CONNECTIONTYPE POSTGIS
CONNECTION "host=localhost dbname=osm user=postgres
password=Hola1234 port=5432"
NAME "places1"
MINSCALEDENOM 166404102
MAXSCALEDENOM 332808204
DATA "geometry from (select * from osm_new_places where type
in ('country','continent') and name is not NULL order by
population asc nulls first) as foo using unique osm_id using
srid=3857"
LABELITEM 'name'
CLASSITEM 'type'
CLASS
EXPRESSION 'continents'
LABEL
FONT scb
TYPE TRUETYPE
ENCODING "utf-8"
SIZE 8
COLOR 100 100 100
OUTLINECOLOR -1 -1 -1
OUTLINEWIDTH 1
BUFFER 4
PARTIALS FALSE
POSITION cc
END
END
END
END # MAP

```

El objeto más importante es el mapa que lo define la etiqueta MAP, esta etiqueta se debe finalizar con la palabra END. Algunos de los atributos del mapa son:

- *NAME "AtlasSanitario"*: nombre del mapa.
- *EXTENT -180 -90 180 90*: extensión del mapa expresada en unidades de salida del mapa.
- *SIZE 800 400*: tamaño en pixels de la imagen del mapa.
- *IMAGECOLOR 128 128 255*: color de fondo de la imagen del mapa.
- *METADATA*
 - *ows_enable_request*: tipos peticiones de servicio aceptadas (*=todas). Servicios como: *GetCapabilities*, *GetMap*, *GetFeatureInfo* y *GetLegendGraphic*.
 - *wms_srs*: define los sistemas de proyección compatibles con los datos.

- `labelcache_map_edge_buffer`: memoria reservada para almacenar información sobre los límites de cada imagen del mapa. Es útil cuando se trabaja con teselas.
- `wms_title`: título del mapa para el servicio WMS.

La etiqueta WEB nos permite configurar características de contenidos del servidor web para la aplicación CGI del servidor de mapas.

- `IMAGEPATH "/var/www/html/tmp/ms_tmp/"`: ruta donde se generarán las imágenes de mapas.
- `IMAGEURL "/tmp/ms_tmp/"`: URL del servidor basada en el sitio web de trabajo donde se generarán las imágenes de mapas.

Otro de los objetos importantes son las capas, que se introducen con la etiqueta LAYER y su definición también termina con END. Algunos de los atributos de este objeto son:

- `NAME "places1"`: nombre de la capa.
- `TYPE RASTER`: tipo de la capa, en este caso de tipo raster.
- `STATUS DEFAULT`: estado de la capa (ON, OFF o DEFAULT). El estado DEFAULT está reservado para depuración, lo que permite que se muestren mensajes de error. El estado ON es el que se debe asignar una vez se haya terminada la depuración, y el estado OFF nos permite desactivar el procesado de la capa y visualización.
- `DATA "[borders.shp] | [geometry from ...]"`: nombre del fichero con los datos geoespaciales, si son de tipo *shape* no es necesario indicar la extensión.

Finalmente la etiqueta CLASS (aunque no aparece en el ejemplo), junto con la etiqueta STYLE, nos permite definir objetos geométricos (círculos, líneas, etiquetas, etc.). Por ejemplo, podríamos definir un círculo de 5 píxeles de diámetro, con un color específico (RGB(200,200,200)), escribiendo:

```
CLASS
  EXPRESSION 'city'
  LABEL
  FONT sc
  TYPE TRUETYPE
  ENCODING "utf-8"
  SIZE 8
  COLOR 68 68 68
  OUTLINECOLOR 255 255 255
  OUTLINEWIDTH 2
  WRAP ' '
  MAXLENGTH 8
  ALIGN CENTER
  BUFFER 4
  PARTIALS FALSE
  POSITION auto
  STYLE
```



```

SIZE 5
SYMBOL "citycircle"
COLOR 200 200 200
OUTLINECOLOR 0 0 0
END
END
END

```

En el ejemplo anterior también hemos seleccionado datos del conjunto asociado con la capa. Para ello se ha utilizado la etiqueta EXPRESSION, con valor 'city', para adjuntar una etiqueta aplicar el estilo comentado anteriormente. Es decir por cada ciudad se mostrará una etiqueta con su nombre y un círculo que representa su posición geográfica.

Otro ejemplo de capa (land0) creada a partir de datos estáticos sería el siguiente:

```

LAYER
TYPE POLYGON
STATUS ON
GROUP "default"
DEBUG 1
NAME "land0"
PROCESSING "APPROXIMATION_SCALE=full"
PROJECTION
  "init=epsg:4326"
END
MINSCALEDENOM 332808204
MAXSCALEDENOM 9999999999
DATA "data/TM_WORLD_BORDERS-0.3.shp"
CLASS
STYLE
  COLOR 242 239 233
END
END
END

```

La etiqueta DATA apunta un fichero con formato 'shape' que contiene información geográfica sobre los límites de los continentes. Hay que destacar que el sistema de proyección elegido para esta capa es el EPSG:4326, correspondiente a las coordenadas geográficas WGS84. Las etiquetas MINSCALEDENOM y MAXSCALEDENOM determinan a partir de qué escala será visible la capa. Finalmente, es importante destacar el papel del valor de la etiqueta PROCESSING, 'APPROXIMATION_SCALE=full', dicho valor nos permite ajustar la resolución de los datos a la resolución del mapa.

Los Mapfiles pueden ser estáticos, obtiene la definición de todos los objetos geospaciales desde ficheros, o dinámicos, la información se obtiene de una fuente de datos externa (por ejemplo PostGIS). Para lo cual tendremos que introducir etiquetas de conexión como CONNECTIONTYPE y CONNECTION dentro de la definición de cada capa.

En este trabajo se utilizan fuente de datos cartográfica del servidor de mapas local y fuentes de datos estáticas constituidas por ficheros en formato 'shape'. La fuente de datos dinámica la constituyen los datos geospaciales que se han

almacenado en una base de datos PostGIS denominada 'osm'. Por lo que se realizan conexiones a dicha base de datos para extraer la información que corresponda en cada caso. A continuación se muestra un ejemplo general de una capa que obtiene los datos de una base de datos PostGIS.

```
LAYER
  NAME "region"
  STATUS ON
  TYPE POLYGON
  CONNECTIONTYPE POSTGIS
  CONNECTION "host=ubuntu-server port=5432 dbname=osm
  user=postgres password=Hola1234"
  DATA "geometry from (select geometry,osm_id ,name as
  name,type from osm_new_waterareas)"
  PROCESSING "CLOSE_CONNECTION=DEFER"
  CLASS
    ...
  END
END
```

A continuación se describen las estructuras de algunas de las capas incluidas en el Mapfile que se utiliza en la fuente de datos del servidor de mapas local :

- *Ciudades:*

```
LAYER
  STATUS ON
  GROUP "default"
  TYPE POINT
  CONNECTIONTYPE postgis
  CONNECTION "host=localhost dbname=osm user=postgres
  password=Hola1234 port=5432"
  NAME "places4"
  MINSCALEDENOM 20800512
  MAXSCALEDENOM 41601025
  DATA "geometry from (select * from osm_new_places where type
  in ('country','continent','city') and name is not NULL order
  by population asc nulls first) as foo using unique osm_id
  using srid=3857"
  LABELITEM 'name'
  CLASSITEM 'type'
  CLASS
    EXPRESSION 'city'
    LABEL
      FONT sc
      TYPE TRUETYPE
      ENCODING "utf-8"
      SIZE 8
      COLOR 68 68 68
      OUTLINECOLOR 255 255 255
      OUTLINEWIDTH 2
      WRAP ' '
      MAXLENGTH 8
      ALIGN CENTER
      BUFFER 4
      PARTIALS FALSE
      POSITION auto
```

```
STYLE
  SIZE 5
  SYMBOL "citycircle"
  COLOR 200 200 200
  OUTLINECOLOR 0 0 0
END
END
END
```

- *Lagos, pantanos, etc.:*

```
LAYER
  TYPE POLYGON
  STATUS ON
  PROJECTION
    "init=epsg:3857"
  END
  NAME "waterarea5"
  GROUP "default"
  CONNECTIONTYPE POSTGIS
  CONNECTION "host=localhost dbname=osm user=postgres
password=Hola1234 port=5432"
  DATA "geometry from (select geometry,osm_id ,name as
name,type from osm_new_waterareas) as foo using unique osm_id
using srid=3857"
  MAXSCALEDENOM 20800512
  MINSCALEDENOM 10400256
  PROCESSING "CLOSE_CONNECTION=DEFER"
  LABELITEM "name"
  CLASSITEM "type"
  CLASS
    EXPRESSION "riverbank"
    STYLE
      COLOR 153 179 204
      OUTLINECOLOR "#B3C6D4"
      OUTLINEWIDTH 0
    END
  END
  CLASS
    STYLE
      COLOR 153 179 204
      OUTLINECOLOR "#B3C6D4"
      OUTLINEWIDTH 0
    END
  END
END
```

- *Autopistas y autovías:*

```
LAYER
  GROUP "default"
  STATUS ON
  TYPE LINE
  MINSCALEDENOM 10400256
  MAXSCALEDENOM 20800512
  NAME "roads5"
```

