



## **Open Atles del Sistema de Sanitat Pública de Catalunya**

---

**Treball de Fi de Grau – Àrea de Sistemes d'Informació Geogràfica  
Grau d'Enginyeria Informàtica**

Gabriel Garcia-Mascaraque Gil  
Consultora: Anna Muñoz Bolas

Curs 2014-2015. Segon semestre.



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FITXA DEL TREBALL FINAL

---

<b>Títol del treball:</b>	Open Atlas del Sistema de Sanitat Pública de Catalunya
<b>Nom de l'autor:</b>	Gabriel Garcia-Mascaraque Gil
<b>Nom del consultor:</b>	Anna Muñoz Bolas
<b>Data de lliurament (mm/aaaa):</b>	06/2015
<b>Àrea del Treball Final:</b>	Sistemes d'Informació Geogràfica
<b>Titulació:</b>	Grau d'Enginyeria Informàtica
<b>Resum del Treball:</b>	
<p>El present projecte consisteix en la creació d'un atlas sanitari de Catalunya. Aquest atlas consisteix en una aplicació web formada per un visor de mapes que permet la visualització de diferents capes d'informació relacionades amb l'àmbit sanitari públic de Catalunya, com ara la qualitat de la cobertura sanitària.</p> <p>El projecte s'estructura en dues grans fases: l'estudi i la implementació.</p> <p>En la primera es realitza un anàlisi de la tecnologia existent, es determina quines solucions són les més adients per a aquesta implementació i s'analitzen els pros i contres de les alternatives.</p> <p>En la fase d'implementació es desenvolupa l'aplicació web com a tal, aplicant les decisions tecnològiques a les que s'arriba a la primera fase, i s'instal·len i parametritzen els serveis necessaris a la capa de servidor, els quals són: un servidor web per a servir les pàgines que formen el visor, i un servidor de mapes per a servir les capes d'informació.</p> <p>Paral·lelament a aquestes dues fases es realitza un estudi de dades obertes de l'àmbit sanitari i se'n fa un tractament per a que puguin ser utilitzades amb eines SIG.</p>	

**Abstract:**

This project comprises the creation of a healthcare atlas about Catalonia. This atlas is a web application consisting of a map viewer which allows the visualisation of different layers of information related to the public health sector in Catalonia, such as quality health coverage.

The project is divided into two phases: study and development.

The first is an analysis of existing technology, which also determines which solutions are most appropriate for this implementation and discusses the pros and cons of other alternatives.

The implementation stage includes the development of the web application applying the technology researched in the first stage, and also the installation and configuration of the necessary services in the server layer, which are: a Web server to serve pages that make up the viewer, and a map server to serve the layers of information.

Parallel to these two phases there's also an study on open data about the public healthcare system. These datasets are then treated in a way so that they can be used with GIS tools.

**Paraules clau:**

SIG, GIS, aplicacions web, sanitat, cartografia.

# Índex

---

1	Introducció .....	3
1.1	Context i justificació del Treball.....	3
1.2	Objectius del Treball .....	3
1.3	Enfocament i mètode seguit.....	4
1.4	Planificació del Treball .....	4
1.5	Anàlisi de riscos.....	10
1.6	Breu sumari de productes obtinguts .....	12
1.7	Breu descripció dels altres capítols de la memòria .....	12
2	Introducció als Sistemes d'Informació Geogràfica.....	13
2.1	Què és un SIG? .....	13
2.2	Tipus de dades en SIG .....	14
2.3	Coordenades i projeccions geogràfiques.....	16
3	Tecnologia existent .....	19
3.1	Eines SIG d'escriptori .....	19
3.2	Bases de dades espacials .....	19
3.3	Servidors de mapes.....	21
3.4	Llibreries JavaScript per a la implementació de visors web de mapes. ....	21
3.5	Altres llibreries JavaScript de propòsit general .....	22
3.6	Altres utilitats.....	23
4	Estudi i tractament de les dades .....	24
4.1	Dades cartogràfiques .....	24
4.2	Capa de suport per a les dades sanitàries .....	27
4.3	Dades sanitàries .....	29
5	Implementació del servidor de mapes.....	33
5.1	Sol·licituds WMS .....	37
5.2	Sol·licituds WFS.....	39
6	Disseny del visor web .....	41
6.1	Decisions sobre l'entorn de treball en el desenvolupament del producte ....	41
6.2	Anàlisi de requisits .....	43
6.3	Arquitectura.....	44
7	Implementació del visor web .....	47
7.1	Estructura de la solució.....	47
7.2	La Vista – Càrrega de capes .....	53
7.3	La Vista – Enriquant el contingut .....	57
8	Implementacions futures .....	61
9	Conclusions.....	63
10	Glossari .....	64
11	Bibliografia.....	66
12	Annexos .....	67

## Llista de figures

Fig. 1 – EDT del projecte .....	6
Fig. 2 – Diagrama Gantt del projecte.....	9
Fig. 3 – Mapa de l'estudi realitzat pel doctor John Snow .....	13
Fig. 4 – Exemple de representació espacial amb el model ràster .....	14
Fig. 5 – Exemple de mapa de calor (heatmap) .....	15
Fig. 6 – Càlcul de coordenades en un sistema de coordenades geogràfiques.....	16
Fig. 7 - Càlcul de coordenades en un sistema de coordenades cartesianes .....	17
Fig. 8 – Mapamundi fent servir la projecció de Mercator .....	18
Fig. 9 – Logotip del projecte Leaflet .....	22
Fig. 10 – Logotip del projecte OpenLayers .....	22
Fig. 11 – Esquema de l'estructura de capes del visor web.....	24
Fig. 12 – Mapa obtingut del propi visor web del projecte OSM .....	26
Fig. 13 – Mapa obtingut del visor web de l'ICGC.....	26
Fig. 14 – Regions sanitàries de Catalunya .....	27
Fig. 15 – “Merge” d'objectes vectorials per a la creació de la capa de suport de Regions Sanitàries .....	28
Fig. 16 – Regions sanitàries amb nombre d'assegurats .....	30
Fig. 17 – Capa de centres sanitaris .....	31
Fig. 18 – Capa de donacions de sang per cada 1000 habitants assegurats (any 2013) .	32
Fig. 19 – Accés al panell de configuració del servidor GeoServer .....	33
Fig. 20 – Configuració de l'espai de treball a GeoServer.....	34
Fig. 21 – Magatzems de dades utilitzats a GeoServer .....	34
Fig. 22 – Reprojectat de sistema de referència d'una capa a GeoServer .....	35
Fig. 23 – Parametrització de l'extensió d'una capa a GeoServer .....	35
Fig. 24 – Exemple d'estil SLD per a punts .....	36
Fig. 25 – Capa de regions sanitàries amb estil SLD a GeoServer .....	37
Fig. 26 – Capa de regions sanitàries obtinguda amb WMS de GeoServer .....	38
Fig. 27 – Arquitectura de l'aplicació .....	44
Fig. 28 – Estructura general de la solució/projecte.....	47
Fig. 29 – Estructura de fitxers de la vista.....	51
Fig. 30 – Vista Centres .....	54
Fig. 31 – Vista Centres – Elements que es defineixen a la vista .....	54
Fig. 32 – Capa de centres carregada per primera vegada .....	57
Fig. 33 – Finestres emergents amb Bootstrap.....	58
Fig. 34 – Cercador d'elements de la capa.....	59
Fig. 35 – Llegenda d'una capa amb atributs categòrics .....	60
Fig. 36 – Llegenda d'una capa amb atributs numèrics .....	60
Fig. 37 – Gràfica dinàmica en D3 .....	60

# 1 Introducció

---

## 1.1 Context i justificació del Treball

En plena expansió de l'era tecnològica en la que ens trobem el volum i la demanda de dades que generem i consumim augmenta de forma exponencial amb cada dia que passa. Aquestes dades s'emmagatzemen i cal que posteriorment s'analitzin per treure'n algun profit.

Un d'aquests casos és el del Departament de Salut de la Generalitat de Catalunya, que publica de forma regular una gran quantitat de dades obertes que poden ser consultades pels ciutadans per a conèixer millor diversos aspectes de la sanitat pública. Aquestes dades, però, solen ser inconnexes entre elles i difícils d'analitzar.

En aquest anàlisi les components visual i geogràfica poden jugar un paper molt important i aquí és on entren en joc els SIG. Els SIG o Sistemes d'Informació Geogràfica són eines que permeten la representació i anàlisi de grans volums de dades que disposin de component geogràfica.

Amb l'ús de tecnologies SIG es pot ajudar al ciutadà a entendre millor aquestes dades ja que la visualització damunt d'un mapa permet apropar les dades a un món més real.

## 1.2 Objectius del Treball

L'objectiu principal d'aquest projecte és la realització d'un visor web que, mitjançant tecnologies SIG, representi una selecció de dades obertes de l'àmbit sanitari públic de Catalunya.

Amb el desenvolupament d'aquest projecte s'assoliran els següents objectius generals i específics:

### 1.2.1 Objectius Generals:

- Implementar un servidor de mapes.
- Implementar un visor SIG web.
- Integrar dades de diverses fonts sobre una plataforma SIG.

### 1.2.2 Objectius Específics:

- Conèixer a fons un dels servidor de mapes Open Source disponibles (MapServer, GeoServer o altres).
- Estudiar i conèixer les diverses llibreries JavaScript de codi lliure que permeten implementar un visor GIS web.
- Mostrar informació de diferents fonts organitzada en capes.
- Analitzar la informació que es representi a l'atles i treure'n conclusions estadístiques.

## 1.3 Enfocament i mètode seguit

Aquest projecte està emmarcat en una àrea d'estudi poc present en la titulació però, en canvi, molt present en el món del desenvolupament d'aplicacions.

És per aquest motiu que, des d'un inici, s'ha plantejat com un projecte amb una forta càrrega d'aprenentatge i recerca on la consulta de projectes similars fonamenten la major part del temps invertit.

La segona part del projecte, centrada en el desenvolupament d'aplicacions web, és un àmbit molt més tractat als estudis i amb patrons de treball molt ben definits.

## 1.4 Planificació del Treball

El desenvolupament d'aquest projecte s'ha estructurat en quatre blocs que es descriuen a continuació. Aquests blocs són un punt de partida i no es corresponen a l'entrega de les PAC, que es descriuen més endavant.

### **Bloc 01: State-of-the-art. Anàlisi de la tecnologia existent**

- 01.1: Recerca de les diferents solucions tecnològiques disponibles actualment per a resoldre el TFG plantejat.
- 01.2: Selecció justificada de la tecnologia a utilitzar i descripció detallada de les etapes de la seva implementació.

### **Bloc 02: Estructuració de la informació**

- 02.1: Descripció de les diferents fonts de dades relatives als indicadors de salut que es faran servir en el projecte.



- 02.2: Recerca de dades cartogràfiques de tipus Open Source, com per exemple les de OpenStreetMap (OSM), o bé determinació de les fonts cartogràfiques que es faran servir en el projecte.
- 02.3: Anàlisi de la integració de les diferents fonts de dades en la tecnologia seleccionada per a la resolució del TFG.

### **Bloc 03: Implementació del servidor de mapes**

- 03.1: Anàlisi dels servidors de mapes Open Source i tria del més adient.
- 03.2: Implementació del servidor de mapes amb la tecnologia seleccionada.

### **Bloc 04: Implementació del visor web GIS**

- 04.1: Anàlisi de les diferents biblioteques i *frameworks* JavaScript que compleixin els requeriments funcionals del projecte.
- 04.2: Descripció detallada de les funcionalitats que cal implementar i disseny del portal web.
- 04.3: Implementació de les funcionalitats en el visor web GIS.

#### 1.4.1 Estructura de Descomposició del Treball (EDT)

S'ha optat per una estructura de descomposició del treball en dos grans grups que alhora s'han subdividit en quatre blocs principals de tasques (definites a l'apartat 1.4).

D'una banda s'han agrupat totes aquelles tasques relacionades amb l'estudi necessari per a dur a terme el projecte i tota la tasca de documentació del projecte en el grup "Estudi i Documentació".

En segon lloc es defineix el grup "Implementació" per a englobar totes aquelles tasques pròpies de la producció de software: instal·lació, parametrització i programació.

En darrer lloc hi ha un grup "Presentació" per a tenir separades les tasques d'elaboració de la presentació final del projecte i la memòria d'aquest.

Es mostra a continuació (Il·lustració 1) un esquema EDT simplificat del projecte:

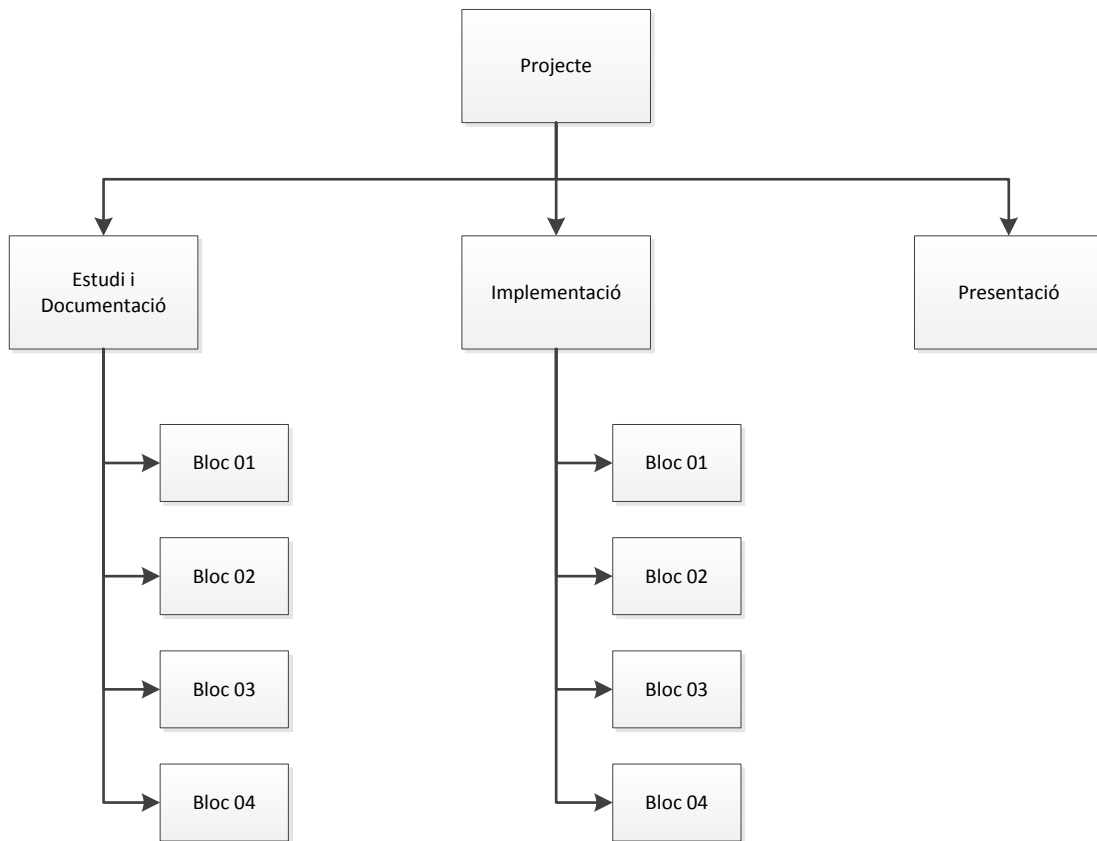


Fig. 1 – EDT del projecte

#### 1.4.2 Fites clau del projecte (PAC)

S’ha establert un calendari de fites basat en el calendari d’entrega de les PAC, tal i com es mostra a la següent taula:

Fita	Contingut
<b>PAC 1</b>	Pla de Treball
<b>PAC 2</b>	Estudi i tractament de dades
<b>PAC 3</b>	Estudi de tecnologia i inici del desenvolupament
<b>PAC 4</b>	Implementació final i documentació

### 1.4.3 Calendari del projecte

En aquest apartat es presenta una descomposició més a fons de cada tasca individual que compon el projecte. Es mostra, en primer lloc, el llistat de tasques a realitzar i tot seguit el diagrama Gantt del projecte:

Tasca	Durada	Inici	Fi
<b>Estudi i Documentació</b>	56 dies	jue 26/02/15	dom 10/05/15
<b>Planificació</b>	8,25 dies	jue 26/02/15	dom 08/03/15
Elaboració del Pla de Treball	5 dies	jue 26/02/15	mié 04/03/15
Revisió del Pla de Treball	2 dies	sáb 07/03/15	dom 08/03/15
<b>Bloc 01</b>	10 dies	mar 10/03/15	dom 22/03/15
Estudi de la tecnologia existent	5 dies	mar 10/03/15	lun 16/03/15
Presca de decisió: Tecnologia a emprar	0 dies	lun 16/03/15	lun 16/03/15
Documentació Bloc 01 a la memòria del projecte	5 dies	lun 16/03/15	dom 22/03/15
<b>Bloc 02</b>	10 dies	lun 23/03/15	dom 05/04/15
Estudi de les fonts de dades cartogràfiques existents	2 dies	lun 23/03/15	jue 26/03/15
Estudi dels indicadors de salut que es faran servir	5 dies	vie 27/03/15	jue 02/04/15
Documentació Bloc 02 a la memòria del projecte	5 dies	vie 27/03/15	jue 02/04/15
Correccions per entrega PAC 2	3 dies	jue 02/04/15	dom 05/04/15
<b>Bloc 03</b>	5 dies	jue 02/04/15	mié 08/04/15
Estudi dels diferents servidors de mapes disponibles	2 dies	jue 02/04/15	sáb 04/04/15
Presca de decisió: Servidor de mapes a emprar	0 dies	sáb 04/04/15	sáb 04/04/15
Documentació Bloc 03 a la memòria del projecte	5 dies	jue 02/04/15	mié 08/04/15
<b>Bloc 04</b>	24,5 dies	jue 09/04/15	dom 10/05/15
Estudi de les biblioteques JavaScript disponibles per a la implementació del visor web	5 dies	jue 09/04/15	mié 15/04/15
Disseny conceptual del portal web	7 dies	jue 09/04/15	sáb 18/04/15
Documentació Bloc 04 a la memòria del projecte	15 dies	mié 15/04/15	lun 04/05/15
Correccions per entrega PAC 3	4 dies	mié 06/05/15	dom 10/05/15
<b>Implementació</b>	46,5 dies	mar 10/03/15	dom 10/05/15
<b>Bloc 01</b>	1 dia	mar 10/03/15	mié 11/03/15
Instal·lació del software GIS necessari	1 dia	mar 10/03/15	mié 11/03/15
<b>Bloc 02</b>	7 dies	lun 23/03/15	jue 02/04/15
Tractament previ de les dades	7 dies	lun 23/03/15	jue 02/04/15
<b>Bloc 03</b>	10 dies	lun 06/04/15	dom 19/04/15
Implementació del servidor de mapes	10 dies	lun 06/04/15	dom 19/04/15
<b>Bloc 04</b>	16 dies	dom 19/04/15	dom 10/05/15
Implementació del visor web	16 dies	dom 19/04/15	dom 10/05/15
Versió preliminar	12 dies	dom 19/04/15	mar 05/05/15
Versió final	4 dies	mié 06/05/15	dom 10/05/15
<b>Presentació</b>	18,5 dies	mar 12/05/15	sáb 06/06/15
Compilació final de la memòria del projecte	10 dies	mar 12/05/15	dom 24/05/15
Tasca	Durada	Inici	Fi
Elaboració de la presentació virtual	5 dies	mar 12/05/15	lun 18/05/15
Modificació de la presentació virtual	7 dies	jue 28/05/15	sáb 06/06/15
Modificació de la memòria final del projecte	7 dies	jue 28/05/15	sáb 06/06/15
Debat virtual	0 dies	mar 23/06/15	mar 23/06/15
Debat virtual	0 dies	mar 23/06/15	mar 23/06/15
<b>Fites</b>	71 dies	sáb 07/03/15	lun 08/06/15
<b>PAC1</b>	3,25 dies	sáb 07/03/15	mar 10/03/15
Entrega prèvia	0 dies	sáb 07/03/15	sáb 07/03/15
Entrega definitiva	0 dies	mar 10/03/15	mar 10/03/15
<b>PAC2</b>	3,5 dies	jue 02/04/15	dom 05/04/15
Entrega prèvia	0 dies	jue 02/04/15	jue 02/04/15
Entrega definitiva	0 dies	dom 05/04/15	dom 05/04/15

<b>PAC3</b>	4,75 dies	mié 06/05/15	mar 12/05/15
<b>Entrega prèvia</b>	0 dies	mié 06/05/15	mié 06/05/15
<b>Entrega definitiva</b>	0 dies	mar 12/05/15	mar 12/05/15
<b>PAC4</b>	9 dies	jue 28/05/15	lun 08/06/15
<b>Entrega prèvia</b>	0 dies	jue 28/05/15	jue 28/05/15
<b>Entrega definitiva</b>	0 dies	lun 08/06/15	lun 08/06/15



## 1.5 Anàlisi de riscos

En el desenvolupament de qualsevol projecte poden aparèixer problemes o inconvenients que poden afectar el calendari establert. L'anàlisi de riscos intenta minimitzar les conseqüències anticipant-se a l'aparició d'aquests.

Aquest anàlisi consta de les següents fases:

- Identificació dels riscos
- Anàlisi qualitatiu dels riscos
- Anàlisi quantitatiu dels riscos
- Planificació de la resposta als riscos
- Monitorització dels riscos

### 1.5.1 Identificació dels riscos

S'han detectat alguns riscos que poden afectar a l'execució del projecte. Aquests riscos, si s'arriben a produir, poden afectar la durada d'alguna fase del projecte i, potser, a la durada total del projecte si aquesta fase era crítica. S'enumeren a continuació:

#### **Riscos**

<i>Viatge laboral</i>	Existeix la probabilitat d'haver de viatjar per motius laborals. Habitualment aquests viatges no excedeixen més de dos dies entre setmana però convertiria aquests dies en inhàbils.
<i>Puntes de feina a la universitat</i>	Degut a la realització d'altres assignatures és molt probable que hi hagi puntes de feina per a entregar activitats en dates a priori conegudes.
<i>Avaria al maquinari: Pèrdua de dades</i>	Pot produir-se una avaria greu al maquinari (discs durs) utilitzat que provoqui la pèrdua de les dades existents fins a la data.
<i>Avaria al maquinari: Altres</i>	Pot produir-se una avaria greu al maquinari que impedeixi l'ús de la màquina. Aquí s'inclouen altres components com ara una font d'alimentació, el processador o algun perifèric.
<i>Mala planificació</i>	A mesura que avanci el projecte pot fer-se evident que la planificació era errònia i s'havia subestimat.

### 1.5.2 Anàlisi qualitatiu dels riscos

En aquest anàlisi s'estudia la probabilitat que es produeixin aquests riscos i l'impacte que tindrien en el cas que així sigui.

<b>Riscos</b>	<b>Probabilitat</b>	<b>Impacte</b>
<i>Viatge laboral</i>	Baixa	Baix
<i>Puntes de feina a la universitat</i>	Alta	Mig
<i>Avaria al maquinari: Pèrdua de dades</i>	Baixa	Alt
<i>Avaria al maquinari: Altres</i>	Baixa	Baix
<i>Mala planificació</i>	Mitja	Alt

### 1.5.3 Anàlisi quantitatiu dels riscos

En aquest anàlisi s'avaluen els costos que es tindrien en cas que els riscos es converteixin en problemes existents. Degut a la naturalesa del projecte, un projecte de fi d'estudis universitaris, el cost de caure en els riscos exposats serà, en la majoria de casos, temporal. Aquest cost temporal es traduirà, en cas de produir-se el problema, en un endarreriment en l'execució de les tasques en procés d'aquell moment.

<b>Riscos</b>	<b>Cost temporal</b>	<b>Cost econòmic</b>
<i>Viatge laboral</i>	1 o 2 dies	-
<i>Puntes de feina a la universitat</i>	Entre 2 i 5 dies	-
<i>Avaria al maquinari: Pèrdua de dades</i>	Variable. Podria fins i tot cancel·lar el projecte.	60 €
<i>Avaria al maquinari: Altres</i>	1 dia	50 – 300 €
<i>Mala planificació</i>	Variable. Podria fins i tot cancel·lar el projecte.	-

### 1.5.4 Planificació de la resposta als riscos

Per fer front als riscos existents s'han estudiat possibles solucions per a minimitzar-ne la probabilitat i l'impacte dels mateixos.

<b>Riscos</b>	<b>Acció correctora</b>	<b>Probabilitat final / Impacte final</b>
<i>Viatge laboral</i>	No n'hi haurà. Si s'arriba a produir no es podrà evitar.	Baixa / Baix
<i>Puntes de feina a la universitat</i>	Dedicar més esforç temporal.	Alta / Baix
<i>Avaria al maquinari: Pèrdua de dades</i>	Es faran còpies de seguretat periòdiques en un mitjà extern.	Baixa / Nul
<i>Avaria al maquinari: Altres</i>	No es pot minimitzar el risc.	Baixa / Baix
<i>Mala planificació</i>	Dedicar més esforç temporal.	Mitja / Mig

## **1.6 Breu sumari de productes obtinguts**

Aquest projecte té com a resultat una aplicació web formada per un visor de mapes amb capes espacials que representen dades de tipus ràster i vectorials. El visor permet a l'usuari la interacció amb el mapa amb les funcionalitats típiques d'aquest tipus de producte (navegació, zoom, etc.) a més de la interacció amb les pròpies dades, que es presenten en forma gràfica de diverses maneres.

La implementació fa ús de bases de dades espacials, fitxers d'informació espacial i un servidor de mapes.

## **1.7 Breu descripció dels altres capítols de la memòria**

En els següents capítols s'aprofundeix en les diferents fases del projecte.

Es comença amb una visió general de l'àmbit tecnològic dels SIG, fent una mica d'història així com un recull de nocions sobre geodèsia i cartografia. Continua amb un repàs a solucions tecnològiques existents i amb la presa de decisions que s'ha dut a terme per a implementar la solució d'aquest projecte. Per últim es detallen els processos de parametrització i implementació de la solució final.



## 2 Introducció als Sistemes d'Informació Geogràfica

### 2.1 Què és un SIG?

Hi ha moltes maneres de definir què és exactament un SIG (o GIS, en anglès i com és coneix més sovint). Es pot començar per traduir l'acrònim. Un SIG és un *Sistema d'Informació Geogràfica* o *Geographic Information System* en anglès; és a dir, és un sistema que permet l'estudi d'informació sobre dades geogràfiques.

En cap moment s'ha fet referència a la computació ja que aquest camp d'estudi és anterior a l'aparició dels primers computadors. De fet, es sòl fer referència a un famós estudi sobre l'epidèmia de còlera que va colpejar Londres al 1840 dut a terme pel doctor **John Snow**. El doctor Snow va recollir sobre plànols els domicilis dels afectats i morts pel còlera i això va servir per demostrar que l'origen del brot era una font pública situada al carrer Broad. Va ser el primer científic en fer servir plànols en un estudi d'aquestes característiques.

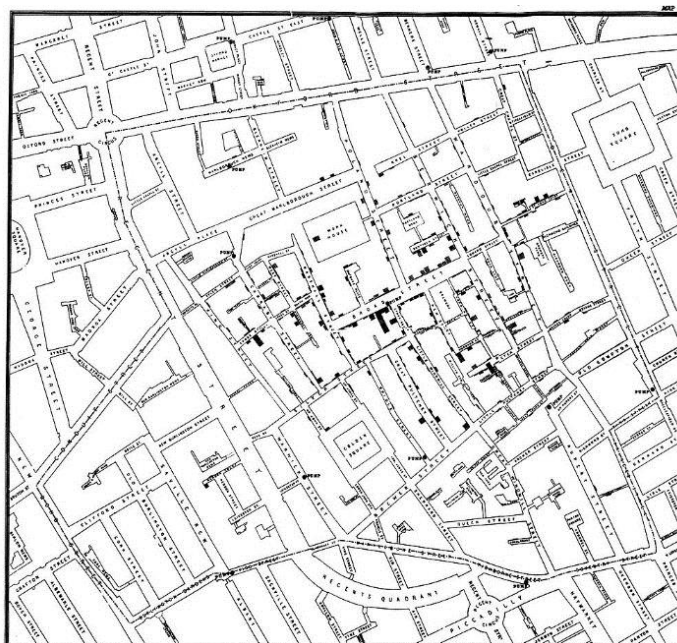


Fig. 3 – Mapa de l'estudi realitzat pel doctor John Snow

És cert, però, que quan avui dia es fa referència a SIG es parla gairebé en exclusiva de sistemes informàtics. Llavors, tornant a la definició de SIG inicial i més enfocada a l'ús actual d'aquests sistemes, es pot definir un SIG com un **conjunt de maquinari, programari i persones que suporten la captura, gestió, anàlisi i representació de dades espacials**<sup>1</sup>. En qualsevol projecte SIG, doncs, podem distingir diferents processos que es duen a terme:

- **Entrada:** Recol·lecció i emmagatzemat de dades espacials de diferents orígens.
- **Preprocés:** Tractament de les dades obtingudes per a que siguin compatibles entre sí (per exemple, fent les conversions adients entre diferents sistemes de referència).

<sup>1</sup> Diversos autors. Font: <http://giscommons.org/introduction-concepts/>.

- Anàlisi: Realitzar tasques d'anàlisi per a obtenir informació de les dades espacials.
- Sortida: Generació de representacions visuals d'aquesta informació obtinguda, en forma de mapes.

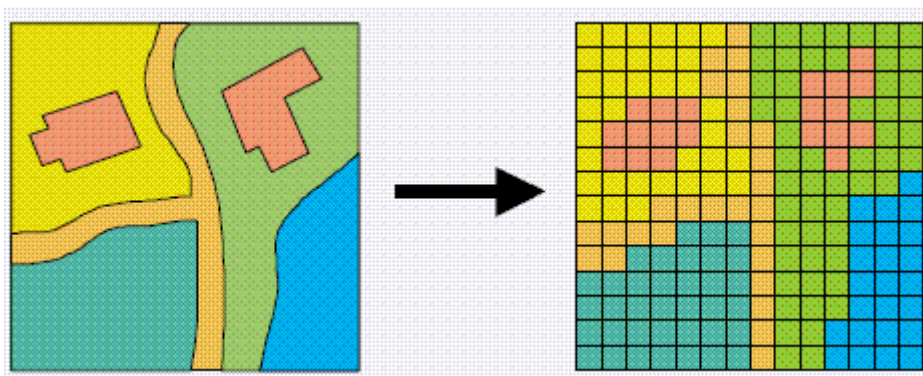
Les eines SIG actuals permeten dur a terme totes aquestes tasques, però en ocasions també pot ser interessant combinar aquestes eines amb altres peces de software que, tot i no ser pròpies d'aquest camp d'estudi, es poden integrar perfectament. A l'apartat 3 es presenta una visió global de la tecnologia actual que té a veure amb els Sistemes d'Informació Geogràfica.

## 2.2 Tipus de dades en SIG

A més dels tipus de dades tradicionals que es poden trobar en qualsevol base de dades els Sistemes d'Informació Geogràfica treballen amb dades de tipus espacial. Hi ha principalment dos models per a representar dades de tipus espacial:

### 2.2.1 Dades ràster

Al model ràster les dades es representen en forma d'una matriu rectangular de cel·les, que realment són els píxels d'una imatge, on cada una d'aquestes cel·les representa una porció del territori representat i a cada una d'elles s'associa el valor d'un atribut en forma de color. Dos cel·les amb el mateix color tenen el mateix valor per a l'atribut que s'hi representa. Per aquest motiu una capa d'informació ràster només pot representar un únic atribut. Si es vol representar més atributs calen més capes.



*Fig. 4 – Exemple de representació espacial amb el model ràster*

Les imatges ràster tenen una resolució fixada i per tant poden perdre qualitat a mesura que l'observador s'hi acosta. Aquesta simplicitat, però, permet una elevada velocitat de procés i pot ser molt útil per a representar certs tipus d'informació.

En aquest projecte es mostraran alguns exemples de capes ràster com ara les capes de topografia que proporcionen diversos serveis d'accés obert com *Open Street Maps*, *Google Maps* o l'Institut Cartogràfic i Geològic de Catalunya; o informació generada programàticament com són els mapes de calor basats en la densitat.

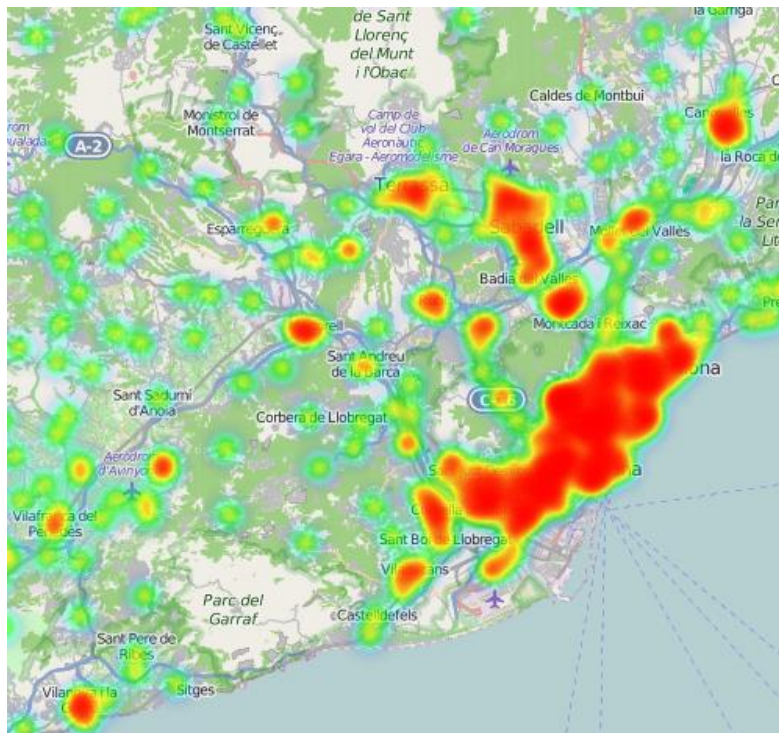


Fig. 5 – Exemple de mapa de calor (heatmap)

## 2.2.2 Dades vectorials

Al model vectorial es fan servir línies i punts per identificar els objectes que es volen representar. Aquestes línies i punts poden formar polígons per a representar àrees sobre el territori.

A les capes vectorials no s'omple tot l'espai, a diferència de com ho fan les capes ràster, i per això només hi ha informació allà on es dibuixen les figures geomètriques.

Aquesta representació es realitza de forma matemàtica i per tant permet treballar-hi en moltes resolucions amb una precisió infinita. A més, l'espai requerit per al seu emmagatzematge és molt inferior que les imatges ràster. El principal inconvenient d'aquest model de representació és el major temps de procés respecte al ràster, degut a la complexitat de la geometria.

Aquests formats, a més, permeten acompanyar la informació espacial amb informació dels tipus tradicionals de qualsevol base de dades relacional (cadena de text, nombres, etc.). D'aquesta manera s'obtenen objectes visibles a l'espai amb informació diversa associada.

## 2.3 Coordenades i projeccions geogràfiques

Quan es parla de mapes habitualment es pensa en coordenades. Les coordenades són l'eina que permet ubicar de forma exacta la posició d'un punt a la superfície terrestre.

Hi ha tres tipus de sistemes de coordenades:

- Les coordenades geogràfiques: Aquests sistemes fan servir una esfera tridimensional per a determinar la posició sobre la superfície terrestre. Un punt d'aquesta esfera es determina amb dos valors, latitud i longitud, que són el valor dels angles calculats des del centre de l'esfera fins a la posició sobre la superfície, tal com es mostra a la figura següent.

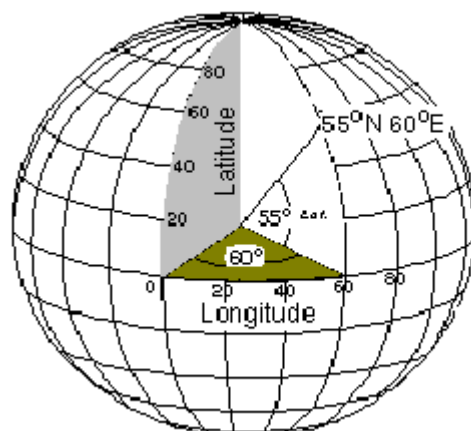


Fig. 6 – Càlcul de coordenades en un sistema de coordenades geogràfiques

Aquests sistemes divideixen l'esfera en paral·lels i meridians. L'origen de coordenades  $0^\circ$ ,  $0^\circ$  es troba a la intersecció entre el paral·lel corresponent a l'equador terrestre i el meridià de Greenwich.

- Les coordenades cartesianes: Aquests sistemes, similars als de coordenades geogràfiques prenen tres coordenades x, y i z que es calculen com la distància

entre el centre de l'esfera i la posició sobre la superfície terrestre basant-se en uns hipotètics eixos que travessen el centre de l'esfera.

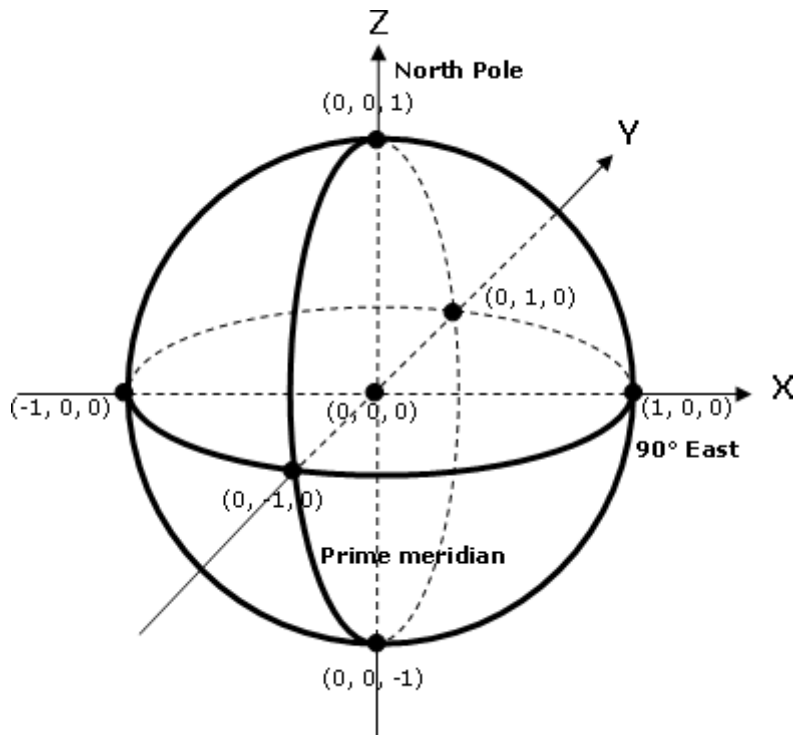
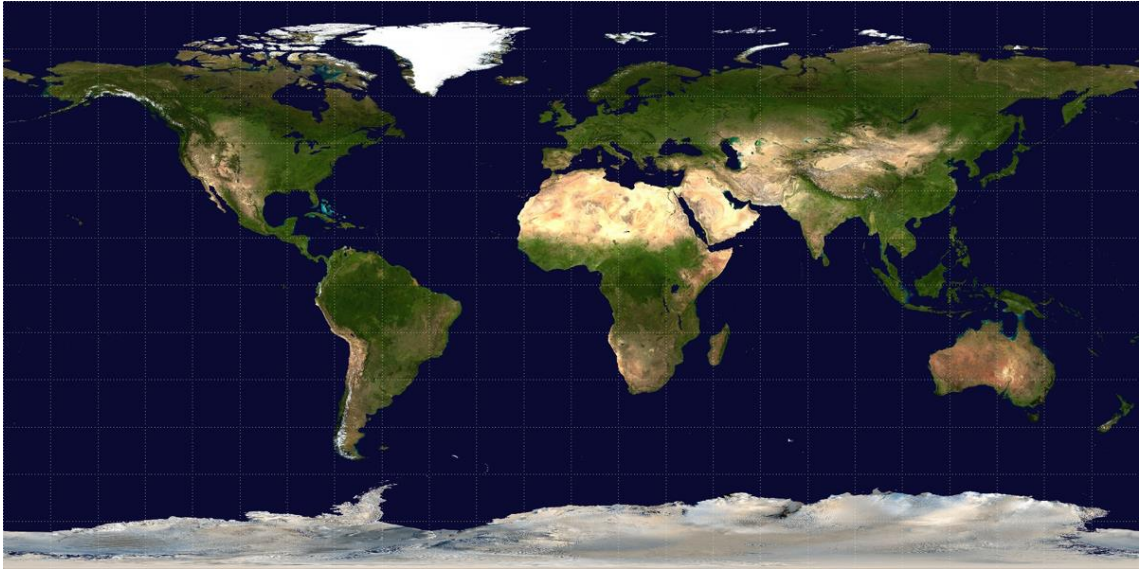


Fig. 7 - Càlcul de coordenades en un sistema de coordenades cartesianes

- Les coordenades projectades: Els sistemes de coordenades projectades porten la representació a les dues dimensions d'un pla on els punts es representen amb dues coordenades x, y amb l'origen de coordenades 0, 0 al centre del pla.

Per a treballar amb mapes, com en el cas d'aquest projecte, és més pràctic fer-ho sobre una superfície plana. És per això que es fa necessari passar d'una representació en tres dimensions a una representació bidimensional. Per a realitzar aquesta transformació es du a terme una **projecció cartogràfica**.

Hi ha infinitat de projeccions que s'han implementat al llarg dels anys. Totes elles, però, tenen mancances. En la transformació és literalment impossible passar d'una esfera a un pla sense crear distorsions. Hi ha diferents tipus de projeccions que aconsegueixen precisió en algun (però no tots) dels següents aspectes: forma, mida, escala, direcció. És necessari, doncs, triar aquella que millor representi les dades que es volen treballar. En aquest projecte, centrat exclusivament a la regió de Catalunya es farà servir un projecció de Mercator, que tot i que distorsiona molt la mida de les superfícies en latituds elevades, funciona molt bé en latituds més properes a l'equador.



*Fig. 8 – Mapamundi fent servir la projecció de Mercator*

## 3 Tecnologia existent

---

En aquest apartat es fa un ràpid cop d'ull a les diverses eines que existeixen al mercat i es defineixen quines es faran servir per al desenvolupament del projecte.

### 3.1 Eines SIG d'escriptori

Aquestes eines són les que habitualment s'anomenen simplement SIG. Són les eines que s'han definit al capítol anterior, que permeten el tractament de dades, l'anàlisi i la producció de representacions noves amb aquestes dades. Hi ha un gran nombre d'aplicacions en aquesta categoria, tant propietàries com lliures i a continuació s'enumeren alguns exemples de les més esteses:

- ArcGIS: Aplicació propietària desenvolupada per ESRI<sup>2</sup>.
- Geomedia: Aplicació propietària desenvolupada per Intergraph<sup>3</sup>.
- QGIS: Aplicació de codi obert anteriorment anomenada Quantum GIS<sup>4</sup>.
- gvSIG: Aplicació de codi obert originàriament desenvolupada per a la Generalitat Valenciana<sup>5</sup>.

Per al desenvolupament del projecte s'ha utilitzat concretament **QGIS** pels següents motius:

- És software lliure i, per tant, gratuït.
- Existeix una gran quantitat de recursos disponibles degut a la gran quantitat d'usuaris que el fa servir.