```

CONNECTIONTYPE postgis
CONNECTION "host=localhost dbname=osm user=postgres
password=Hola1234 port=5432"
DATA "geometry from (select osm_id,geometry,name as
name,ref,type from osm_new_mainroads where type in
('trunk','motorway') order by z_order asc) as foo using unique
osm_id using srid=3857"
LABELITEM "name"
CLASSITEM "type"
CLASS
  EXPRESSION "motorway"
  TEXT ("[ref]")
  STYLE
    WIDTH 0.5
    COLOR 253 146 58
  END
END
CLASS
  EXPRESSION "trunk"
  TEXT ("[ref]")
  STYLE
    WIDTH 0.5
    COLOR 255 195 69
  END
END
END

```

4.2. VISOR WEB

Quizás una de las principales características del visor web sea que pueda ser utilizado desde distintos tipos de pantallas, esto es algo que actualmente se conoce como diseño 'responsive'. Para realizar un desarrollo completo de interfaces web que cumplan con esta característica puede ser necesario un gran esfuerzo y tiempo de desarrollo. No es objetivo principal de este trabajo la especialización en técnicas de programación en HTML5 y CSS3, estándares que nos permitirían alcanzar dicho objetivo. Se ha optado por utilizar una plantilla como base para el desarrollo del visor web y centrarnos en el manejo de las librerías especializadas en el tratamiento de mapas.

ArcGIS¹⁵ es una completa plataforma de información que permite crear, analizar, almacenar y difundir datos, modelos, mapas y globos en 3D, poniéndolos a disposición de todos los usuarios. Desde esta plataforma se ha obtenido una plantilla 'responsive' para el visor web, bajo licencia Apache License 2.0¹⁶.

En primer lugar se tratarán los aspectos relacionados con la interfaz de usuario, después se describirán las funciones de utilidad y finalmente se comentarán los scripts (cliente y servidor) que hacen posible la funcionalidad.

¹⁵ <http://www.esri.es>

¹⁶ <http://www.apache.org/licenses/LICENSE-2.0.html>

Interfaz de usuario

El diseño de la plantilla se adapta a diferentes tipos de pantalla, se modifican los controles de interacción para hacer la interfaz muy usable e intuitiva. La interfaz de usuario se estructura de la siguiente manera (figura 9):

- Mapa (1): operaciones y controles habituales en este tipo de contenidos (zoom, movimiento de mapa, control de capas, puntos de información, etc.).
- Barra de herramientas (2): permite seleccionar el mapa base, realizar mediciones entre dos puntos y geolocalizaciones.
- Panel de control (3): información general sobre la cartografía que se está visualizando, consultas de datos sanitarios, información sobre leyendas y configuración de impresión.
- Localización en el mapa (4): información de la posición geográfica basada en la posición del puntero del ratón, escala del mapa y zoom actual.



Figura 9: interfaz de usuario del visor web

En la figura 10 se muestra la interfaz de usuario para calcular la cobertura asistencial basada en los centros de atención hospitalaria. Las consultas se desarrollan entorno a dos campos de filtro: tipo de hospitalización y temporada.

▼ **Cobertura Asistencial**

Calcular

Tipo Hosp.

Temporada

► **Riesgo de epidemias**

► **Gasto sanitario**

Figura 10: interfaz de usuario para calcular la cobertura asistencial

El riesgo de epidemias (figura 11) dispone de 3 campos de filtro: tipo de enfermedad contagiosa, porcentaje de afectados y temporada. Al igual que en la interfaz anterior, los valores de los campos de filtro deben ser seleccionados de una lista, el usuario no puede introducir valores arbitrarios.

The screenshot shows a web interface with a sidebar on the left containing two expandable sections: 'Cobertura Asistencial' and 'Riesgo de epidemias'. The 'Riesgo de epidemias' section is expanded, revealing a green 'Calcular' button at the top. Below the button are three dropdown menus: 'Enfermedad' with the value 'Todas', 'Casos (%)' with the value 'Todos', and 'Temporada' with the value 'Todas'. Below these filters is another sidebar section labeled 'Gasto sanitario'.

Figura 11: interfaz de usuario para calcular el riesgo de epidemias

La interfaz para las consultas sobre el gasto sanitario (figura 12) incluye 4 campos de filtro: indicador de gasto, temporada, valor de gasto en farmacia y valor de gasto en facturación al CatSalut.

The screenshot shows a web interface similar to the previous one. The sidebar on the left has 'Cobertura Asistencial' and 'Riesgo de epidemias' collapsed, and 'Gasto sanitario' expanded. The 'Gasto sanitario' section contains a green 'Calcular' button and four dropdown menus: 'Indicador' with the value 'Selección', 'Temporada' with 'Todos', 'Farmacia' with 'Todos', and 'CatSalut' with 'Todos'.

Figura 12: interfaz de usuario para calcular el gasto sanitario

Las leyendas se describen en una interfaz similar a la de las consultas (figura 13). La descripción de las leyendas se organiza dentro del contexto asociado.

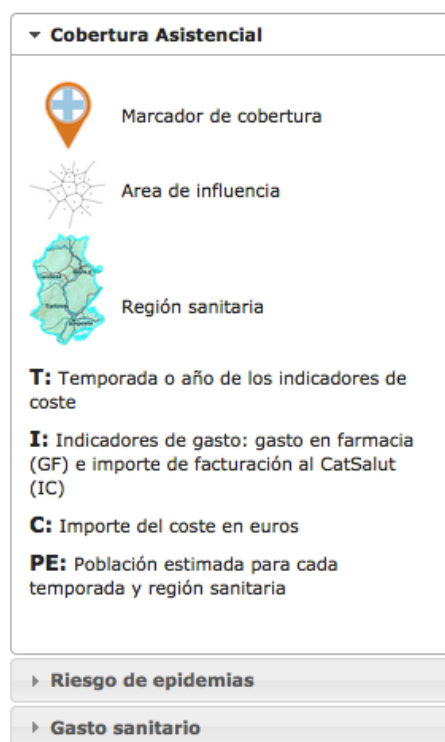


Figura 13: interfaz de usuario con información sobre los símbolos y leyendas del mapa.

El diseño de la interfaz de usuario se ha llevado a cabo utilizando hojas de estilo sobre el DOM (Document Object Model) de los documentos HTML. Además se utilizan algunas librerías Javascript, como jQuery, que permiten interactuar con los objetos del DOM.

Funciones

- *Cambiar fuente cartográfica:* esta función cambia la fuente de información cartográfica. Se puede seleccionar la fuente cartográfica del servidor local de mapas o bien la del ICGC.
- *Geolocalización basada en palabras:* es un buscador de localizaciones geográficas basado en el nombre de ciudades, municipios, lugares, etc. Si la búsqueda tiene resultados satisfactorios se realizará un zoom automático y se mostrará la localización aproximada en el área central del visor.
- *Zoom sobre el mapa:* función de zoom sobre el mapa, se muestra la escala correspondiente en la parte inferior de la interfaz.
- *Movimiento del mapa:* nos permite desplazar el mapa dentro del área de visualización.
- *Visualización y ocultación de capas vectoriales e imágenes ráster:* esta función permite cambiar las capas base del mapa, sólo se podrá visualizar una al mismo tiempo.
- *Calcular la cobertura asistencial e influencias de los centros sanitarios:* para determinar el área de influencia de los centros sanitarios (C1, C2, C3,...) se calcularán las celdas generadas por el algoritmo de Voronoi

(figura 14). De manera resumida, el diagrama mostrará los puntos geográficos a menor distancia de cada centro sanitario. El número de hospitalizaciones registradas en cada centro nos dará información sobre la cobertura asistencial del mismo.

El área de influencia de cada centro de atención hospitalaria se calcula sobre las regiones sanitarias seleccionadas. Es uno de los cálculos más complejos ya que requiere en primer lugar calcular el área de influencia, obtener la población estimada en cada área de influencia (por temporada) y finalmente realizar los cálculos. Estos cálculos se almacenan en objetos JSON que se reutilizarán para mostrar los gráficos estadísticos correspondientes.

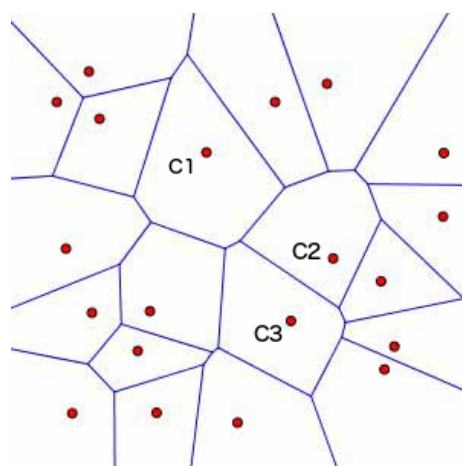


Figura 14: representación gráfica de las áreas de influencia de los centros sanitarios mediante diagramas de Voronoi

- *Calcular el riesgo de epidemias:* al igual que en el caso anterior es necesario conocer la población estimada de cada región, pero ahora lo obtenemos desde la base de datos si no se ha obtenido antes. Para optimizar el rendimiento cada función de cálculo comprueba si ya se ha calculado la población de las regiones sanitarias, si no se ha realizado lo hace y almacena los resultados. Estos resultados podrán ser utilizados para el cálculo de la cobertura o el gasto sanitario. Se utilizan los indicadores sanitarios asociados con el número de casos registrados en distintas temporadas, regiones sanitarias y tipos de enfermedad. La consulta se debe realizar sobre un tipo de enfermedad, y se pueden filtrar los datos por temporada y porcentaje de afectados.

La prevalencia se calcula a partir de los afectados de una región y la población de la misma.

$$Prev = \frac{af_exposed}{pob}$$

Prev: Prevalencia.

af_exposed: afectados expuestos a la enfermedad.

pob: población objetivo.

El riesgo relativo se calcula definiendo como población no expuesta a la enfermedad todos los que son de otra región sanitaria distinta a la de estudio. Y la población afectada por la enfermedad y no expuesta a la misma son todos aquellos que han contraído la enfermedad y no son de la región de estudio.

$$RR = \frac{\frac{af_exposed}{(af_exposed + naf_exposed)}}{\frac{af_notexposed}{(af_notexposed + naf_notexposed)}}$$

af_exposed: afectados y expuestos a la enfermedad.

naf_exposed: no afectados y expuestos.

af_notexposed: afectados y no expuestos.

naf_notexposed: no afectados y no expuestos.

También se calcula el porcentaje de afectados por cada región y temporada.

- *Calcular el gasto sanitario*: se calcula el gasto sanitario por temporada e indicador e gasto. Los indicadores de gasto utilizados son: gasto en farmacia y gasto de facturación al CatSalut. Se tiene en cuenta la población de la región sanitaria que se consulta para calcular el gasto medio por habitante, y por habitante y tipo de indicador de gasto.
- *Buscar el centro hospitalario más cercano*: esta función busca el centro de atención primaria más cercano a la posición indicada por el usuario. Para ello calcula el tiempo de desplazamiento y distancia a todos los centros de atención primaria, devolviendo el que se encuentra a menor tiempo de desplazamiento. Se utiliza el servicio gratuito Nominatim¹⁷ de Open Street Map. Este servicio devuelve un objeto JSON como resultado, que se evalúa para determinar si se ha encontrado algún centro de atención hospitalaria cercano. Finalmente se muestra gráficamente en el mapa el recorrido desde el punto seleccionado por el usuario al centro más cercano. Además se muestra un panel con la información, paso a paso, del recorrido. Para realizar esta función se ha utilizado la librería 'leaflet-routing-machine.js' basada en Leaflet.
- *Búsqueda (geolocalización)*: esta función se utiliza para mostrar en el mapa la localización asociada con el término de búsqueda introducido, se implementa con la ayuda del plugin OSMGeocoder (Open Source). La geolocalización se realiza con el servicio gratuito Nominatim¹⁸ de Open Street Map.
- *Visualizar leyendas del mapa*: dependiendo de la simbología actual del mapa se muestran las leyendas que correspondan.

¹⁷ Más información en nominatim.openstreetmap.org

¹⁸ Más información en nominatim.openstreetmap.org

Scripts

Las funciones descritas en el apartado anterior se llevan a cabo la ayuda de dos tipos de scripts: de cliente y de servidor. Los scripts de cliente están implementados en Javascript y los de servidor en PHP. Los primeros tienen como objetivo principal la interacción del usuario con la interfaz del visor web: zoom sobre el mapa, mover el mapa, cambiar el mapa base, etc. Los scripts de tipo servidor se utilizan principalmente para obtener los datos de los indicadores sanitarios, calcular la cobertura asistencial, el gasto sanitario, el riesgo de epidemias, etc. Algunas de las funciones de estos scripts se comunican con los scripts de cliente mediante objetos de tipo JSON (Javascript Object Notation), que los utiliza para mostrar dinámicamente los resultados de las consultas de usuario sobre indicadores sanitarios.

Los ficheros de script del lado del cliente son los siguientes:

- *control.OSMgeocoder.js*: plugin para geolocalización, utilizado en la función de búsqueda.
- *d3.min.js*: librería especializada en la manipulación de documentos basados en datos.
- *dimple.v2.1.3.js*: librería basada en D3 para la representación de gráficos a partir de datos.
- *grapher.js*: este script contiene la funcionalidad encargada de representar los gráficos estadísticos. Se utiliza la librería Dimple¹⁹, complementaría a D3, que simplifica el uso de la misma para la generación de gráficos.
- *health-openatlas-1.0.js*: incluye las funciones Javascript relacionadas con el cálculo de la cobertura asistencial, el riesgo de epidemias, gasto sanitario y la búsqueda del centro sanitario más cercano. Prepara los parámetros GET o POST que se envían en peticiones AJAX a los scripts de servidor. Los resultados se recogen en objetos serializados en formato JSON.
- *jquery-2.1.3.min.js*: núcleo de jQuery, contiene la mayoría de las funciones de la API.
- *jquery-ui.min.js*: complemento jQuery especializado en funciones para la interfaz de usuario.
- *leaflet-routing-machine.js*: librería para calcular la distancia y tiempo mínimo al centro de atención primaria más cercano.
- *leaflet.ajax.js*: librería para el manejo de mapas.
- *map.js*: incluye las funciones Javascript relacionadas con mapas. Inicializa y configura el mapa, almacena propiedades de estado y define eventos del mapa.

¹⁹ Librería Open Source desarrollada por [PMSI-AlignAlytics](#)

- *page.js*: incluye las funciones Javascript relacionadas con la interfaz de usuario. Principalmente se utilizan miembros de las librerías de jQuery.
- *page-header.js*: contiene las funciones de utilidad para la interfaz de usuario.
- *turf.min.js*: librería con funciones para el análisis de datos geospaciales.
- *voronoi.js*: incluye funciones de las librerías D3 (*d3.min.js*) y Turf (*turf.min.js*). Las funciones de la librería D3 se utilizan para cargar los datos, en formato CSV, con la información geográfica de los centros de atención hospitalaria. Información que después se utilizará para calcular las celdas de Voronoi. La librería Turf se utiliza para calcular si un punto está dentro de una región sanitaria o en el mapa.

La búsqueda de un punto en el interior de una región sanitaria se realiza para estimar la población asociada con la misma. Para ello se obtienen los datos de población de los municipios de Cataluña y se cruzan las coordenadas de los mismos con la región sanitaria. Se contabiliza la población de los municipios. Posteriormente este dato se utiliza en diferentes cálculos: riesgo relativo de epidemias, cobertura asistencial, gasto medio por habitante, etc.

Se ha utilizado como interfaz de desarrollo Eclipse Luna apropiado para trabajar con proyectos PHP. Los contenidos se han organizado según el tipo de fichero y el modo en que participan en el proyecto (figura 15). Para gestionar el acceso a los datos de los indicadores sanitarios se ha utilizado el patrón de diseño 'Proxy'. Este patrón consiste en interponer un intermediario (Proxy) entre un objeto y los demás que lo utilizan (figura 16). Algunas de sus ventajas como su sencilla estructura, facilidad para el manejo de referencias y control de creación de objetos, han contribuido a la decisión de utilizarlo.

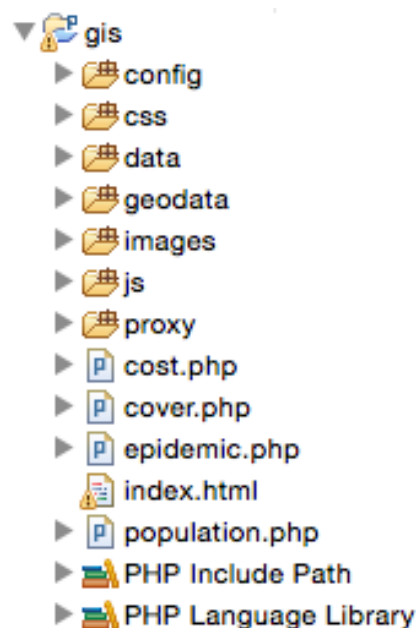


Figura 15: estructura de ficheros del patrón MVC

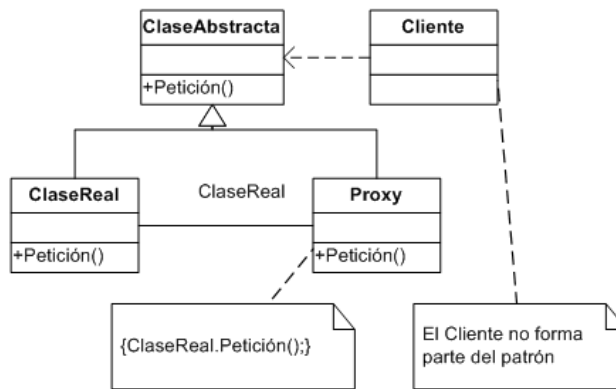


Figura 16: diagrama general del patrón Proxy.

Los ficheros que componen el patrón 'Proxy' son los siguientes:

- *cataloniahealth.php*: representa a la clase del cliente.
- *health.php*: representa a la clase real.
- *ihealth.php*: representa a la clase abstracta.
- *proxy.php*: representa a la clase del proxy.

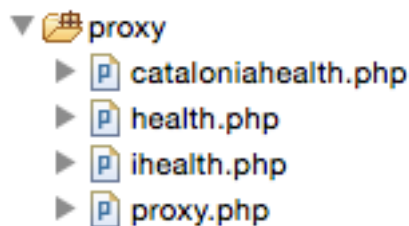


Figura 17: ficheros PHP del patrón Proxy.

La instanciación de la clase 'Cliente' del proxy se realiza desde los ficheros de script correspondientes a las cuatro funciones principales del visor: *cost.php*, *cover.php*, *epidemic.php* y *population.php*. Los cuatro realizan funciones similares, en el cuadro siguiente se muestra el código correspondiente al fichero 'cover.php':

- Comprobar el método de invocación HTTP (POST).
- Instanciar la clase del 'Cliente' del proxy, 'CataloniaHealth'.
- Comprobar los parámetros recibidos: *params* y *regions*. El primer parámetro almacena las distintas opciones de filtro de la interfaz (temporada, porcentaje de riesgo, tipo de enfermedad, etc.). El segundo parámetro contiene las regiones seleccionadas para la consulta.
- Invocar al método 'calcCover' que se encarga de iniciar el cálculo de la cobertura asistencial. Los resultados se serializan antes de devolverlos a la petición AJAX que se inicia desde la interfaz del visor (*health-openatlas-1.0.js*).