### 3.2 Bases de dades espacials

Les dades espacials es poden emmagatzemar principalment de tres maneres:

---

<sup>2</sup> <http://www.esri.es/es/productos/arcgis/>

<sup>3</sup> <http://www.intergraph.com/sgi/products/>

<sup>4</sup> <http://www.qgis.org/es/site/>

<sup>5</sup> <http://www.gvsig.com/es>

- En fitxers de dades vectorials, fent servir formats estandarditzats que puguin ser tractats per les eines SIG com ara:
  - Shapefile (.shp)
  - GeoJSON
  - Keyhole Markup Language (.kml)
  
- En fitxers de dades ràster, fent servir formats d'imatge enriquits per a ser tractats amb eines SIG com ara:
  - GeoTIFF
  - Esri Grid
  - MrSID
  
- En SGBD<sup>6</sup> espacials: aquests SGBD tenen extensions per a treballar amb dades espacials. Alguns dels més utilitzats són:
  - PostGIS
  - Oracle Spatial
  - Microsoft SQL Server

Aquest darrer tipus té alguns beneficis afegits respecte a l'ús de fitxers com ara la possibilitat d'indexar el contingut del magatzem de dades o la modificació concurrent per part de varis usuaris.

En aquest projecte es farà servir una combinació de tots els tipus enumerats.

D'una banda moltes dades obtingudes en fonts públiques a internet permeten l'exportació de dades en fitxers, per tant es faran servir aquests. I d'altra banda, un cop pretractades, les dades s'importaran a un servidor de Base de Dades Relacionals on s'emmagatzemaran per a ser consultades des de l'aplicació final.

En concret es farà servir **PostGIS** pels següents motius:

- És una extensió de PostgreSQL, un SGBD de codi lliure i, per tant, gratuït.
- S'integra nativament amb QGIS, el SIG escollit per al tractament i anàlisi de les dades.

---

<sup>6</sup> SGBD: Sistemes de Gestió de Bases de Dades.



### 3.3 Servidors de mapes

La tercera eina necessària per al desenvolupament d'aquest projecte és el servidor de mapes. Aquest tipus de servei s'encarrega d'oferir mapes a d'altres aplicacions que els mostraran, per exemple, en un visor web. Aquest, precisament, és l'ús que se'n donarà en aquest projecte.

Existeixen diversos servidors de mapes al mercat, com ara:

- MapServer: de codi lliure, basat en CGI<sup>7</sup> servits des d'un servidor Apache<sup>8</sup>.
- GeoServer: de codi lliure. És la implementació de referència de l'*Open Geospatial Consortium* (OGC).
- QGIS Server: el servidor de mapes del SIG QGIS.

En aquesta implementació s'ha fet servir **GeoServer**, en gran part per la comoditat en l'administració i gestió dels continguts que ofereix. Aquestes tasques es realitzen mitjançant un entorn d'administració web on es poden realitzar connexions amb les diferents fonts de dades així com configurar paràmetres de les dades ofertes (estil de capes, conversió de projeccions, etc.).

### 3.4 Llibreries JavaScript per a la implementació de visors web de mapes.

Internet, a més d'una bona font d'inspiració a l'hora de dur a terme un projecte com aquest, és una font inesgotable de recursos en quant a mitjans per al desenvolupament de software.

En el sector dels SIG hi ha un gran nombre de llibreries JavaScript de codi lliure que permeten desenvolupar aplicacions professionals amb quantitats molt lleugeres de codi.

En aquest projecte s'han estudiat dos coneguts *frameworks* per al desenvolupament de visors de mapes web: **Leaflet**<sup>9</sup> i **OpenLayers**<sup>10</sup>.

---

<sup>7</sup> *Common Gateway Interface* o Interfície d'entrada comú: programa que s'executa al servidor i és sol·licitat des d'un client web.

<sup>8</sup> Apache HTTP Server: un servidor web de l'Apache Software Foundation (<https://httpd.apache.org/>)

<sup>9</sup> Web del projecte Leaflet: <http://leafletjs.com>

<sup>10</sup> Web del projecte OpenLayers: <http://openlayers.org/>



Fig. 9 – Logotip del projecte Leaflet



Fig. 10 – Logotip del projecte OpenLayers

La primera, Leaflet, és una llibreria relativament nova però molt popular. És molt lleugera, amb tant sols 33 Kb de mida, la qual cosa fa molt indicada per a aplicacions mòbils, a més de ser molt atractiva visualment.

OpenLayers és segurament la biblioteca més utilitzada en el sector, amb una trajectòria més consolidada i avui dia, en la seva versió 3.5, incorpora moltes optimitzacions que la fan una competidora viable.

Després de realitzar proves de concepte amb totes dues llibreries s'acaba optant per OpenLayers degut a l'excessiva dependència de *plugins* de tercers que té Leaflet.

### 3.5 Altres llibreries JavaScript de propòsit general

Fora del sector dels SIG es troben infinitat de llibreries que permeten enriquir les aplicacions web. A continuació es descriuen les que s'han utilitzat en la implementació d'aquest projecte:

- jQuery<sup>11</sup>: Una de les llibreries de propòsit general més emprades. Permet la manipulació del DOM i dels events, modificar estils CSS, afegir animacions i fer crides AJAX.
- Bootstrap<sup>12</sup>: Una compilació de *plugins* de la llibreria jQuery per a enriquir aplicacions web.
- D3<sup>13</sup>: Llibreria per a la representació de dades. Permet la vinculació d'objectes que contenen dades amb el DOM i així aplicar transformacions que permeten, per exemple, la representació en forma de gràfics animats d'aquestes dades.

---

<sup>11</sup> Web del projecte jQuery: <https://jquery.com/>

<sup>12</sup> Web del projecte Bootstrap JS: <http://getbootstrap.com/>

<sup>13</sup> Web del projecte D3: <http://d3js.org/>

- Colorbrewer<sup>14</sup>: Petita aplicació que genera gradients de color molt útils en cartografia.

### 3.6 Altres utilitats

- JSON .NET<sup>15</sup>: Llibreria .NET amb un conjunt d'utilitats per al procés d'objectes JSON amb .NET. Aquesta llibreria permet, per exemple, convertir una cadena JSON a objectes .NET i després aplicar-hi seleccions amb expressions lambda (fent servir la sintaxi LINQ<sup>16</sup>).

---

<sup>14</sup> Web de l'aplicació Colorbrewer: <http://colorbrewer2.org/>

<sup>15</sup> Web del projecte JSON .NET: <http://www.newtonsoft.com/json>

<sup>16</sup> Més informació a: <https://msdn.microsoft.com/es-es/library/bb397926.aspx>

## 4 Estudi i tractament de les dades

---

L'objectiu d'aquest projecte és la representació de dades sanitàries de l'àmbit de Catalunya. Per aquest motiu és necessari:

- Una base cartogràfica que representi aquest territori amb precisió.
- Una capa que serveixi de suport per algunes de les dades.
- Un conjunt de dades sanitàries a visualitzar.

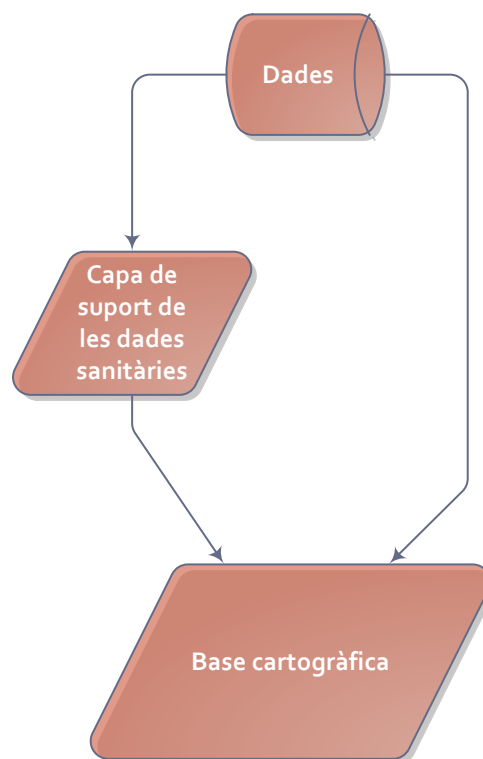


Fig. 11 – Esquema de l'estructura de capes del visor web

Tot seguit es detallen les dades amb les que s'ha treballat per a la implementació del visor, quines transformacions s'hi ha realitzat i un exemple dels resultats obtinguts.

### 4.1 Dades cartogràfiques

Tenint en compte que el projecte es centra en una zona concreta del planeta, en aquest cas Catalunya, s'han estudiat diferents opcions pel que fa a una base cartogràfica que suporti les dades que es volen representar. Cada una d'elles té els seus punts forts i febles.

## OpenStreetMaps

### Descripció

*OpenStreetMaps* o *OSM* és un servei de mapes d'accés lliure amb nombrosos servidors WMS de tercers per a sol·licitar mapes des d'altres aplicacions, com per exemple un visor web com el d'aquest projecte.

Disposa de la cartografia de tot el món i s'actualitza de forma col·laborativa per usuaris d'arreu del planeta.

### SRC

EPSG:3857. Sistema de referència de coordenades projectades amb una projecció de Mercator. També s'anomena WGS84 i és l'SRC més estès en visualitzadors web com ara aquest o GoogleMaps.

### Ús al projecte

Aquesta font de mapes s'ha pres com la base per aquest projecte pels motius següents:

- Cobreix tot el món. Això, tot i no ser imprescindible degut a l'abast de les dades que s'hi representaran, poden permetre que usuaris que no coneguin Catalunya s'ubiquin millor.
- Està integrat amb la llibreria JavaScript utilitzada, OpenLayers 3.
- Altres serveis similars requereixen llicències d'ús per accedir a l'API.

S'han incorporat diverses visualitzacions d'aquest servei. A més de la capa bàsica d'OSM, que es veu a l'exemple següent, s'ha inclòs la possibilitat de visualitzar altres capes de tercers:

- Stamen Toner: visualització d'alt contrast blanc-negre.
- Map Quest Satellite: visualització d'imatges satèl·lit.

### Exemple

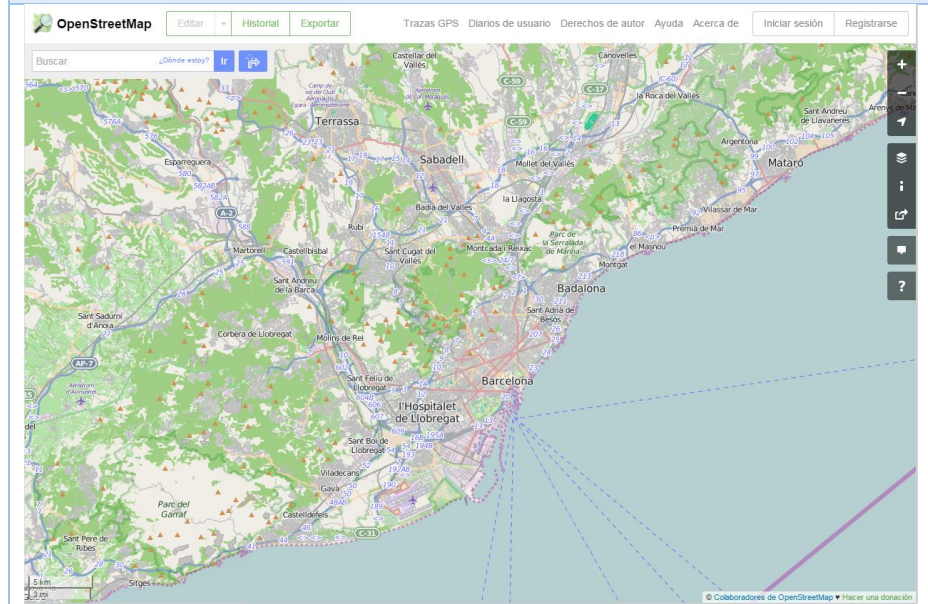


Fig. 12 – Mapa obtingut del propi visor web del projecte OSM

## Institut Cartogràfic i Geològic de Catalunya

Descripció	<p>L'ICGC disposa d'una base cartogràfica del territori català així com nombroses capes temàtiques d'aquest mateix àmbit.</p> <p>Permet la connexió mitjançant el protocol estàndard WMS per a sol·licitar mapes des d'altres aplicacions, com per exemple un visor web com el d'aquest projecte.</p> <p>També ofereix la possibilitat de descarregar fitxers de dades vectorials per a elaborar mapes derivats.</p>
SRC	<p>Diversos. Per a aquest projecte s'utilitzarà el servei WMS ràpid en coordenades geogràfiques ja que pot projectar el mapa sobre el EPSG:3857 i així fer que "encaixi" amb les altres capes cartogràfiques.</p> <p>S'utilitza com a capa cartogràfica secundària ja que està limitada a Catalunya. Es permetrà a l'usuari activar-la a discreció.</p>
Ús al projecte	<p>No obstant s'ha utilitzat la Base Municipal que proporciona l'ICGC com a punt de partida per a l'elaboració de la capa de suport de les dades. Aquesta base està formada per un conjunt de capes vectorials que defineixen els límits dels diferents nivells administratius del territori català: els municipis i les províncies. Es descarreguen els fitxers en format .shp (shapefile) per a ser tractats més endavant (es detalla al següent apartat).</p>

### Exemple

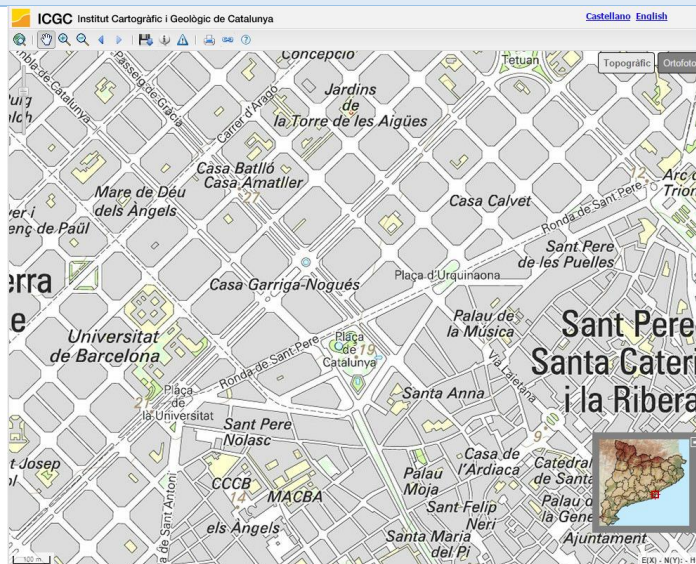


Fig. 13 – Mapa obtingut del visor web de l'ICGC

## 4.2 Capa de suport per a les dades sanitàries

Les dades sanitàries que es volen representar són principalment dades quantitatives o qualitatives indexades sobre una referència espacial. Per posar un exemple, la Generalitat ofereix llistats amb el recompte d'ingressos hospitalaris en un municipi concret. Com es veu, en aquest cas hi ha una dada espacial clara, el municipi. Però en altres ocasions les dades poden estar agrupades, per exemple, per Àrea Bàsica de Salut.