```

<?php
require_once 'config/global.php';
header("Content-type: text/javascript");
if ( $_SERVER['REQUEST_METHOD'] === 'POST' ) {
    if( isset($_POST['params']) &&
        isset($_POST['regions']) ) {
        $data = new CataloniaHealth();
        $params = $_POST['params'];
        $regions = $_POST['regions'];
        echo json_encode($data->calcCover($params,
$regions),JSON_FORCE_OBJECT);
    }else{
        echo json_encode(['error'=>'Parámetros
incorrectos']);
    }
}else
    echo json_encode(['error'=>'Método de llamada
incorrecto']);
?>

```

En el lado del servidor hay otros ficheros de script que se utilizan para configuración de propiedades genéricas (datos de conexión a la base de datos y referencia de ficheros). La carpeta 'config' contiene dichos ficheros:

- *database.php*: contiene un array con la configuración de conexión a la base de datos PostgreSQL.
- *global.php*: contiene la definición del controlador y acciones por defecto.

El último fichero de script de servidor está en la carpeta 'data', se denomina 'connection.php' y almacena la clase de conexión a la base de datos PostgreSQL. Esta clase contiene un único método encargado de la conexión (connect) y que se invoca desde la clase real del patrón proxy (Health).

Los datos geospaciales de las regiones sanitarias se almacenan en ficheros GEOJSON en la ruta /geodata/layers (figura 18).

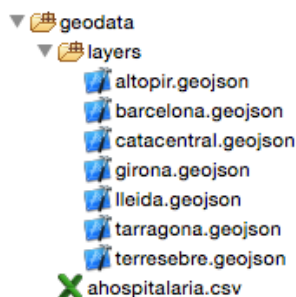


Figura 18: ubicación de los ficheros de datos geospaciales de las regiones sanitarias.

Los datos geoespaciales de los centros de atención hospitalaria se almacenen en un fichero con formato CSV (Comma Separator Values), se encuentran en la ruta /geodata (figura 18).

4.3. SERVIDOR DE MAPAS

Antes de comenzar el proceso de instalación de MapServer v.6.4 es conveniente instalar un servidor web compatible con CGI (Common Gateway Interface), en este trabajo se ha seleccionado Apache Web Server 2. Este servidor web es compatible con MapServer, soporta CGI y no se distribuye bajo licencia comercial, estas son algunas de las razones de la elección. También se instalará PostgreSQL, la última versión en el momento de escribir este documento es la 9.4.1. Finalmente se instalará el complemento PostGIS que se utilizará como ayuda para la creación de base de datos geoespaciales.

Para acelerar el resultado de las consultas de datos geoespaciales se habilitará la función de caché de MapServer, denominada MapCache. Esta función está especializada dividir el contenido de las capas en teselas o fragmentos (tiles) de menor e igual tamaño. Con ello se consigue aumentar la rapidez con la que se obtienen los resultados de una consulta WMS.

En este apartado también se incluirán los ficheros de script para mapas (Mapfiles), utilizados por el servidor para mostrar la cartografía base y las capas vectoriales que contienen la definición de las regiones, sectores y centros sanitarios.

Instalación de software

A continuación se enumera el software seleccionado y los pasos seguidos para la instalación de los mismos.

- Instalación y prueba de PostgreSQL

- Instalación

```
sudo apt-get update  
sudo apt-get install postgresql postgresql-contrib
```

- Prueba

```
sudo -i -u postgres
```

- Instalación de PostGIS

```
sudo apt-get install postgresql-9.4-postgis postgresql-server-dev-9.4
```

- Instalación y prueba de Apache Web Server 2

```
sudo apt-get install apache2
```

Para probar la instalación basta con abrir un navegador de Internet y escribir *http://localhost* en la barra de dirección. Se mostrará como resultado la página web predeterminada de Apache Web Server.

El proceso de instalación de Mapserver ha sido el siguiente:

- Instalación de Python y carga de repositorios de UbuntuGis

```
sudo apt-get install python-software-properties
sudo add-apt-repository ppa:ubuntugis/ubuntugis-unstable
sudo apt-get update
```

- Instalación y prueba de Mapserver

```
sudo apt-get install cgi-mapserver mapserver-bin
```

Una vez instalado el servidor de mapas se puede probar utilizando un navegador de Internet, en la dirección escribiremos: *http://ubuntu-server/cgi-bin/mapserv?* El servidor web nos devolverá la respuesta del servidor de mapas, que en este caso será similar a: 'No query information to decode. QUERY_STRING is set, but empty'. Este mensaje indica que el servidor de mapas está funcionando pero que no ha recibido parámetros de consulta.

- Instalación de Imposm

Imposm es un componente software que se utilizará para importar los datos geospaciales desde el fichero de cartografía descargado a la base de datos de PostgreSQL. Además se instalan algunas dependencias de Imposm y se configura el entorno de ejecución.

- Instalar dependencias de Imposm

```
sudo apt-get install build-essential python-dev
protobuf-compiler \
libprotobuf-dev libtokyocabinet-dev \
python-psycopg2 libgeos-c1
```

- Configuración del entorno

```
sudo apt-get install python-virtualenv
virtualenv venv
source venv/bin/activate
```

- Instalación de Imposm

```
sudo aptitude install python-pip
sudo pip install imposm
```

- Instalación de MapCache

MapCache es un sistema de cache, complemento de MapServer, que proporciona información cartográfica en formato de teselas o fragmentos (tiles). La finalidad de este sistema es acelerar la respuesta WMS en aplicaciones que necesitan recuperar constantemente información cartográfica, sin pérdida de las principales características de las capas.

Antes de instalar MapCache se debe comprobar la versión del servidor Apache Web Server que tenemos.

```
/usr/sbin/apache2 -V
```

Si tenemos la versión 'threaded' debemos instalar el complemento *apache2-threaded-dev*, en caso contrario instalamos el complemento *apache2-prefork-dev*, ambos complementos de Apache Web Server.

```
sudo apt-get install apache2-threaded-dev
```

O bien:

```
Sudo apt-get install apache2-prefork-dev
```

Además se instalarán otras dependencias, como son:

```
sudo apt-get install libpng12-dev
sudo apt-get install libjpeg62-dev
sudo apt-get install libcurl4-gnutls-dev
```

Después de la instalación de los complementos anteriores se carga la configuración y se reinicia el servicio de Apache Web Server:

```
sudo ldconfig
sudo service apache2 restart
```

Se mapea el módulo de cacheado en la configuración del servidor web:

```
<IfModule mapcache_module>
  <Directory /path/to/directory>
    Order Allow,Deny
    Allow from all
  </Directory>
  MapCacheAlias /mapcache "/path/to/directory/mapcache.xml"
</IfModule>
```

Se reinicia el servicio HTTP de Apache Web Server:

```
sudo apachectl restart
```

Y finalmente se añade el módulo fast-cgi al servidor web:


```
sudo apt-get install libapache2-mod-fcgid
edit httpd.conf with module info
```

Ficheros de script para mapas

El fichero de configuración del mapa, `atlasanitario.map`, está compuesto por 95 capas unas enlazadas a fuentes de datos estáticas y otras a fuentes dinámicas.

El mapa publicado en el servidor local lo podemos visualizar en el navegador, para ello introducidos la siguiente URL:

```
http://ubuntu-server/cgi-bin/mapserv?map=/home/rafa/gis/basemaps/atlassianitario.map&layers=all&mode=map
```

Descomponiendo la petición GET de la URL anterior tenemos:

- `ubuntu-server`: nombre del host donde está instalado MapServer y alojado el mapa.
- `/cgi-bin/mapserv?`: dirección de la aplicación del servidor de mapas.
- `map=/home/rafa/gis/basemaps/atlassianitario.map`: parámetro en el que se indica la dirección completa donde se encuentra el mapa.
- `layers=all`: este parámetro recoge el nombre de la capa que se quiere mostrar, o bien todas (all).
- `mode=map`: modo elegido para mostrar el mapa. Admite multitud de valores (recogidos en la documentación del CGI de Mapserver) que nos permiten cargar plantillas HTML, etiquetas, leyendas, iconos de leyendas, etc.

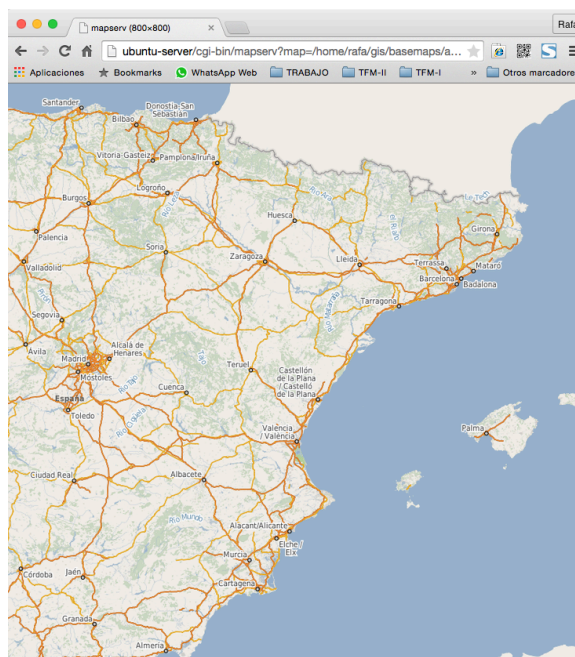


Figura 19: visualización del mapa en un navegador.

5. PRUEBAS Y RESULTADOS FINALES

Las pruebas representan un pilar fundamental en el desarrollo software en cualquier metodología. Durante las pruebas se ha conmutado entre el perfil de usuario y desarrollador con el objeto de minimizar los errores y comprobar que los resultados son aceptables.

Se han realizado tres tipos de pruebas:

- *Pruebas de aceptación*: orientadas a reflejar los requerimientos definidos por el usuario.
- *Pruebas unitarias*: realizadas por el desarrollador, enfocadas a comprobar que se satisfacen correctamente todos los puntos de función.
- *Pruebas de integridad o de conjunto*: la intención ahora es probar la satisfacción de requisitos, cohesión y acoplamiento de funciones relacionadas.

Las primeras pruebas se han realizado en el bloque 4 de las tareas planificadas (véase el apartado 1.4). En dicho bloque se recogen las tareas de implementación del servidor de mapas, del servidor de datos geoespaciales y del servidor de datos alfanuméricos. A continuación se describen los tipos de pruebas realizadas:

- *Implementación del servidor de mapas*
 - *Pruebas unitarias*: han consistido en comprobar cada tarea de implementación del servidor de mapas. Se han ejecutado uno a uno los comandos de instalación y configuración, verificando los resultados (por ejemplo, la instalación y configuración del servidor de mapas se ha comprobado escribiendo la dirección <http://ubuntu-server/cgi-bin/mapserver?> en la barra de dirección del navegador). Para la importación de datos ha sido necesario, además de descargarlos, instalar una herramienta de importación que pase los datos desde un fichero a la base de datos PostGIS. La prueba de esta tarea ha consistido en comprobar que se han creado todos los elementos de la base de datos correspondientes al fichero de datos geoespaciales. De manera similar se han probado los ficheros de mapas, se han ido configurando las distintas capas y probando que se visualizan en el mapa (parámetro layers de la API CGI).
 - *Pruebas de integración*: en primer lugar se ha probado el correcto funcionamiento del servidor de mapas sobre el servidor web, para ello se han ejecutado consultas básicas. Después se han consultado distintas capas y finalmente el mapa completo.
 - *Pruebas de aceptación*: se pueden cargar todas las capas y el mapa completo, las operaciones de desplazamiento y zoom funcionan correctamente (utilizando plantillas y parámetros CGI).

- *Implementación del servidor de datos geoespaciales*
 - *Pruebas unitarias*: tras la ejecución de comandos de instalación y configuración del servidor de bases de datos PostgreSQL, se ha verificado que el servidor de datos escucha en un puerto TCP abierto y sin restricciones. Además se ha comprobado el acceso local y remoto al servidor de datos. Después se ha probado la base de datos geoespaciales, para ello se han realizado distintas consultas con y sin parámetros (pruebas en servidor y cliente remoto), verificando que existen entidades geográficas como: city, countries, town, highway, etc.
 - *Pruebas de integración*: se han programado ficheros de mapas con capas cuyos orígenes de datos están definidos en la base de datos geoespacial. Estos ficheros han sido probados en el servidor de mapas utilizando el navegador de Internet.
 - *Pruebas de aceptación*: han consistido en validar las consultas de datos geoespaciales visualizando sus resultados tanto en el administrador de la base de datos, como en el servidor de mapas (haciendo uso del navegador web).

- *Implementación del servidor de datos alfanuméricos*
 - *Pruebas unitarias*: algunas de las pruebas ya se han realizado en el apartado anterior. Así que las pruebas se han centrado en comprobar el rendimiento y la adecuación del modelo de datos de la base de datos de indicadores sanitarios. Se han probado las consultas y se ha verificado que los conjuntos de datos se corresponden con los tipos asociados.
 - *Pruebas de integración*: han consistido en comprobar que los datos pueden ser consultados desde el servidor local y el equipo cliente.
 - *Pruebas de aceptación*: se ha validado el formato de los datos de respuesta a los distintos tipos de consultas: gasto por temporada, gasto por indicador, cobertura asistencial por temporada, número de casos de una enfermedad contagiosa en una región determinada, etc.

Superadas las pruebas del bloque 4 se consiguen los siguientes resultados: servidor web compatible con el servidor de mapas, servidor de mapas (caché de mapas), ficheros de configuración de mapas, datos geoespaciales importados en la base de datos geoespacial, datos relativos a los indicadores sanitarios y población cargados en la base de datos alfanumérica.

El último bloque de tareas (bloque 5) se centra en la implementación del visor web GIS. Se han realizado las siguientes pruebas:

- *Interfaz de usuario*
 - *Pruebas unitarias*: se ha comprobado el diseño variando el tipo de navegador, tamaño y tipo de dispositivo. También se han probado

los eventos en los distintos elementos de la interfaz (botones, mapa, listas, etc).

- *Pruebas de integración:* se han realizado pruebas sobre cada funcionalidad concreta (cobertura asistencial, gasto sanitario, riesgo de epidemias, centro más cercano, etc.), evaluando los resultados, la interacción, diseño, etc.
- *Pruebas de aceptación:* las pruebas han consistido en verificar el desempeño general de las funciones, la adecuación de los resultados y la calidad del diseño (usabilidad, homogeneidad del diseño, etc.).
- *Lógica de servidor*
 - *Pruebas unitarias:* se han realizado pruebas de los componentes del patrón proxy, comprobando que los parámetros de entrada son adecuados y que los resultados son los que se esperan, y están en el formato adecuado. Los ficheros de script del servidor se han probado desde el navegador web.
 - *Pruebas de integración:* para realizar pruebas de este tipo se ha esperado al desarrollo de la lógica de cliente. Estas pruebas se describen en el siguiente apartado.
 - *Pruebas de aceptación:* parte de este tipo de pruebas se completan en la lógica de cliente, el resto han consistido en validar la entrada y salida de las funciones de la lógica del servidor, en la que intervienen varios scripts relacionados.
- *Lógica de cliente*
 - *Pruebas unitarias:* principalmente han consistido en probar las funciones Javascript, el manejo de objetos del DOM y las reglas de estilo. Estas pruebas han sido realizadas en su mayoría en el navegador web.
 - *Pruebas de integración:* se ha probado la interacción y cohesión de los distintos módulos JavaScript. Por ejemplo, el módulo Javascript del atlas sanitario que gestiona el estado del mismo interactúa con el resto de módulos: módulo de mapa, módulo de gráficos, módulo de Voronoi, etc. Se ha comprobado que los cambios de estado se realizan adecuadamente. Además se prueban, de manera indirecta, el servidor de mapas, el servidor de datos sanitarios, scripts de servidor, la serialización de objetos desde ficheros, etc.