Tot i que qualsevol d'aquests elements són vàlids, s'ha decidit optar per només un d'ells com a base per a representar les dades i aquest és el de les Regions Sanitàries. Aquesta és l'agrupació més externa que fa el Departament de Salut de la Generalitat la qual serveix per a descentralitzar el sistema sanitari i fer-lo més proper a tota la població. Correspon a set regions amb una extensió similar que divideixen el territori català:

- Alt Pirineu i Aran
- Barcelona
- Camp de Tarragona
- Catalunya Central
- Girona
- Lleida
- Terres de l'Ebre



Fig. 14 – Regions sanitàries de Catalunya

El Departament de Salut no proporciona aquesta divisió en cap format que es pugui treballar amb eines SIG. És per aquest motiu que es decideix l'elaboració pròpia, fent servir com a base la Base Municipal de l'ICGC.

A la definició de cada regió sanitària<sup>17</sup> es determina quines comarques i/o municipis en formen part. Per exemple, la regió de Girona inclou:

- Les comarques d'Empordà, Baix Empordà, Garrotxa, Gironès, Pla de l'Estany, Ripollès i Selva.
- Els municipis del Maresme: Calella, Canet de Mar, Malgrat de Mar, Palafolls, Pineda de Mar, Sant Cebrià de Vallalta, Sant Iscle de Vallalta, Sant Pol de Mar, Santa Susanna i Tordera (la resta de municipis del Maresme pertanyen a la regió Barcelona).

Fent servir QGis s'ha generat una nova capa vectorial on cada regió està formada per la suma ("merge") dels municipis que hi pertanyen.

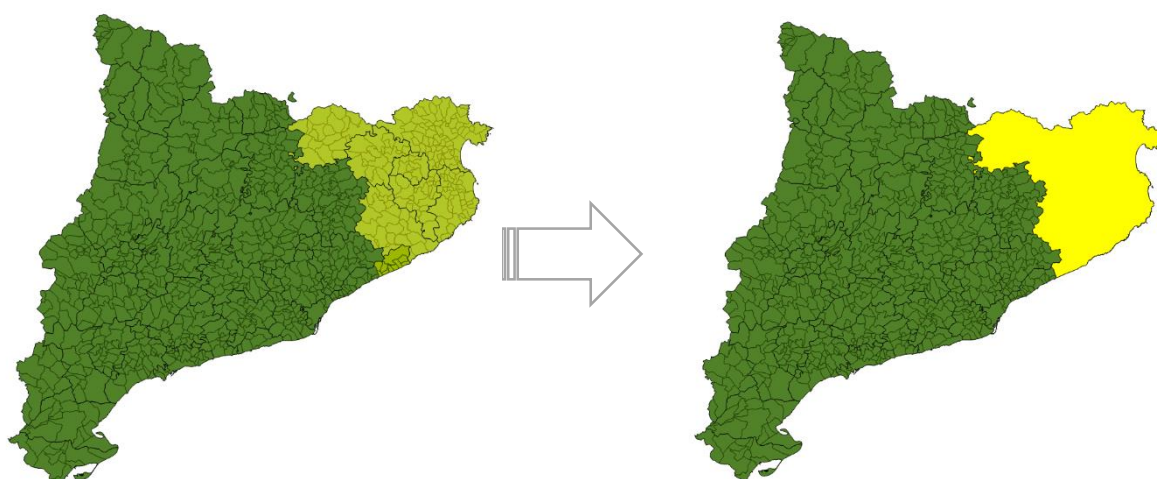


Fig. 15 – "Merge" d'objectes vectorials per a la creació de la capa de suport de Regions Sanitàries

Finalment s'incorpora aquesta capa com una taula de PostgreSQL fent servir les pròpies eines d'importació de què disposa QGis.

---

<sup>17</sup> <http://catsalut.gencat.cat/ca/catsalut-territori/regions-sanitaries/>



### 4.3 Dades sanitàries

El departament de Salut de la Generalitat de Catalunya proporciona, a través del seu portal de Dades Obertes<sup>18</sup>, informació compilada sobre diferents aspectes de la sanitat pública catalana.

Entre aquestes dades podem trobar informació sobre temes com ara:

- Informació sobre el diagnòstic de malalties als centres públics
- Informació sobre donacions de sangs
- Catàlegs de centres sanitaris
- Indicadors sobre l'estat de la salut la població

Cada una d'aquestes fonts està elaborada per diferents organismes i per tant no estan estructurades. És per aquest motiu que les fonts que s'han incorporat en aquest projecte s'han hagut de tractar per a poder-les representar sobre una mateixa base, en aquest cas la de regions sanitàries, la qual s'ha descrit a l'apartat anterior.

A continuació es detalla el tractament que s'ha fet les dades obtingudes:

#### ***Població assegurada***

<i>Origen</i>	Dades obertes del Departament de Salut de la Generalitat de Catalunya - <a href="http://observatorisalut.gencat.cat/ca/detalls/article/13_IND_poblacio_assegurada_CatSalut">http://observatorisalut.gencat.cat/ca/detalls/article/13_IND_poblacio_assegurada_CatSalut</a>
<i>Format</i>	Full de càlcul (.xlsx)
<i>SRC (si procedeix)</i>	N/A
<i>Descripció</i>	<p>Conjunt de fulls de càlcul que contenen la cobertura sanitària de la població on les dades s'agrupen per diferents criteris:</p> <ul style="list-style-type: none"><li>- Edat</li><li>- Territori</li><li>- Sexe</li></ul> <p>Aquestes dades són importants per a donar pes a les diferents regions obtingudes per a la capa de suport.</p>
<i>Tractament realitzat</i>	S'han realitzat resums per regió sanitària i aquestes dades s'han afegit a la taula PostGIS de Regions. S'han utilitzat les dades més recents (any 2015).

<sup>18</sup> Dades Obertes de la Generalitat: <http://dadesobertes.gencat.cat/ca/>

A continuació es mostra una visualització en QGis que mostra la capa de suport de Regions realçada amb colors segons la quantitat d'habitants assegurats pel sistema de salut pública.

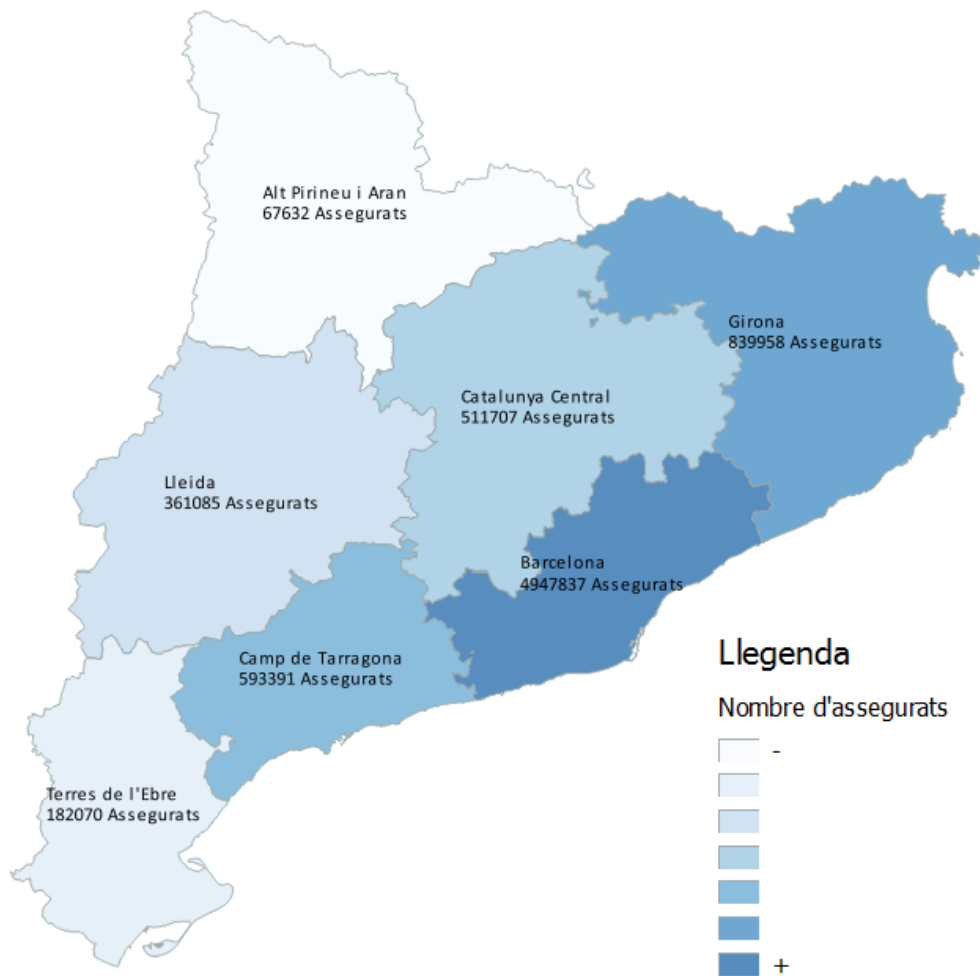


Fig. 16 – Regions sanitàries amb nombre d'assegurats

### Centres sanitaris

Origen	Explorador de dades d'equipaments de la Generalitat de Catalunya - <a href="http://serveisoberts.gencat.cat/equipaments/explorer">http://serveisoberts.gencat.cat/equipaments/explorer</a>
Format	XML (.xml)
SRC (si procedeix)	ED50 / UTM Zone 31N (EPSG 23031)
Descripció	Fitxer XML amb informació de tot tipus referent als centres sanitaris de Catalunya. Entre aquestes dades hi ha les coordenades geogràfiques i les coordenades projectades.

Tractament  
realitzat

El fitxer XML té un format que es fàcilment importable a una taula de base de dades. Es realitzen les següents tasques:

- Càrrega del fitxer XML com una taula a PostGIS.
- Creació d'un camp de geometria de punts basada en els camps que contenen les coordenades UTM (projectades). Per a això es fa servir la funció ST\_GeomFromText de PostGIS.
- Càrrega de la taula com una capa vectorial a QGIS.
- Fer servir la conversió de sistemes de referència "al vol" de QGIS per a que la capa es representi en l'SRC ETRS89, com les altres fonts de dades.

A continuació es mostra una visualització en QGIS de la capa generada sobre la capa de regions sanitàries generada anteriorment.

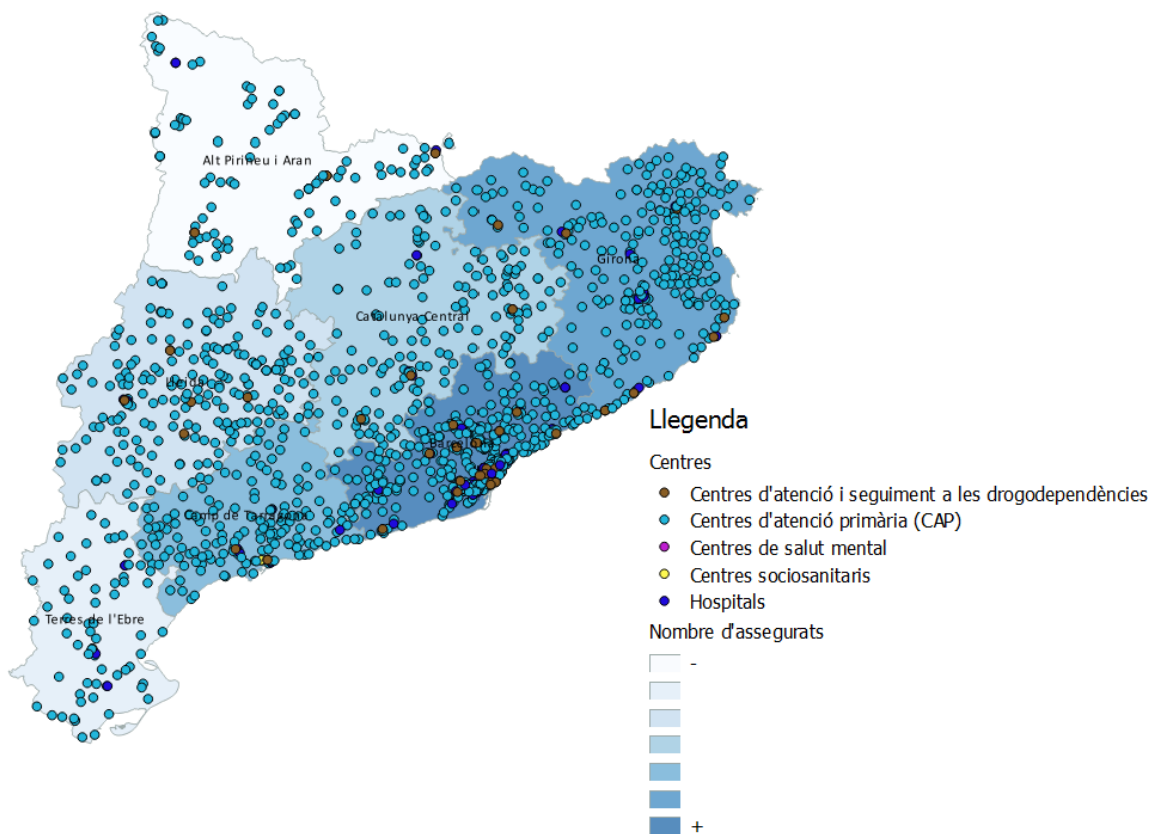


Fig. 17 – Capa de centres sanitaris

## Donacions de sang

Origen

Dades obertes del Departament de Salut de la Generalitat de Catalunya -  
<http://dadesobertes.gencat.cat/ca/cercador/detall-cataleg/?id=4090>

<i>Format</i>	Full de càlcul (.xls)
<i>SRC (si procedeix)</i>	N/A
<i>Descripció</i>	<p>Full de càlcul amb el nombre de donacions de sang realitzades a Catalunya en un any, agrupat per municipi, comarca i regió sanitària.</p> <p>Conté dades dels anys 2012 i 2013</p>
<i>Tractament realitzat</i>	<p>Es prenen les dades referents al nombre de donacions per cada regió sanitària i s'incorporen a la taula de regions creada anteriorment a PostGIS.</p> <p>Cal tenir en compte, però, que si es vol comparar entre regions sanitàries no és factible fer servir el nombre total de donacions a cada regió ja que, òbviament, la regió amb més població serà la que més donacions generi.</p> <p>Per això s'ha creat també un camp calculat per a obtenir el rati de donacions cada 1000 habitants assegurats.</p> <p>A continuació es mostra una visualització en QGis que mostra la capa de suport de Regions realçada amb colors segons aquest rati de donacions per 1000 habitants assegurats.</p>

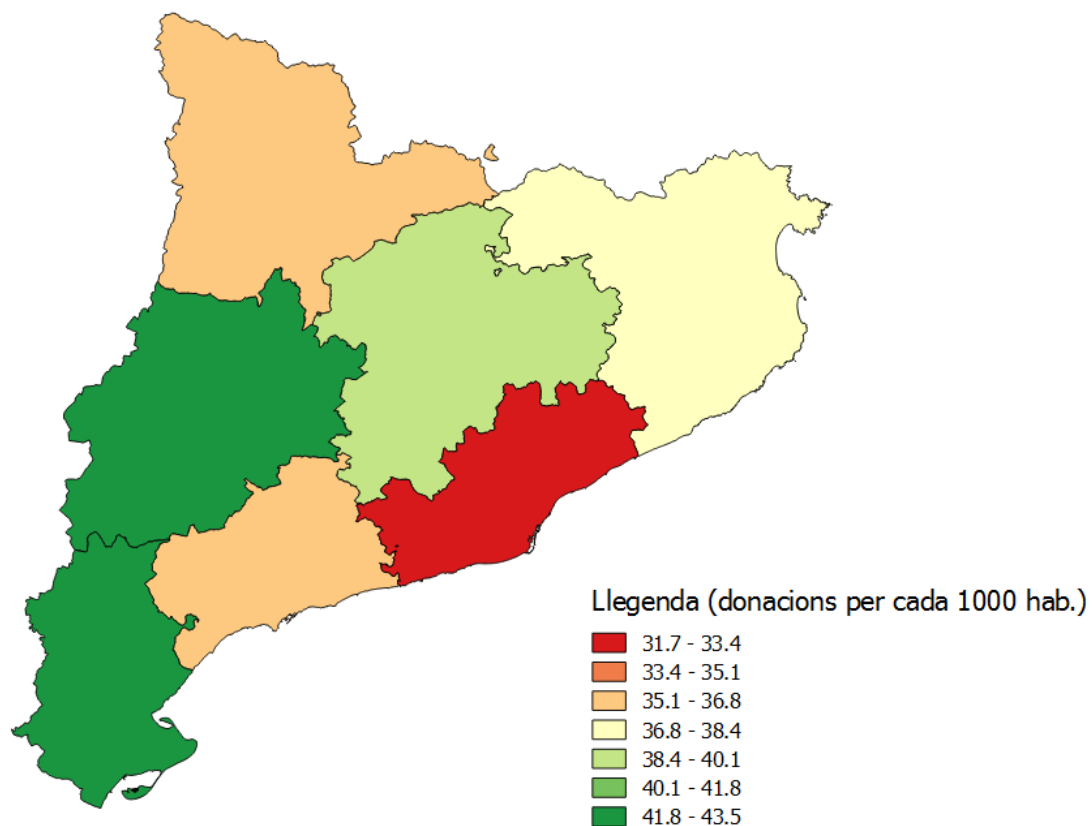


Fig. 18 – Capa de donacions de sang per cada 1000 habitants assegurats (any 2013)

## 5 Implementació del servidor de mapes

El servidor de mapes que s'utilitza en aquest projecte és GeoServer. Es tracta d'una plataforma de codi lliure implementada en Java que permet la publicació de mapes per a ser consumits per altres aplicacions.

En el desenvolupament d'aquest projecte s'ha instal·lat sobre una estació Windows fent servir el port 8080, normalment reservat per a servidors web secundaris. Es fa servir aquest port tant per configurar el propi servidor com per realitzar peticions des d'altres aplicacions.

L'accés al portal de configuració de GeoServer, una vegada instal·lat, es realitza navegant a la URL <http://localhost:8080/geoserver>.



Identificado como admin. [Cerrar sesión](#)

### Acerca de GeoServer

Información general sobre GeoServer

#### Información de compilación

**Version**  
2.7.0

**Revisión de Git**  
b7c9bf9bbe0c5f86a3d30e132146ac9fa368b4b

**Fecha de compilación**  
20-Mar-2015 19:44

**Versión de GeoTools**  
13.0 (rev 3e400364d95c06a3e449c34dbc3c587a12bbdfb6)

**GeoWebCache Version**  
1.7.0 (rev 1.7.x/c87a6d54381fd571424095702a43a6fd403e5148)

#### More Information

El proyecto GeoServer es una implementación Java (J2EE) totalmente transaccional de las especificaciones Web Feature Service, Web Coverage Service y Web Map Service del Open Geospatial Consortium.

La documentación para esta versión se encuentra disponible online en la siguiente ubicación. La wiki de GeoServer se utiliza para las últimas actualizaciones. Por favor utilícela para compartir sus experiencias, pistas y trucos sobre GeoServer. El organizador de tareas es el lugar donde informar sobre defectos y realizar peticiones de nuevas funcionalidades. También sea tan amable de agregarse usted mismo al Mapa de Usuarios para demostrar su apoyo a GeoServer.

Useful Links:

- [Documentación](#)
- [Wiki](#)
- [Seguimiento de incidencias](#)

**Servidor**

- Estado del servidor
- Logs de GeoServer
- Información de contacto
- Acerca de GeoServer

**Datos**

- Previsualización de capas
- Espacios de trabajo
- Almacenes de datos
- Capas
- Grupos de capas
- Estilos

**Servicios**

- WCS
- WFS
- WMS

**Settings**

- Global
- JAI
- Coverage Access

**Tile Caching**

- Tile Layers
- Caching Defaults
- Gridsets
- Disk Quota

**Seguridad**

- Settings
- Authentication
- Passwords
- Users, Groups, Roles
- Seguridad de los datos
- Seguridad de los servicios

Fig. 19 – Accés al panell de configuració del servidor GeoServer

GeoServer permet servir mapes de diferents tipus de fonts de dades. En aquesta parametrització s'ha creat un espai de treball per emmagatzemar-hi aquestes fonts:

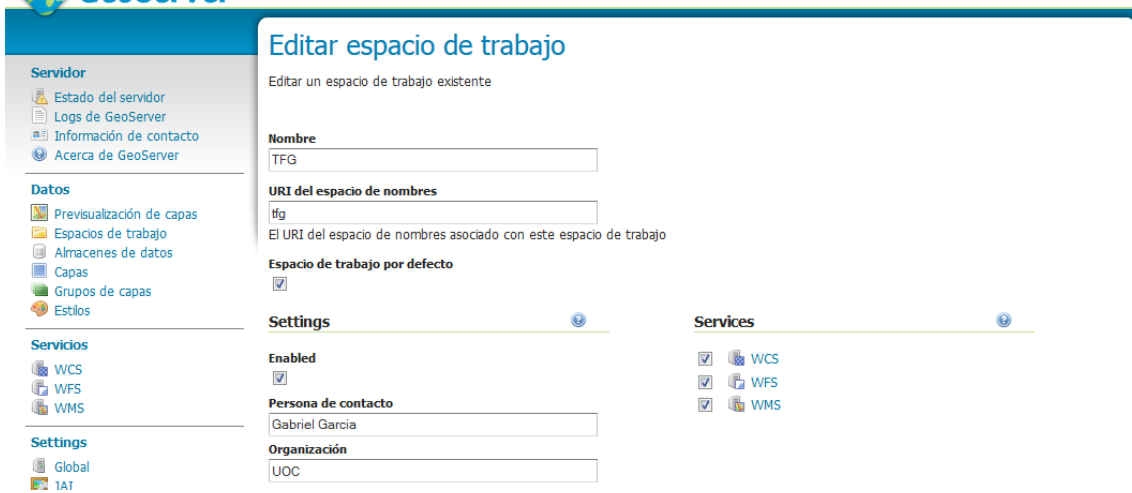


Fig. 20 – Configuració de l'espai de treball a GeoServer

I tot seguit s'han configurat alguns orígens de dades, que en GeoServer s'anomenen "Magatzems de dades". En aquest cas d'estudi hi ha magatzems de tres tipus diferents:

- Directori amb fitxers de dades vectorials en format *shapefile*.
- Directori amb fitxers de dades ràster en format GeoTIFF.
- Connexió a una base de dades PostGIS.

Hi ha, però, altres tipus d'orígens de dades disponibles a GeoServer com ara la connexió a d'altres servidors per fer un servei WMS en cascada.

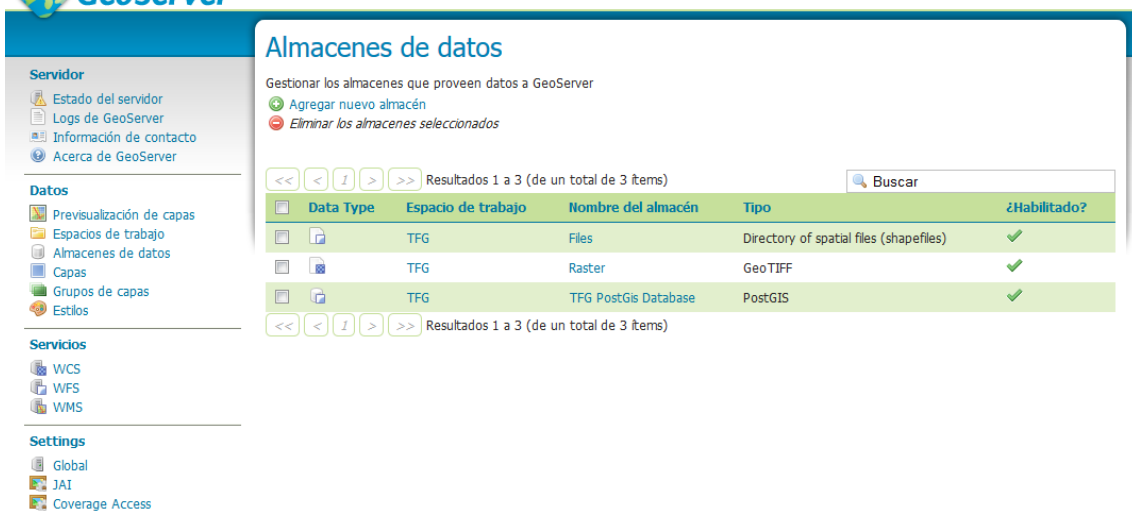


Fig. 21 – Magatzems de dades utilitzats a GeoServer

Finalment s'han configurat les capes que s'obtenen de cada magatzem. En aquesta parametrització cal tenir presents alguns paràmetres importants de cara a la implementació del visor web:

- El sistema de referència: cal indicar en quin sistema arriben les dades i en quin sistema es volen transformar. A continuació es mostra un exemple on es reprojecta una capa en ED50 (com les obtingudes de l'ICGC) cap a WGS84, que és la projecció amb la que treballa la base cartogràfica triada, OpenStreetMaps.

**Sistema de referencia de coordenadas**

---

**SRS nativo**  
 [EPSG:ED50 / UTM zone 31N...](#)

**SRS declarado**  
  [EPSG:WGS 84 / Pseudo-Mercator...](#)

**Gestión de SRC**

Fig. 22 – Reprojectat de sistema de referència d'una capa a GeoServer

- L'extensió: cal indicar els límits geogràfics de les dades de la capa. Aquestes coordenades es poden calcular de forma automàtica per GeoServer que les introduirà en base a la projecció utilitzada.

**Encuadres**

---

**Encuadre nativo**

Min X	Min Y	Máx X	Máx Y
17.443.47818428	4.942.734,267002	366.579.0786135	5.292.220.712059

[Calcular desde los datos](#)

**Encuadre Lat/Lon**

Min X	Min Y	Máx X	Máx Y
0,1566974306114	40,526396635719	3,2930358915696	42,870100757521

[Calcular desde el encuadre nativo](#)

Fig. 23 – Parametrizació de l'extensió d'una capa a GeoServer

- L'estil: GeoServer permet publicar les dades amb uns paràmetres d'estil preconfigurats. El servidor disposa d'alguns estils per defecte però es poden definir de nous fent servir l'estàndard SLD<sup>19</sup>. Es tracta de fitxers XML amb un marcatge bastant intuïtiu. El següent exemple mostra com obtenir l'estil per a representar punts com un cercle blanc amb contorn negre:

<sup>19</sup> Definició de l'estàndard SLD (*Styled Layer Descriptor*) a l'Open Geospatial Consortium: <http://www.opengeospatial.org/standards/sld>

```

<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor version="1.0.0"
  xsi:schemaLocation="http://www.opengis.net/sld
http://schemas.opengis.net/sld/1.0.0/StyledLayerDescriptor.xsd"
  xmlns="http://www.opengis.net/sld"
  xmlns:ogc="http://www.opengis.net/ogc"   xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <NamedLayer>
    <Name>capitals</Name>
    <UserStyle>
      <Name>capitals</Name>
      <Title>Capital cities</Title>
      <FeatureTypeStyle>
        <Rule>
          <Title>Capitals</Title>
          <PointSymbolizer>
            <Graphic>
              <Mark>
                <WellKnownName>circle</WellKnownName>
                <Fill>
                  <CssParameter name="fill">
                    <ogc:Literal>#FFFFFF</ogc:Literal>
                  </CssParameter>
                </Fill>
                <Stroke>
                  <CssParameter name="stroke">
                    <ogc:Literal>#000000</ogc:Literal>
                  </CssParameter>
                  <CssParameter name="stroke-width">
                    <ogc:Literal>2</ogc:Literal>
                  </CssParameter>
                </Stroke>
              </Mark>
              <Opacity>
                <ogc:Literal>1.0</ogc:Literal>
              </Opacity>
              <Size>
                <ogc:Literal>6</ogc:Literal>
              </Size>
            </Graphic>
          </PointSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>

```

Aquest codi obté el següent resultat:

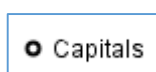


Fig. 24 – Exemple d'estil SLD per a punts



Amb aquest llenguatge es poden obtenir idèntics resultats als obtinguts amb, per exemple, QGis. De fet, aquest SIG permet l'exportació d'estils en format SLD. A continuació es mostra el resultat de donar estil a la capa de regions sanitàries fent servir SLD per realçar cada element segons la quantitat d'assegurats que té cada regió:

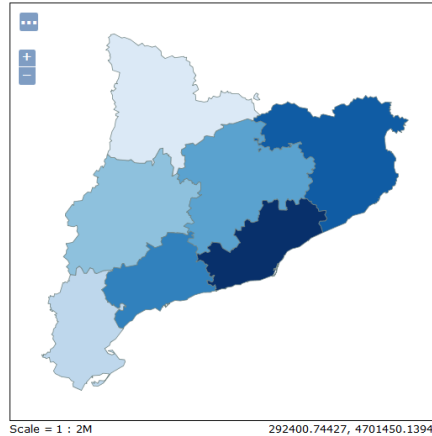


Fig. 25 – Capa de regions sanitàries amb estil SLD a GeoServer

Un cop configurades les capes a GeoServer ja poden ser consultades per altres aplicacions.

Des del visor web implementat en aquest projecte es fa ús de dos protocols que proporciona GeoServer: WMS i WFS.

## 5.1 Sol·licituds WMS

Aquest acrònim significa *Web Map Service* i és un servei que proporciona mapes en forma d'imatges en diferents formats com ara PNG, JPG, GIF, GeoTIFF o SVG entre d'altres.