 - *Pruebas de aceptación:* este tipo de pruebas están muy cerca de las pruebas finales ya que, como se ha comentado anteriormente, se prueban todos elementos necesarios para validar si la solución propuesta es adecuada.

6. CONCLUSIONES Y MEJORAS

La primera conclusión de todas, en mi afán por aprender, es que cada vez se menos. El presente trabajo me ha dado la oportunidad de ampliar conocimientos en el tema principal del mismo, los sistemas de información geográfica.

Se han conseguido los objetivos previstos al comienzo de este trabajo. A continuación se asocian las conclusiones y mejoras correspondientes a cada uno de ellos:

- *Diseñar e implantar un servidor de mapas.* La conclusión inmediata respecto de este objetivo general es que es necesario dominar y combinar, con determinado nivel, múltiples tecnologías. Como se ha visto en los apartados anteriores ha sido necesario instalar y configurar un sistema operativo de red, suficientemente estable y compatible con el servidor de mapas. Además, para la instalación del servidor de mapas ha sido necesario conocer comandos específicos del sistema operativo. Con una intención puramente pedagógica se ha instalado un servidor de bases de datos orientado a sistemas GIS, lo que ha permitido disponer de información geoespacial dinámica. La información geoespacial estática se ha tratado con herramientas especializadas (QGIS), generando ficheros con distintos formatos (shape, GEOJSON, etc.). Y la configuración de los elementos de mapas se ha realizado de manera programática en ficheros específicos del servidor de mapas.

Sin conocimientos expertos, ni experiencia suficiente, se puede decir que alcanzar este objetivo ha sido costoso. Sin embargo, se ha conseguido aprender a implementar las características fundamentales de un servidor de mapas. La primera mejora que se podría plantear sería la de seguir profundizando en el tema para:

- Optimizar el rendimiento del servidor de mapas.
- Implementar un WFS para hacer posible la descarga de capas de información geográfica vectorial completa, es decir, la descarga de su geometría y de su tabla de atributos asociada.
- Diseñar e implementar un visor web GIS. Las diferentes librerías y plugins JavaScript existentes hacen relativamente fácil su uso y la obtención de resultados profesionales. En este trabajo se han utilizado, como se había previsto, tres de ellas: Leaflet, D3 y Turf. Se ha conseguido integrar distintas funciones de dichas librerías dentro del diseño web del visor.

Una mejora sería modificar adaptar el plugin de geolocalización utilizado para mostrar los puntos de paso hasta llegar al centro de atención hospitalaria más cercano. En este mismo aspecto, también se puede mejorar la interfaz para introducir una animación que muestre cada punto de paso en el mapa.

- Integrar datos geoespaciales y alfanuméricos procedentes de distintas fuentes. De manera general la mejora más importante consistiría en completar los datos geoespaciales introduciendo mayor nivel de detalle

en el mapa y obtener más datos sanitarios, en especial los relativos a las personas atendidas en los centros de atención primaria. También se podría incluir más dinamismo proporcionando una herramienta de importación de datos sanitarios o bien una herramienta de adquisición de datos desde servidores de fuentes confiables como la del ICGC.

Se ha avanzado en el nivel de conocimiento y experiencia en el tratamiento de tecnologías GIS, y es fácil entender que durante el proceso de aprendizaje es difícil pensar en muchas mejoras, es al final de dicho proceso cuando surgen.

7. GLOSARIO

En este apartado se incluirán definiciones y abreviaturas más relevantes, propias del ámbito de las tecnologías de la información, del ámbito de la información en el contexto sanitario y las relativas al ámbito de la geodesia.

AJAX: es un acrónimo que significa JavaScript asíncrono y XML (Asynchronous JavaScript And XML). Es una tecnología que permite a una aplicación cliente solicitar datos a un servidor sin interferir con la interacción del usuario.

CatSalut: Servicio Catalán de Salud.

CGI: interfaz de entrada común (Common Gateway Interface, abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME²⁰. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGIs.

Componente software: según Sommerville (2007), una unidad de implementación ejecutable, construida mediante un conjunto de clases, que ofrece un conjunto de servicios a través de una interfaz.

Datum: un datum geodésico es una referencia de las medidas tomadas. En geodesia un datum es un conjunto de puntos de referencia en la superficie terrestre con los cuales las medidas de la posición son tomadas y un modelo asociado de la forma de la tierra (elipsoide de referencia) para definir el sistema de coordenadas geográfico. Datum horizontales son utilizados para describir un punto sobre la superficie terrestre. Datum verticales miden elevaciones o profundidades. En ingeniería y drafting, un datum es un punto de referencia, superficie o ejes sobre un objeto con los cuales las medidas son tomadas.

Infraestructura de Datos Espaciales: es un sistema informático integrado por un conjunto de recursos (catálogos, servidores, programas, aplicaciones, páginas web,...) que permite el acceso y la gestión de conjuntos de datos y servicios geográficos (descritos a través de sus metadatos), disponibles en Internet, que cumple una serie normas, estándares y especificaciones que regulan y garantizan la interoperabilidad de la información geográfica

*Diagramas de Voronoi*²¹: también denominados polígonos de Thiessen, nombrados en honor al meteorólogo estadounidense Alfred H. Thiessen, son una construcción geométrica que permite construir una partición del plano euclídeo. Se crean al unir los puntos entre sí, trazando las mediatrices de los segmento de unión.

²⁰ Multipurpose Internet Mail Extension, es un conjunto de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.)

²¹ Fuente de la definición Wikipedia, puede ampliar la definición en <http://es.wikipedia.org>

Document Object Model (DOM): es un modelo de objetos para la representación de documentos. Proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

Elipsoide y Geoide: el elipsoide es una superficie curva cerrada cuyas tres secciones ortogonales principales son elípticas, es decir, son originadas por planos que contienen dos ejes cartesianos. Se denomina geoide al cuerpo de forma casi esférica aunque con un ligero achatamiento en los polos (esferoide), definido por la superficie equipotencial del campo gravitatorio terrestre.

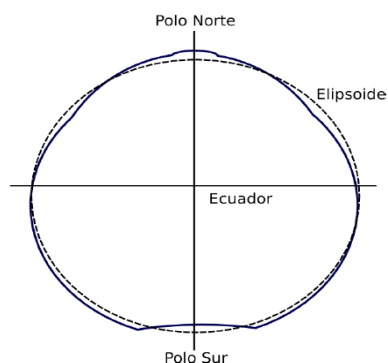


Figura 20: elipsoide y geoide

Fuente: [Navegación Aérea, Cartografía y Cosmografía \(NACC\)](#).

Epidemia: es una enfermedad que se propaga durante un cierto periodo de tiempo en una zona geográfica determinada y que afecta simultáneamente a muchas personas.

*Geodesia*²²: La geodesia es la ciencia que estudia la figura, las dimensiones y el campo gravitatorio de la Tierra, así como su variación en el tiempo.

FastCGI: es una extensión de CGI cuyo objetivo es reducir la carga de procesamiento de las aplicaciones web, permitiendo al servidor web atender más peticiones a la vez.

Indicador de salud: definición de un fenómeno en el ámbito sanitario cuya información asociada puede ser utilizada con objeto de medición o consulta.

Información geográfica: información que contiene datos que poseen una posición implícita (la población de una sección censal, una referencia catastral, etc.) o explícita (coordenadas obtenidas a partir de datos capturados mediante GPS, etc.)

*JavaScript*⁵: es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

*JSON*⁵: acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

²² Definición extraída de UOC. Geodèsia i Cartografia.

Latitud y longitud geográfica: la latitud geográfica (φ) es el ángulo medido sobre el plano meridiano que contiene al punto entre el plano ecuatorial y la normal al elipsoide en P. La longitud geográfica (λ) es ángulo medido sobre el plano ecuatorial entre el meridiano origen y el plano meridiano que pasa por P.

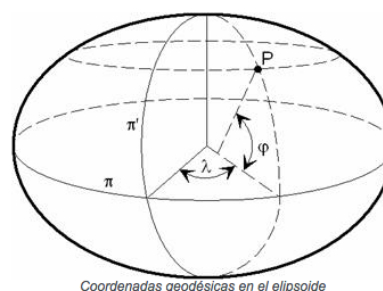


Figura 21: latitud y longitud geográfica

Fuente: [Instituto Geográfico Nacional Geodesia](#).

OSM (Open Street Map): es una plataforma que promueve el crecimiento, desarrollo y distribución de datos geoespaciales libres, para que cualquiera que lo desee pueda usarlos y compartirlos.

PHP: es un lenguaje de scripts de propósito general que es interpretado en el lado del servidor. Está especialmente indicado para el desarrollo web con contenidos dinámicos.

Prevalencia: es la proporción de individuos de una población que presentan un determinado trastorno en un momento dado.

Riesgo relativo: en estadística y epidemiología, el riesgo relativo es el cociente entre el riesgo en el grupo con el factor de exposición o factor de riesgo, y el riesgo en el grupo de referencia (que no tiene el factor de exposición) como índice de asociación.

Servidor de mapas: servidores que proveen cartografía a través de Internet, tanto en modo vectorial como con imágenes. La especificación estándar para estos servidores es la OGC (Open Geospatial Consortium).

SIG: Sistema de Información Geográfica (GIS²³). Existen múltiples definiciones de este tipo de sistemas, a continuación se enumeran algunas de ellas:

- SIG es un sistema de hardware, software, datos, personas, organizaciones y convenios institucionales para la recopilación, almacenamiento, análisis y distribución de información de territorios de la Tierra (Deuker, Kjerne, 1989).
- Un SIG es una poderosa caja de herramientas para recoger, almacenar, recuperar, transformar y visualizar datos del mundo real (Burrough, 1986).
- Un SIG es un sistema de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión (NCGIA, 1990).

Sistemas de coordenadas²⁴: es una creación artificial que permite la definición analítica de la posición de un objeto o un fenómeno.

²³ Geographic Information System

Sistema de Gestión de Bases de Datos (SGBD): es una colección de programas de aplicación que proporciona al usuario de bases de datos los medios necesarios para realizar tareas como la definición de datos, manipulación, mantenimiento y control de la privacidad y seguridad de los datos.

SSI: es un conjunto de directivas que se escriben en las páginas HTML y que se evalúan en el servidor web cuando se solicita la página HTML. SSI permite añadir contenido generado de forma dinámica a las páginas web, sin tener que programar toda la página mediante CGI, ASP, PHP o alguna tecnología similar

SSL: son protocolos criptográficos que proporcionan comunicaciones seguras por una red, normalmente Internet.

TLS (Transport Layer Security): es un protocolo Internet definido por primera vez en 1999 y actualizado por última vez en 2011. Se basa en las especificaciones previas de SSL (1994, 1995, 1996) desarrolladas por Netscape Communications para agregar el protocolo HTTPS a su navegador Netscape Navigator.

Web Map Service (WMS): son sistemas de consulta de capas de información de forma dinámica desde la web. Permiten la visualización, combinación y consulta puntual de datos de imágenes cartográficas generadas a partir de una o varias fuentes (mapa digital, datos de un SIG, ortofotos, etc) y cargados desde uno o varios servidores.

²⁴ Introducción a los sistemas de información geográfica y geotelemática. Botella, A.; Muñoz, A.; Olivella, R.; et Al. (2011).

8. BIBLIOGRAFIA Y REFERENCIAS

- Botella, A.; Muñoz, A.; Olivella, R.; et Al. (2011). Introducción a los sistemas de información geográfica y geotelemática. Barcelona: Editorial UOC.
- Polo, M. Desarrollo de software basado en la reutilización. Módulo didáctico PID_00184467. UOC.
- Mariné P.; Rodríguez, J. Componentes de la gestión de proyectos: las áreas de conocimiento. Módulo didáctico PID_00153559. UOC.
- Luque, I., Gómez-Nieto, M.A. Diseño y uso de bases de datos relacionales (1997). Madrid. Editorial RA-MA.
- Map Server Open Source Web Mapping: <http://mapserver.org/es/> (visitado: 10-05-2015 a 01-06-2015).
- Geographic Information Systems Stack Exchange : <http://gis.stackexchange.com/> (visitado: 10-05-2015 a 01-06-2015).
- Documentación sobre PostgreSQL: <http://www.postgresql.org/es/documentacion> (visitado: 17-05-2015 a 08-05-2015).
- Spatial and Geographic objects for PostgreSQL: <http://postgis.net>. (visitado: 17-05-2015 a 28-05-2015).
- jQuery API documentation: <http://api.jquery.com/> (visitado: 06-05-2015 a 10-05-2015).
- Leaflet API documentation: <http://leafletjs.com/reference.html> (visitado: 06-05-2015 a 23-05-2015).
- D3 API documentation: <https://github.com/mhstock/d3/wiki> (visitado: 06-05-2015 a 23-05-2015).
- Turf API documentation: <http://turfjs.org/static/docs/> (visitado: 06-05-2015 a 23-05-2015).
- Dimple API documentation: <https://github.com/PMSI-AlignAlytics/dimple/wiki> (visitado: 18-05-2015 a 27-05-2015)
- Wikipedia: <http://es.wikipedia.org/> (visitado: durante todo el trabajo).

9. ANEXOS

9.1. ANEXO I: PLANIFICACIÓN DE TAREAS DEL PLAN DE TRABAJO

EDT	Nombre de la tarea	Duración	Trabajo	Fecha inicio	Fecha fin
Bloque 1	Plan de trabajo	3 días	12 horas	vie 06/03/15	mié 10/03/15
1.1	Elaboración del plan de trabajo	2 días	8 horas	vie 06/03/15	mar 09/03/15
<i>1.2 (hito)</i>	<i>PEC1: Entrega del plan de trabajo</i>	<i>1 día</i>	<i>4 horas</i>	<i>mar 10/03/15</i>	<i>mié 10/03/15</i>
Bloque 2	State of the art - Analizar la tecnología existente	7 días	28 + 7 horas	mié 11/03/15	vie 19/03/15
2.1	Analizar las diferentes soluciones tecnológicas disponibles actualmente para resolver el TFM planteado	3 días	12 horas	mié 11/03/15	lun 13/03/15
2.2	Seleccionar de manera justificada la tecnología a utilizar y describir con detalle las etapas de su implementación	4 días	16 horas	lun 16/03/15	vie 19/03/15
2.3	Redacción de la memoria	7 días	7 horas	mié 11/03/15	vie 19/03/15
Bloque 3	Estructura de la información	18 días	72+27 horas	vie 20/03/15	vie 14/04/15
3.1	Descripción de las diferentes fuentes de datos relativas a los indicadores de salud que se usaran en el proyecto	4 días	16 horas	vie 20/03/15	jue 25/03/15
3.2	Búsqueda de datos cartográficos de tipo open source, como por ejemplo los de OpenSreetMap (OSM), o bien determinar las fuentes cartográficas que se usarán en el proyecto.	4 días	16 horas	jue 26/03/15	mié 31/03/15
3.3	Análisis de la integración de las diferentes fuentes de datos en la tecnología seleccionada para la resolución del TFM	4 días	16 horas	mié 01/04/15	mar 06/04/15
<i>3.4 (hito)</i>	<i>PEC2: Entrega del borrador de la memoria en la que se incluyen los contenidos relacionados con las tareas desarrolladas hasta el momento</i>	<i>1 día</i>	<i>4 horas</i>	<i>mar 07/04/15</i>	<i>mar 07/04/15</i>
3.5	Análisis de la integración de las	5 días	24 horas	mié 08/04/15	vie 14/04/15

	diferentes fuentes de datos en la tecnología seleccionada para la resolución del TFM (continuación)				
3.6	Redacción de la memoria	18 días	27 horas	mié 20/03/15	vie 14/04/15
Bloque 4	Implementación del servidor de mapas	18 días	64 + 16 horas	lun 15/04/15	mar 08/05/15
4.1	Análisis de los servidores de mapas Open Source y selección del más adecuado.	4 días	16 horas	lun 15/04/15	vie 20/04/15
4.2	Implementación del servidor de mapas con la tecnología seleccionada.	10 días	32 horas	lun 21/04/15	mié 04/05/15
4.3	Pruebas de conjunto	4 días	12 horas	jue 05/05/15	mar 08/05/15
4.4	Redacción de la memoria	13 días	16 horas	lun 15/04/15	mar 01/05/15
Bloque 5	Implementación del visor web GIS	25 días?	100 + 21 horas	mié 04/05/15	lun 05/06/15
5.1	Análisis de las diferentes bibliotecas JavaScript que cumplen con los requisitos funcionales del proyecto I	6 días	8 horas	mié 04/05/15	lun 11/05/15