Per a realitzar peticions a aquest servei s'empren URL en un format definit per aquest protocol. Un exemple de petició al GeoServer d'aquest projecte seria:

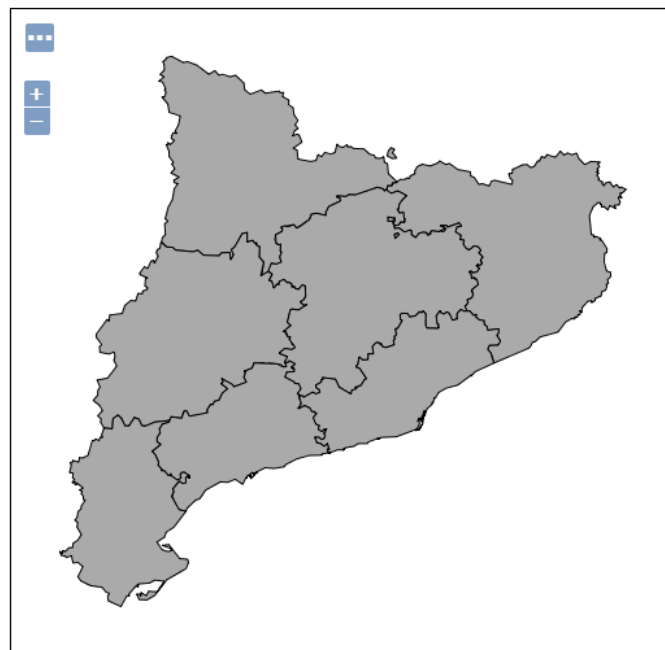
<http://localhost:8080/geoserver/TFG/wms?service=WMS&version=1.1.0&request=GetMap&layers=TFG:Regions&styles=&bbox=258852.90625,4487482.5,528738.0625,4749277.0&width=512&height=496&srs=EPSG:3043&format=application/openlayers>

Descomposant la petició s'obtenen les parts:

- <http://localhost:8080/geoserver/TFG/wms?service=WMS&version=1.1.0>: URL bàsica del servei WMS. La cadena "TFG" correspon al nom de l'espai de treball configurat prèviament. Els paràmetres "service" i "version" indiquen que es farà servir el protocol WMS en la seva versió 1.1.0.

- request: és el tipus de petició que es durà a terme. En aquest cas es realitza la petició *GetMap* que retorna una capa d'aquest espai de treball en forma de imatges. Hi ha altres tipus de peticions com ara *GetCapabilities* que retorna un XML amb una descripció de les capacitats del servidor amb aquest protocol.
- layers: defineix el nom de la capa que es vol recuperar.
- bbox: Significa "Boundary Box" i són les coordenades que defineixen els límits de la capa a representar
- width i height: serveix per a determinar la mida que tindrà la imatge resultant.
- srs: identifica el sistema de referència en el qual es vol el resultat. En aquest cas es vol fer una transformació a l'EPSG:3043.
- format: determina el tipus MIME de les imatges resultants. En aquest cas es retorna com un mapa d'openlayers, que ve integrat amb GeoServer, i que permet certa navegació pel resultat. Es podria, però, demanar altres formats com ara "image/png" o "image/jpeg".

A continuació es mostra el resultat d'aquesta petició. Com es veu, es tracta d'una visualització molt bàsica sense estil, ja que la capa no està parametritzada amb SLD.



Scale = 1 : 2M  
Click on the map to get feature info

Fig. 26 – Capa de regions sanitàries obtinguda amb WMS de GeoServer

## 5.2 Sol·licituds WFS

L'altre protocol que es fa servir en aquest projecte és WFS, que és l'acrònim de *Web Feature Service*. Amb aquest protocol, a diferència de WMS, es poden obtenir mapes en formats vectorials com per exemple shapefile, GeoJSON o KML.

Al projecte es realitzen peticions WFS que retornen el format GeoJSON ja que aquests resultats són fàcilment tractables programàticament ja sigui amb JavaScript (recordem que JSON significa *JavaScript Object Notation*) o amb C# (llenguatge triat en la programació de la part servidor detallat al capítol 6).

Igual que amb WMS, a continuació es mostra un exemple de petició WFS i se'n descriuen alguns dels camps més rellevants:

<http://localhost:8080/geoserver/TFG/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=TFG:Regions&maxFeatures=50&outputFormat=application/json>

Descomposant la petició s'obtenen les parts:

- <http://localhost:8080/geoserver/TFG/ows?service=WFS&version=1.0.0>: URL bàsica del servei WFS. La cadena "TFG" correspon al nom de l'espai de treball configurat prèviament. Els paràmetres "service" i "version" indiquen que es farà servir el protocol WFS en la seva versió 1.0.0.
- request: és el tipus de petició que es demana al servei WFS. En aquest cas GetFeature retorna els objectes existents en una capa determinada. Hi ha altres tipus de peticions com ara *GetCapabilities* que, com amb WMS, retorna un fitxer XML amb les capacitats del servei.
- typeName: determina el nom de la capa que es vol obtenir.
- maxFeatures: es pot determinar el nombre màxim d'objectes que es vol recuperar. El sistema predetermina que siguin 50, però es pot demanar tots els objectes ometent aquest paràmetre. Cal tenir en compte sempre què és el que es vol representar i contraposar-ho amb el rendiment que es vol obtenir. Cal recordar que el tractament de dades vectorials és molt més complex que amb dades ràster i una consulta molt costosa pot tornar l'aplicació lenta i per tant una mala experiència d'usuari.
- outputFormat: aquest paràmetre defineix el format desitjat per al resultat. En aquest cas es vol obtenir el resultat en format GeoJSON.

L'exemple següent mostra una porció de fitxer GeoJSON obtingut de la sol·licitud anterior:

```
{"type":"FeatureCollection","totalFeatures":7,"features":[{"type":"Feature","id":"Regions.1","geometry":{"type":"MultiPolygon","coordinates":[[[[[458209.5962104801,4656729.357583106],[457884.5012544639,4657460.561183445],[457868.85138627584,4657481.131334081],...  
...[298365.8526619738,4493866.767893654]]]]},"geometry_name":"poly","properties":{"regio":"Terres de l'Ebre","bbox":[260188.9733779255,4488778.589263272,321407.716646613,4579076.381763791]}],"crs":{"type":"name","properties":{"name":"urn:ogc:def:crs:EPSG::3043"},"bbox":[258852.90625,4487482.5,528738.0625,4749277]}
```

## 6 Disseny del visor web

---

La implementació del visor web, com tota aplicació informàtica que es desenvolupa, requereix un anàlisi i disseny previs.

En aquest capítol es detallen les diferents fases d'aquests anàlisi i disseny.

### 6.1 Decisions sobre l'entorn de treball en el desenvolupament del producte

Un cop decidides algunes de les tecnologies que donaran suport al desenvolupament del producte cal decidir quines eines es faran servir en la programació pròpiament.

Òbviament, tractant-se de una aplicació web, el resultat seran pàgines HTML. Cal decidir, però, si serà necessari algun altre llenguatge per la part servidor.

Degut a l'experiència de l'autor en desenvolupament web, es decideix que es desenvoluparà una aplicació Client-Servidor sobre la plataforma .NET de Microsoft. En concret es farà ús del *framework* MVC .NET<sup>20</sup> amb llenguatge C#. Aquest *framework* permet la creació d'aplicacions web seguint el patró Model-Vista-Controlador. Aquest patró de programació permet una separació clara entre les tres funcions

- Model: s'encarrega de la part de "negoci". Modelitza, com el propi nom indica, aquells elements propis del cas concret que s'està implementant. En el cas d'aquest projecte el Model s'encarregaria de treballar amb conceptes com "Capa", "Punt" o "Polígon" i d'accedir a bases de dades.
- Vista: s'encarrega de la representació visual. No obté ni modifica dades, simplement les rep i les adapta per a presentar-les a l'usuari.
- Controlador: és l'encarregat de lligar el Model amb la Vista. Respon a les interaccions de l'usuari i retorna a la Vista la informació del Model per a ser representada.

Donat que l'entorn de desenvolupament (IDE) per a tecnologies .NET és Visual Studio i aquest és un software propietari, s'ha utilitzat una llicència universitària mitjançant la plataforma Dreamspark<sup>21</sup> de Microsoft. D'aquesta manera es poden desenvolupar

---

<sup>20</sup> Web del projecte MVC .NET: <http://www.asp.net/mvc>

<sup>21</sup> Portal Dreamspark: <https://www.dreamspark.com/>

aplicacions sense patir els elevats costos de llicència de Microsoft, sempre i quan no se'n faci un ús comercial dels productes derivats.

En desenvolupament web la publicació dels resultats es realitza sobre servidors web com ara IIS o Apache. En aquest projecte la publicació es farà mitjançant dues eines:

- En una primera fase es treballarà a la màquina de desenvolupament i, per tant, s'utilitzarà el servidor de desenvolupament de Visual Studio que va integrat sobre el propi IDE.
- En una segona fase es publicarà sobre un servidor IIS (*Internet Information Services*) per fer l'aplicació pública a la xarxa.

Per últim cal pensar en algun software de control de versions. Tot i ser un projecte individual on no cal compartir codi entre desenvolupadors, fer ús d'aquestes eines és una bona pràctica: permet emmagatzemar històric de canvis entre jornades de treball i alhora permet la recuperació de desastres.

Amb això en ment, es decideix fer ús de Github<sup>22</sup> on s'emmagatzemen tots els canvis realitzats al llarg del desenvolupament. Github ofereix de forma gratuïta repositoris públics per a l'emmagatzematge de projectes i disposa d'una eina d'escriptori per a sincronitzar tots els fitxers de forma senzilla.

A mode de sumari es presenta a continuació el resum de tecnologies utilitzades al projecte:

<b>Resum de tecnologies utilitzades</b>	
<i>SIG per al tractament de dades espacials</i>	QGis 2.8
<i>Base de dades espacials</i>	PostGIS 2.1
<i>Servidor de mapes</i>	GeoServer 2.7
<i>Llibreries JavaScript</i>	OpenLayers 3.5 jQuery 2.1.4 Bootstrap 3.3.4 D3 3.5.5
<i>IDE</i>	Visual Studio 2010 .NET Framework 4
<i>Servidor web</i>	IIS 7

---

<sup>22</sup> Lloc web de Github: <https://github.com/>

## 6.2 Anàlisi de requisits

El visor web de dades sanitàries ha de permetre:

- Visualització de cartografia centrada en l'àmbit de Catalunya. Permetrà a l'usuari navegar aquesta cartografia amb els controls habituals d'un visor de mapes web: moure el focus i canviar el zoom. També permetrà activar o desactivar aquesta capa així com altres que hi hagi carregades.
- Seleccionar capes d'informació sobre diferents temes relacionats amb la sanitat pública. Aquestes capes es visualitzaran sobre la cartografia de base.
- Els elements de les capes visibles han de ser seleccionables. En la selecció d'un element es mostrarà informació detallada sobre aquest.
- La capa d'informació visible mostrarà un resum de les dades. Depenent de les dades aquest resum pot ser, per exemple, un gràfic de sectors que mostri la proporció d'algun atribut entre els diferents elements visibles.
- En tot moment hi haurà una llegenda que tradueixi la codificació de colors que es vegi en aquell moment.
- En capes que tinguin una gran quantitat d'elements es disposarà d'un cercador que permeti introduir una cadena de text i així filtrar el resultat de manera que només es vegin aquells que compleixin amb el criteri escollit.
- En tot moment seran visibles les coordenades corresponents a la posició del cursor.
- Es podrà exportar la capa actual en format GeoJSON, si està disponible.

### 6.3 Arquitectura

Al diagrama següent es mostra de forma esquemàtica l'arquitectura client-servidor que s'implementarà:

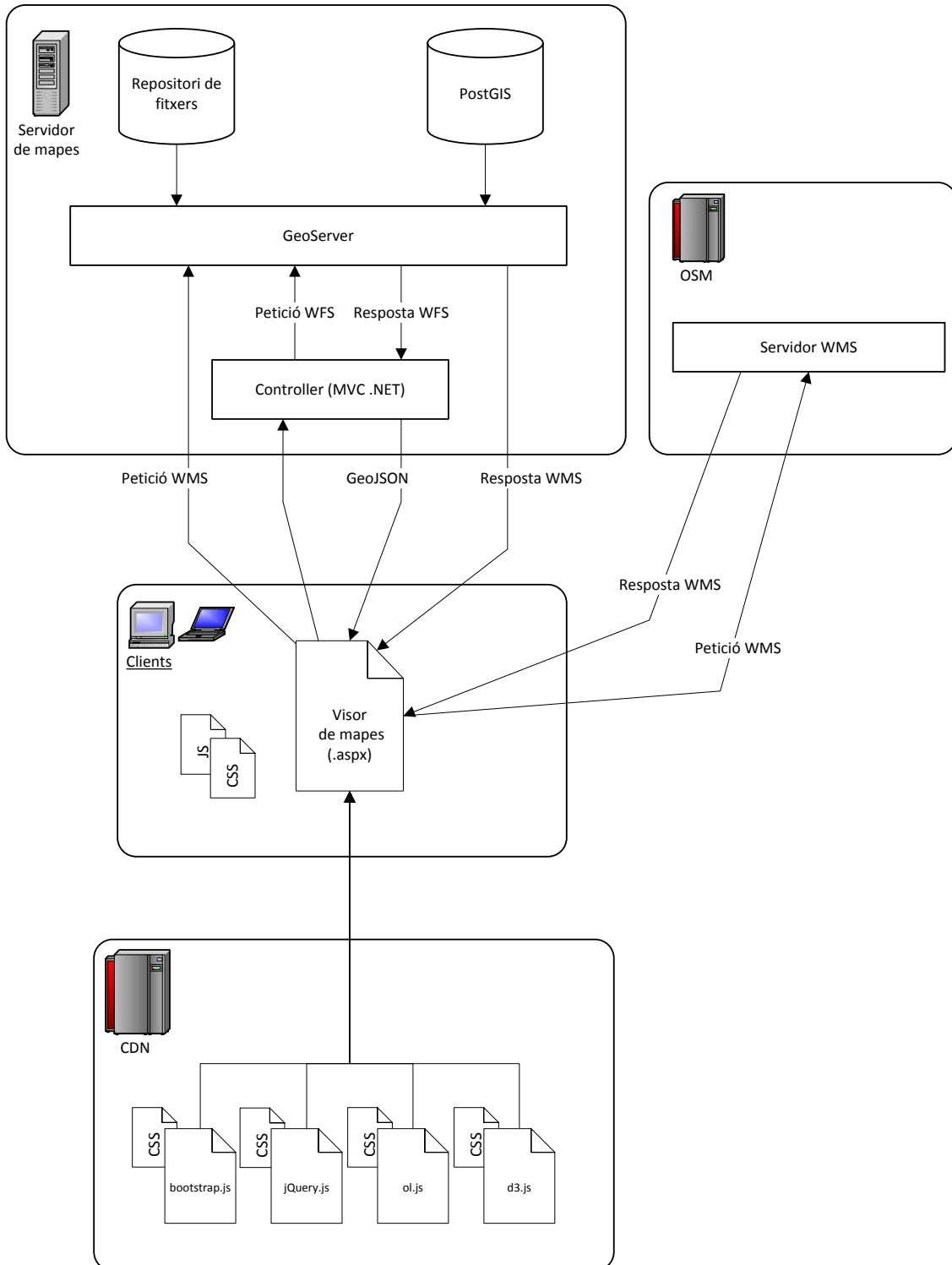


Fig. 27 – Arquitectura de l'aplicació



Els clients accediran mitjançant una connexió HTTP al portal web mitjançant el seu navegador habitual. S'indica que el visor és un fitxer .aspx ja que realment s'emmagatzema en aquest format.