5.2 (hito)	<i>PEC3: Entrega del borrador de la memoria en la que se incluyen los contenidos relacionados con las tareas desarrolladas hasta el momento. Prototipo del servidor de mapas para pruebas y código fuente.</i>	<i>1 días?</i>	<i>4 horas</i>	<i>mar 12/05/15</i>	<i>mar 12/05/15</i>
5.3	Descripción detallada de las funcionalidades que hay que implementar y diseño del portal web.	3 días	12 horas	mié 13/05/15	vie 15/05/15
5.4	Implementación de las funcionalidades en el visor web GIS.	8 días	32 horas	lun 18/05/15	vie 27/05/15
5.5	Pruebas de conjunto	3 días	16 horas	lun 28/05/15	mar 01/06/15
5.6	Elaboración de la presentación virtual	3 días	12 horas	mie 02/06/15	vier 04/06/15
5.7	Redacción de la memoria	21 días	21 horas	mié 04/05/15	mar 04/06/15
5.8 (hito)	<i>PEC4: Entrega final</i>	<i>1 día</i>	<i>4 horas</i>	<i>lun 08/06/15</i>	<i>lun 08/06/15</i>

9.2. ANEXO II: GUIA DE USUARIO

El visor web GIS tiene una interfaz sencilla e intuitiva por lo sólo se describirán las principales funciones del mismo.

- *Inicio de la aplicación:* se debe utilizar un navegador web de última generación e introducir la dirección la URL del sitio web donde se encuentra alojada la aplicación (<http://ubuntu-server/gis/index.html>). A tener en cuenta el nombre del host para pruebas fuera del entorno donde se ha desarrollado el visor, sustituir el nombre del host y ruta relativa por la que corresponda. La cartografía base predeterminada es la de ICGC, el aspecto inicial del visor se muestra en la figura 22.



Figura 22: interfaz inicial del visor GIS, capa base topográfica

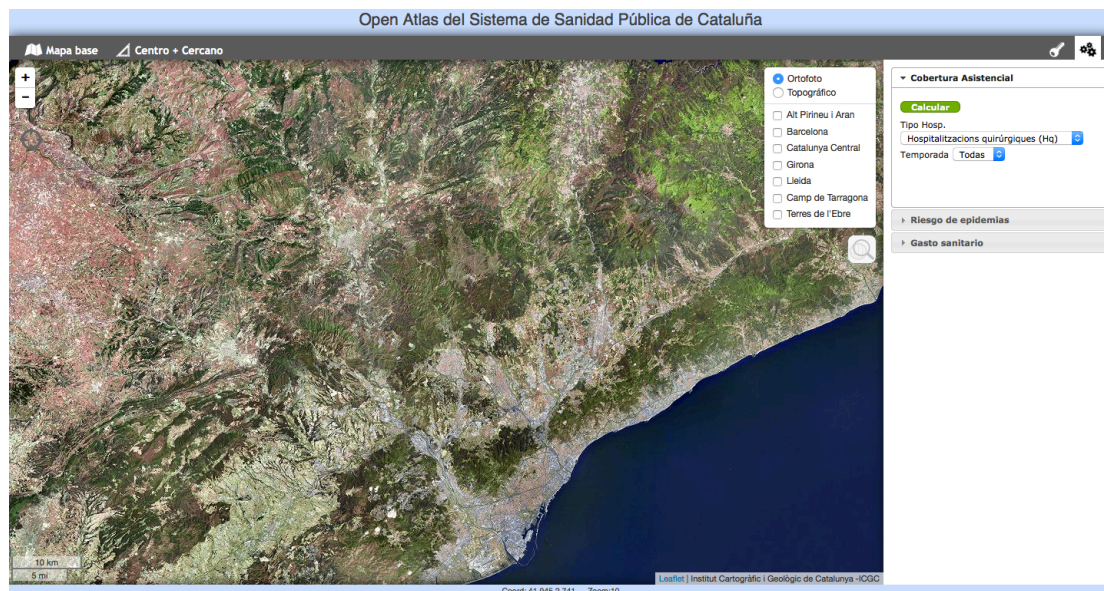


Figura 23: interfaz inicial del visor GIS, capa base ortográfica.

En la barra de estado del visor se muestra las coordenadas geográficas del mapa correspondientes a la posición del puntero del ratón. Además se muestra el nivel de zoom actual.

- **Seleccionar la cartografía base:** seleccione la opción Mapa base de la barra de menú. Después, en el cuadro emergente, seleccione una de las dos opciones: cartografía base del servidor de mapas local o cartografía base del servidor de mapas del ICGC.

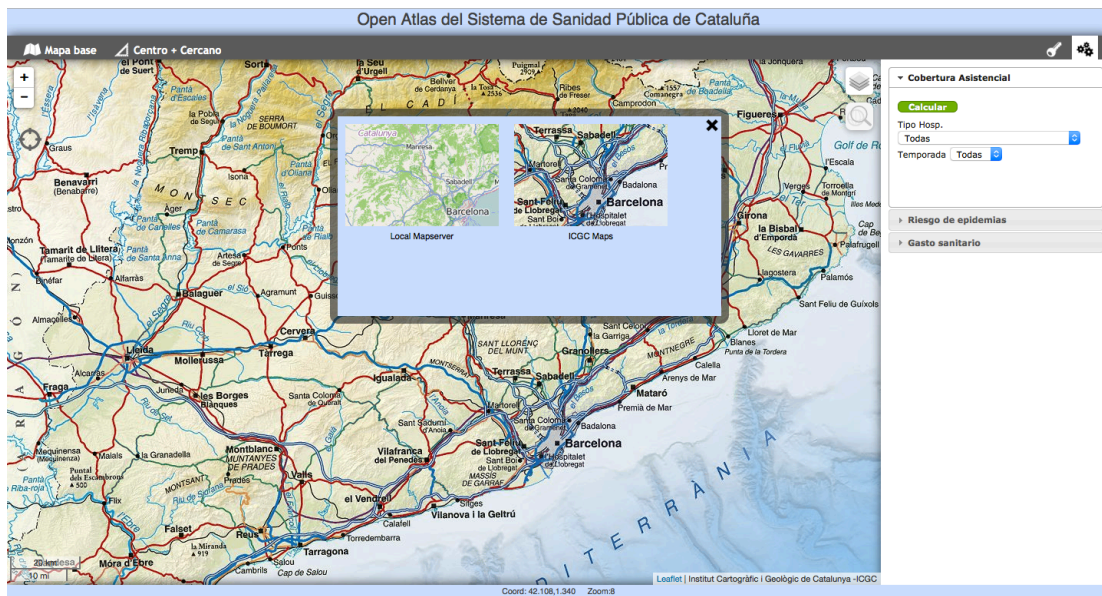


Figura 24: selección del origen de datos geospaciales, servidor de mapas local o servidor de mapas del ICGC.

- **Selección de capas:** desde el control de capas (esquina superior derecha del contenedor del mapa) puede seleccionar la capa base que quiere visualizar y las capas de superposición correspondientes a las regiones sanitarias.



Figura 25: manejo de capas base y capas de regiones sanitarias.

- **Obtención del gasto sanitario:** antes de utilizar los controles de cálculo del gasto sanitario debe seleccionar la región o regiones sanitarias de interés (esta operación siempre debe hacerla antes de iniciar cualquier cálculo). En caso contrario se mostrará un cuadro emergente avisándole.

Open Atlas del Sistema de Sanidad Pública de Cataluña

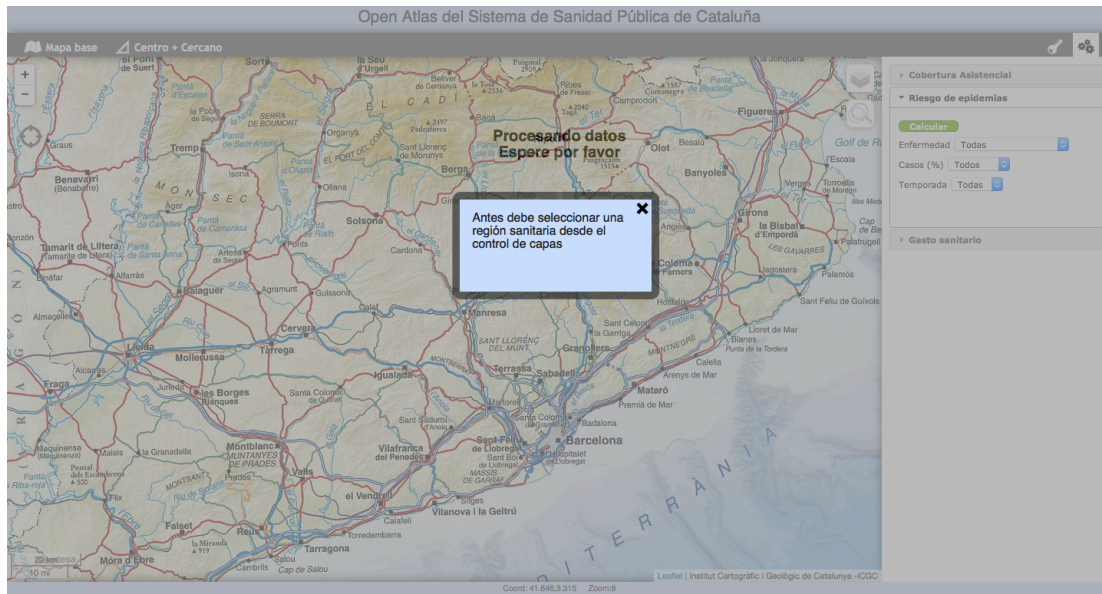


Figura 26: información de usuario para que seleccione una región sanitaria

En la figura se puede ver el resultado del gasto sanitario en la región sanitaria de Barcelona. En el mapa aparece un marcador, centrado en la región, con una etiqueta con la información correspondiente al gasto sanitario. Al lado del nombre de la región sanitaria encontrará un icono desde donde podrá visualizar gráficos estadísticos de los datos obtenidos.



Figura 27: ejemplo de obtención de gasto sanitario para la región sanitaria de Barcelona.

En la figura 28 se muestra un ejemplo de consulta de gasto sanitario utilizando como campo de filtro la temporada (2013).

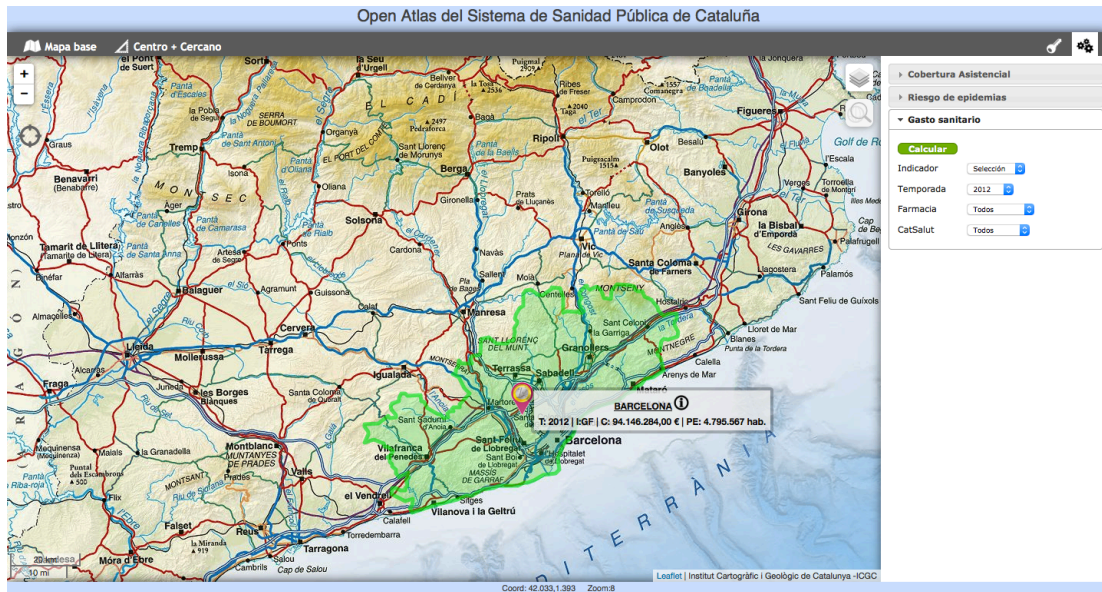


Figura 28: ejemplo de obtención de gasto sanitario para la región sanitaria de Barcelona en el año 2013

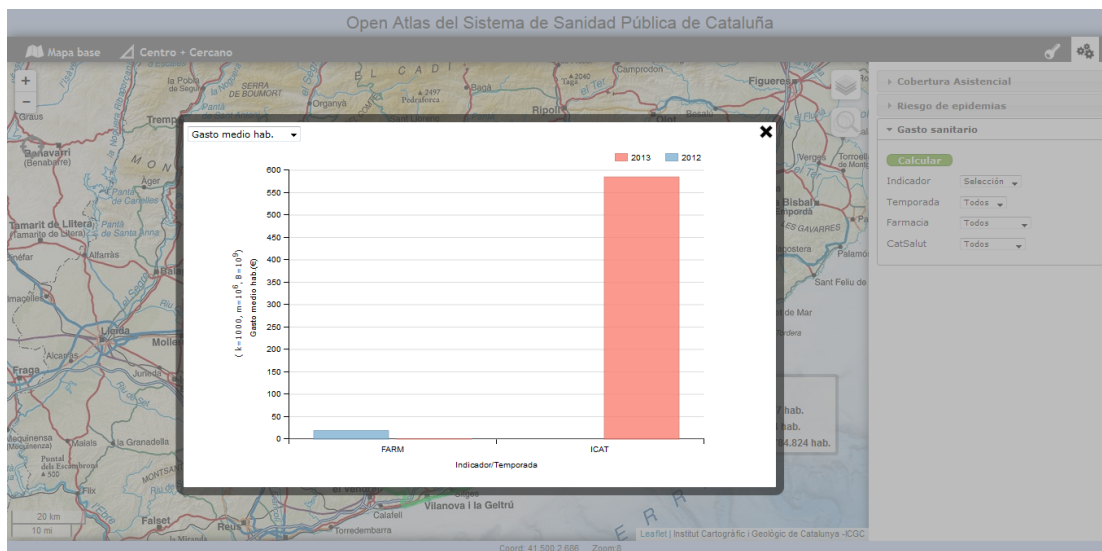


Figura 29: visualización de gráficos estadísticos, gasto medio por habitante y año

- Obtención del riesgo de epidemias: una vez seleccionada la región sanitaria de elegir el tipo de enfermedad contagiosa sobre la que quiere calcular el riesgo de epidemia. Si no lo hace se mostrará un cuadro de diálogo informándole.

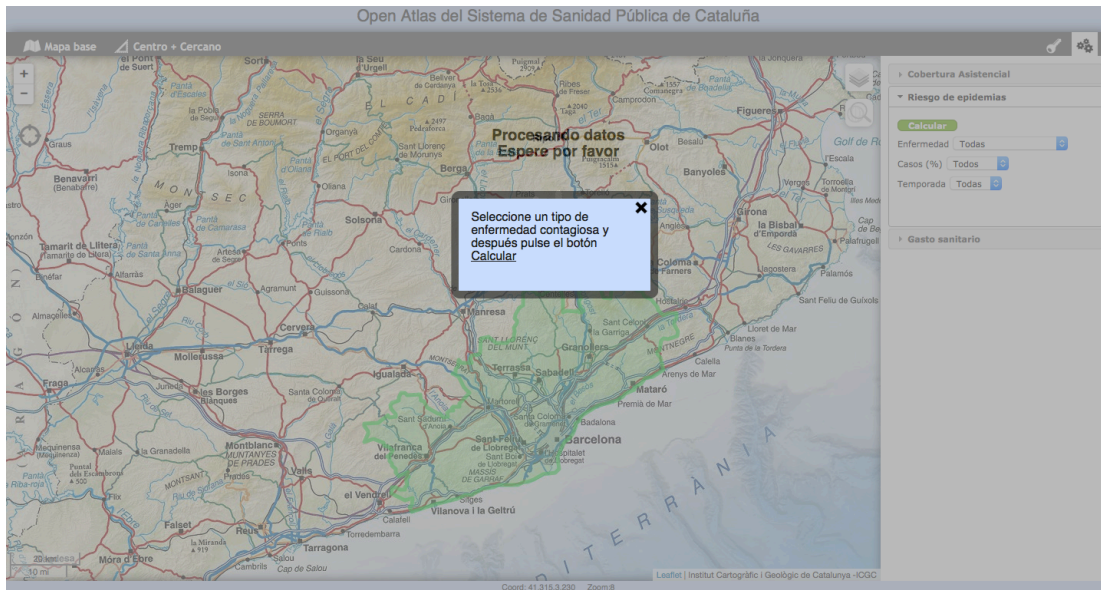


Figura 30: información de usuario para que seleccione un tipo de enfermedad contagiosa.

El riesgo de epidemia se calcula en relación a la región sanitaria elegida y los campos de filtro. Los resultados se muestran en un marcador centrado en la región sanitaria. Al lado se puede ver una etiqueta con un resumen de los cálculos resultantes (temporada, porcentaje de casos de la enfermedad en la región sanitaria y población estimada). En la misma etiqueta, a la derecha del nombre de la región sanitaria dispone de un icono para acceder a los gráficos estadísticos de los datos.

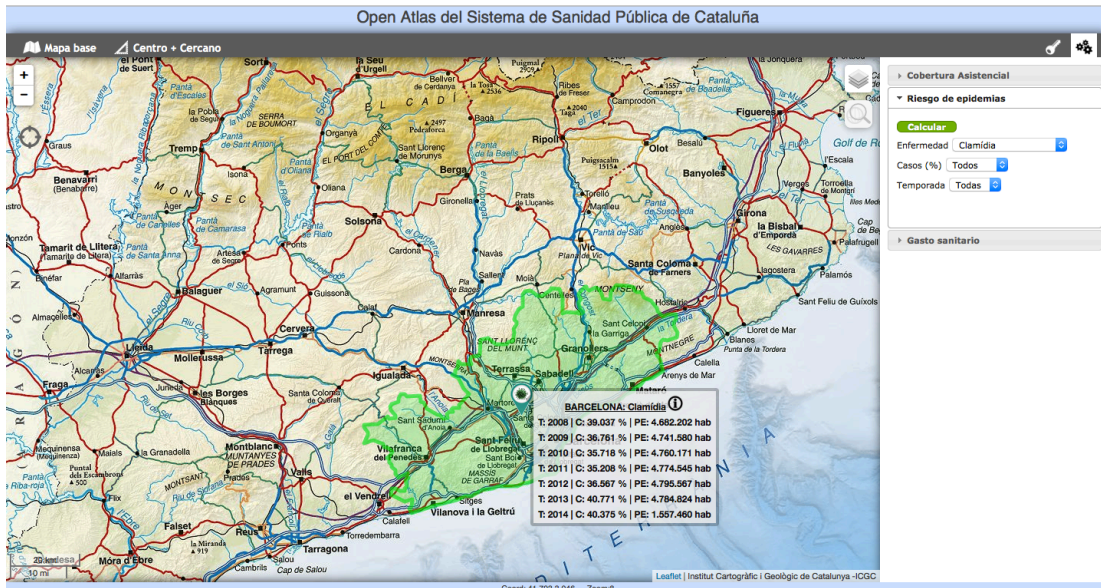


Figura 31: cálculo del riesgo de epidemia para 'la Clamidia' en la región sanitaria de Barcelona.



Figura 32: cálculo del riesgo de epidemia para la 'Clamidia' en la región sanitaria de Barcelona, año 2014.

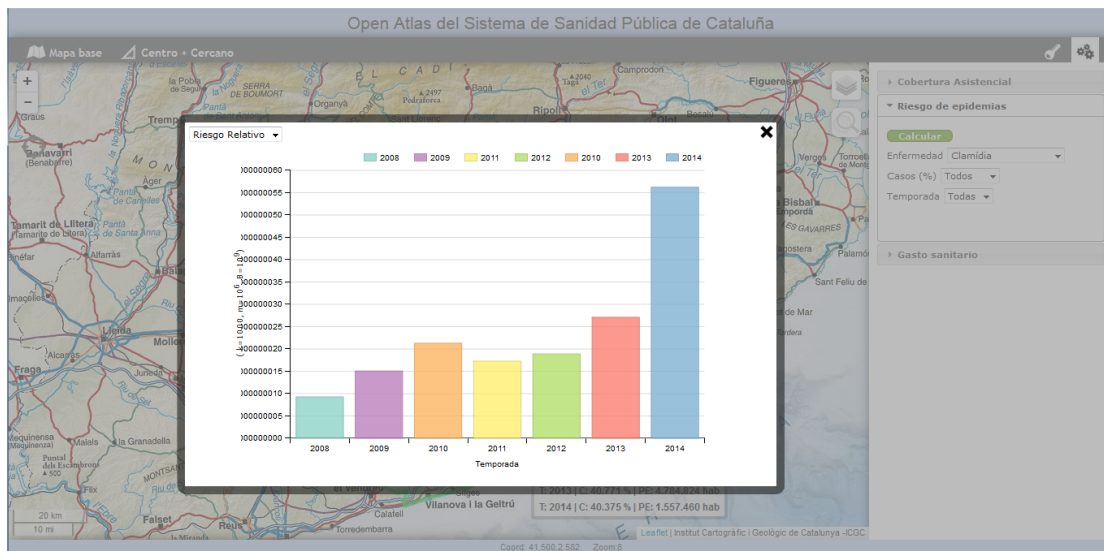


Figura 33: gráficos estadísticos sobre el riesgo relativo de padecer una epidemia de 'Clamidia' en la región sanitaria de Barcelona.

- Cobertura asistencial:** a diferencia de los cálculos anteriores, en el cálculo de la cobertura asistencial se muestran distintos marcadores geográficos, uno por cada centro de atención hospitalaria. Además se dibuja sobre el mapa una capa con las celdas de Voronoi que representan el área de influencia de cada centro hospitalario. Al lado de cada marcador se muestra una etiqueta con los datos resultantes del cálculo de cobertura (número de cada tipo de hospitalización, porcentaje de cada tipo de hospitalización en Cataluña, en la región sanitaria elegida y en el área de influencia de cada centro hospitalario). Debajo del nombre de la región sanitaria se muestra la población estimada del área de influencia del centro hospitalario. A la derecha del nombre de la región sanitaria encontrará un icono desde el que puede acceder a los gráficos estadísticos de los datos.

Open Atlas del Sistema de Sanidad Pública de Cataluña

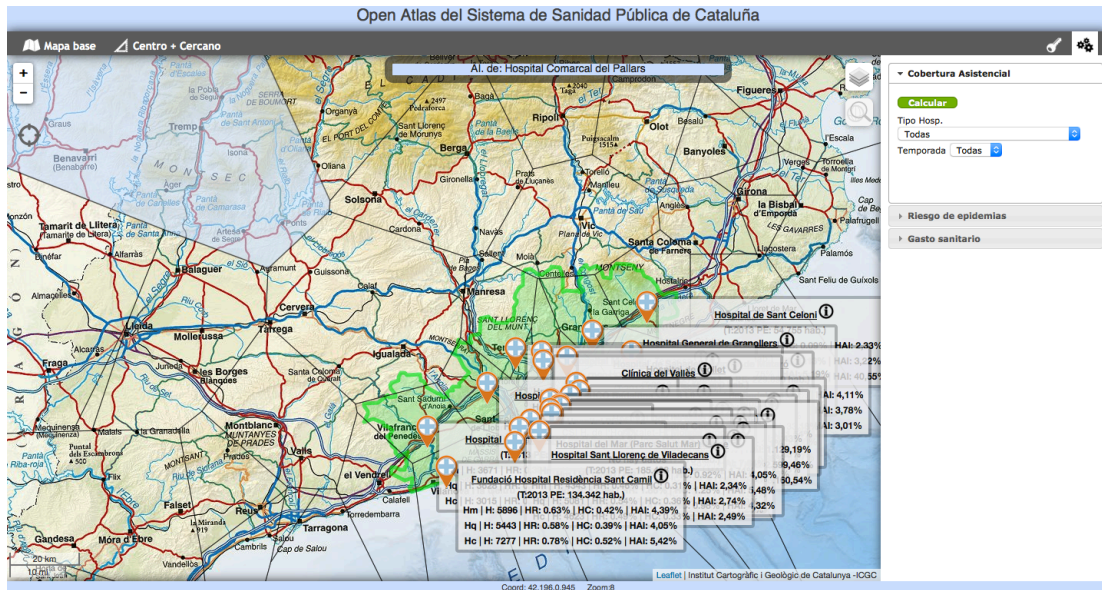


Figura 34: cálculo de la cobertura asistencial en la región sanitaria de Barcelona.

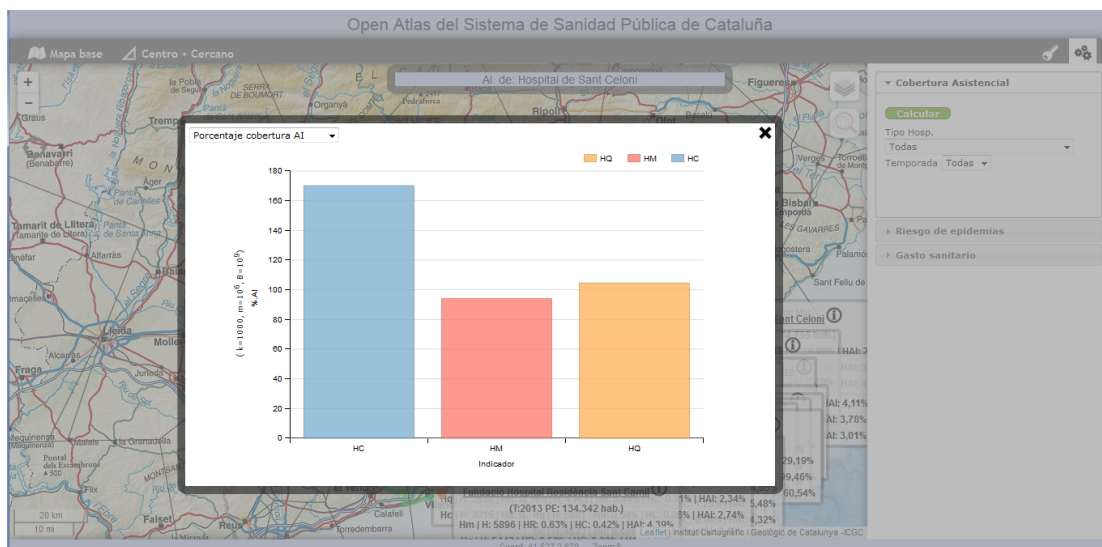


Figura 35: gráficos estadísticos sobre la cobertura asistencial por tipo de hospitalización en la región sanitaria de Barcelona.

- Geolocalización: el visor dispone de un buscador de localizaciones geográficas basado en el nombre lugares conocidos. Como resultado de la búsqueda, si se han encontrado resultados, se amplía y centra el mapa en la localización geográfica del lugar buscado.



Figura 36: ejemplo de geolocalización, búsqueda de la ciudad de 'Mataró'



Figura 37: ejemplo de geolocalización, resultado de la búsqueda de la ciudad de 'Mataró'

- Localización del centro de atención hospitalaria más cercano:** el visor dispone de una función de búsqueda del centro hospitalario más cercano. Para utilizar esta función debe seleccionar un punto en el mapa (click derecho del ratón) y después seleccionar la opción 'Centro+Cercano' del menú de opciones. Si hay resultados se mostrarán dos marcadores en el mapa, uno con la posición geográfica seleccionada y otro con la localización geográfica del centro hospitalario más cercano. En este último marcado se mostrará un globo emergente con el nombre del centro hospitalario, la distancia y el tiempo estimado para llegar. Además en la parte derecha del contenedor del mapa se muestra el itinerario a seguir para llegar al centro hospitalario.



Figura 38: búsqueda del centro hospitalario más cercano a localización geográfica elegida por el usuario.

- Información sobre las leyendas del mapa: puede consultar la descripción de leyendas del mapa.



Figura 39: información sobre las leyendas del mapa