S'inclouen com a part de la vista els fitxers JavaScript i CSS propis de l'aplicació ja que el client hi accedirà directament per a representar-la.

El visor de mapes (la Vista dins el patró MVC) es serveix de diferents serveis per a recopilar tota la informació que s'ha de mostrar:

- D'una banda hi ha les llibreries JavaScript. S'ha optat per fer servir CDN<sup>23</sup> públics en comptes d'allotjar-les al servidor d'aplicacions. D'aquesta manera s'estalvia espai en aquest servidor alhora que s'apropa el contingut a l'usuari ja que aquestes llibreries es serviran des del servidor més proper a aquest. Totes aquestes llibreries poden disposar de fitxers CSS per a decorar els elements que generen. Aquests fitxers CSS també estaran disponible en CDN en els casos que apliqui.
- Per a la capa de base cartogràfica, com ja s'ha esmentat, es decideix fer servir OpenStreetMaps (OSM). Aquesta connexió es farà mitjançant peticions WMS a un dels servidors disponibles. OpenLayers incorpora connexions a OSM de forma nativa, de manera que aquest aspecte queda emmascarat per al programador.
- Per últim hi ha la capa servidor implementada per aquest projecte, que resideix al servidor d'aplicacions. Està compost per:
  - PostGIS: La base de dades espacials que conté la informació de totes les capes a representar en format tabular.
  - Repositori de fitxers: És simplement una carpeta del servidor on s'emmagatzemaran algunes capes en format shapefile o GeoTIFF. És possible que aquest repositori no acabi a la versió definitiva del projecte però en tot cas serà necessària per a realitzar proves.
  - GeoServer: El servidor de mapes que servirà d'enllaç entre les fonts de dades anteriors i l'aplicació web.
  - IIS: El servidor d'aplicacions web que s'ha indicat com a Controlador (dins el patró MVC). Aquest controlador realitza les peticions a GeoServer en el protocol WFS i rep fitxers JSON que llavors trasllada a la vista per a ser representats. S'encarrega també de fer altres tractaments que necessitarà la vista per a mostrar la informació.

---

<sup>23</sup> Content Delivery Network. És una xarxa d'entrega de continguts per a descentralitzar l'accés a contingut estàtic com fitxers JavaScript o CSS.

Finalment hi ha les connexions que realitza directament la vista a GeoServer mitjançant WMS. Aquest tipus de peticions no es faran a través del controlador ja que el resultat ja estarà pretractat per a ser mostrat directament a la vista.

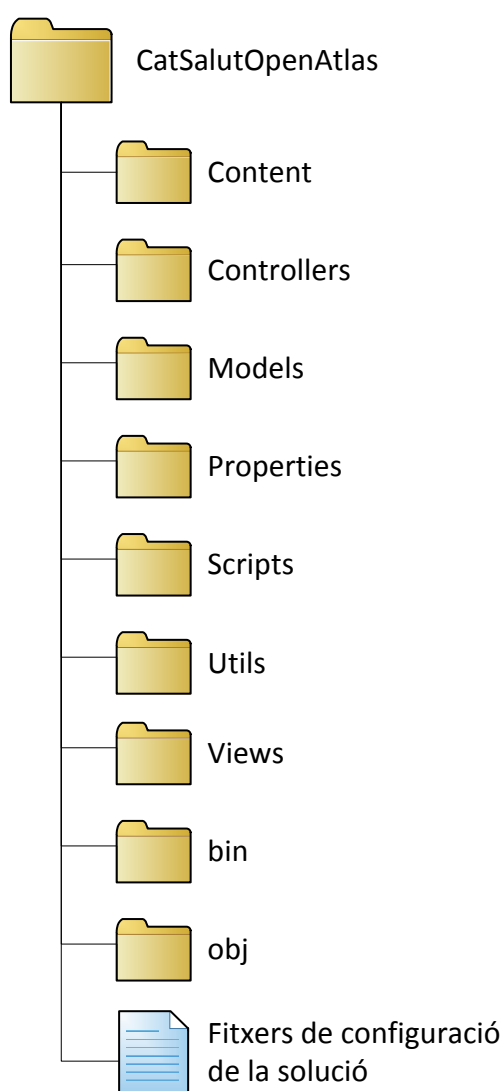
## 7 Implementació del visor web

---

En aquest capítol es detalla la fase d'implementació del visor web, un cop ja parametritzats la resta de serveis.

### 7.1 Estructura de la solució

El projecte (o solució, com és habitual anomenar-lo en l'entorn Visual Studio) s'estructura de la següent manera:



*Fig. 28 – Estructura general de la solució/projecte*

Aquesta estructura de carpetes ve fixada pel *framework*. És possible parametritzar-ne qualsevol aspecte però no és l'objectiu del projecte així que s'ha fet servir la configuració per defecte d'un projecte MVC de .NET.

Com es pot observar existeixen tres carpetes que responen a les tres components del patró:

- **Models:** en aquesta carpeta s'emmagatzemen els fitxers que modelitzen la informació de negoci. Aquests fitxers són classes escrites en C# que defineixen aquesta informació. Són les encarregades de connectar amb els orígens de dades. En aquesta carpeta es pot trobar la classe **Layer** que defineix com són les "Capes" que es visualitzaran:

```
public class Layer
{
    private Global.PROTOCOLS _protocol;
    private Global.LAYERTYPE _layertype;
    private string _name;
    private string _srs;
    private string _json;
    private string _parameter;
    private string _parameterDesc;
    private int _agrupadors;
    private bool _startVisible;
```

Té un conjunt d'atributs membres per a definir:

- El protocol que es fa servir per a obtenir-ne les dades. Pot ser WFS o WMS.
- El tipus de capa que representa. Pot ser una capa de punts o una capa de polígons.
- El nom de la capa.
- El sistema de referència utilitzat.
- La seva representació en format GeoJSON.
- El nom del paràmetre clau que conté els valors que diferenciaran un element d'un altre. És el paràmetre que realment es vol representar.
- El nom del paràmetre que descriu el paràmetre clau. És útil quan el paràmetre a representar és un valor categòric.
- Un valor numèric per a determinar el nombre de grups sobre els quals es volen agrupar els diferents valors del paràmetre clau. Útil per a una representació basada en una escala de colors.
- Un atribut booleà que determina si la capa és visible quan es carrega al mapa. Útil en casos que es carreguin diverses capes alhora.

A més disposa de diversos mètodes per a tractar les dades obtingudes. A continuació es dona la signatura d'aquests mètodes amb la seva documentació:

```
/// <summary>
/// Constructor amb paràmetres
/// </summary>
/// <param name="pProtocol">Protocol</param>
/// <param name="pLayerType">Tipus de capa</param>
/// <param name="pName">Nom de la capa</param>
/// <param name="pSrs">SRS de la capa</param>
/// <param name="pParameter">Paràmetre clau</param>
/// <param name="pParameterDesc">Descripció del paràmetre clau</param>
/// <param name="pAgrupadors">Nombre d'agrupadors</param>
/// <param name="pStartVisible">Visible a l'inici</param>
public Layer(Global.PROTOCOLS pProtocol, Global.LAYERTYPE pLayerType,
string pName, string pSrs, string pParameter, string pParameterDesc, int
pAgrupadors, bool pStartVisible)

/// <summary>
/// Crea una petició al servidor i informa el camp Json amb el resultat
/// </summary>
public void GetGeoJSON()

/// <summary>
/// Genera una URL per a la petició al servidor GeoServer
/// </summary>
/// <returns>La cadena URL de connexió</returns>
private string GenerateRequestUrl()

/// <summary>
/// Calcula el valor màxim d'una propietat del Json d'aquesta capa. Cal
que hi hagi dades a l'atribut Json
/// </summary>
/// <param name="property">Nom de la propietat a calcular</param>
/// <returns>El valor màxim de la propietat</returns>
public Double GetMaxValue(string property)

/// <summary>
/// Calcula el valor mínim d'una propietat del Json d'aquesta capa. Cal
que hi hagi dades a l'atribut Json
/// </summary>
/// <param name="property">Nom de la propietat a calcular</param>
/// <returns>El valor mínim de la propietat</returns>
public Double GetMinValue(string property)

/// <summary>
/// Calcula el valor de separació entre grups. Utilitza el valor màxim,
mínim i el nombre de grups
/// </summary>
/// <returns>Valor numèric del salt entre colors quan es representa un
element</returns>
public double CalculateColorSpacing()
```

No hi ha més classes a la carpeta Models ja que, pel que fa a negoci, no hi haurà més elements en aquesta aplicació.

- **Controllers:** Aquesta carpeta conté els controladors. Aquestes classes estan compostes per mètodes (anomenats "Accions" al *framework* MVC .NET) i cada un d'aquests es correspon a una petició feta des de la Vista. El mètode processa aquesta petició, obté dades del Model i retorna una resposta a la Vista.
- En aquest projecte només hi ha un sol controlador anomenat **Atles**, que s'encarrega de processar totes les peticions de la Vista. Això és així perquè es pot considerar, de forma molt simplista, que aquesta aplicació només té una funció: mostrar capes d'informació.
- A continuació es mostra com a exemple un d'aquests mètodes per a comentar algunes de les particularitats del Controlador:

```
public ActionResult Regions()
{
    List<Layer> layers = new List<Layer>();
    layers.Add(new Layer(Global.PROTOCOLS.WFS, Global.LAYERTYPE.POLYGON,
"RegionsAssegurats", "EPSG:3857", "Assegurats", "Assegurats", 9, true));

    // Valors a passar a la vista
    // Nom de la capa de la qual es vol veure la llegenda
    ViewData["LegendLayer"] = "RegionsAssegurats";
    // Descripció de la vista
    ViewData["layerInfo"] = "Aquesta capa representa les diferents
regions sanitàries amb informació sobre el nombre d'assegurats a cada una.";
    // Cercador al camp...
    ViewData["Cercador"] = null;
    // Html per incloure a la finestra emergent
    ViewData["PopupContent"] = "var info =
'<h1>'+props['regio']+'</h1><br /><p><em>Assegurats en aquesta regió</em>: ' +
props['Assegurats'].toLocaleString() + '</p>';";
    // Títol de la capa per mostrar a la capçalera
    ViewData["layerTitle"] = "Regions sanitàries";
    return View("Map", layers);
}
```

Sense entrar en massa detall es poden destacar alguns elements que intervenen aquí:

- `List<Layer> layers`: Es genera una llista de capes a representar. En l'exemple es crea una corresponent a la capa de regions sanitàries. S'hi indiquen els seus atributs membres per a poder-hi accedir després des de la vista com ara: el tipus de capa, la projecció, el paràmetre clau, el nombre de grups, etc.
- ActionResult: és el tipus de retorn d'aquesta acció. Aquest és el tipus de resposta per defecte i el més habitual ja que correspon a una pàgina

.aspx. Es poden definir altres tipus d'accions com per exemple JsonResult, que retornaria un objecte Json; o FileResult, que retornaria un fitxer.

- ViewData: és un objecte propi del Controlador que serveix per a transferir informació cap a la vista. A l'exemple es veu com es passen algunes cadenes de text que es mostraran en certes parts de la vista. Aquestes cadenes poden ser, fins i tot, porcions de codi JavaScript per a executar-se des del client de manera que es pot tenir diferent codi JavaScript en funció de l'acció executada.
- return View("Map, layers"): retorna un objecte del tipus de retorn ActionResult (com ja s'ha esmentat) que, en aquest cas, prové de la crida al mètode view(). Aquest mètode permet triar quina vista (és a dir, quin fitxer .aspx) es retornarà com a resposta. Si no s'hi indica cap paràmetre es retornarà la vista amb el mateix nom que l'acció. En aquest cas es retornarà la vista "Map" i es passarà la llista de capes a representar. La vista s'encarregarà de representar cada una d'elles.

Per més detalls sobre totes les funcionalitats del *framework* es pot consultar la documentació del mateix, ja que aquí només es descriuran les funcions que s'han fet servir en la implementació.

- **Views**: En darrer lloc es troba la Vista. Aquesta carpeta conté un conjunt de fitxers encarregats de la visualització i ha de tenir la següent estructura:
  - Una subcarpeta per a cada controlador amb el nom d'aquest.
  - Una subcarpeta anomenada "Shared" per a emmagatzemar vistes compartides entre diversos controladors o vistes parcials.
- En aquest cas l'estructura queda de la forma següent:

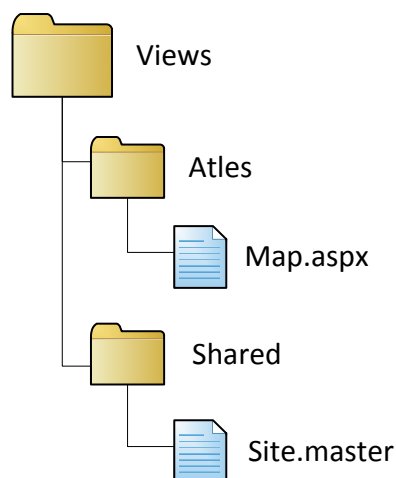


Fig. 29 – Estructura de fitxers de la vista

- Al projecte només es fan servir dos tipus de fitxers:
  - o .aspx: Cada un d'aquests fitxers correspon a una Vista concreta. En el cas del projecte només hi ha una única vista que s'utilitzarà per a representar totes les capes.
  - o .master: Es tracta d'una vista "mestra" que serveix d'esquelet per a inserir-hi altres vistes a dins. En el cas del projecte no seria necessària ja que només hi ha una vista, però és bona pràctica ja que en una propera fase es podrien implementar altres controladors i vistes.
  
- Fent servir aquests dos tipus podem separar clarament aquells objectes que no variaran i escriure'ls només una vegada. A la taula següent es mostra aquesta separació:

<b>Site.master</b>
<p>Declaració del DOCTYPE:</p> <pre>&lt;!DOCTYPE html&gt; &lt;html xmlns="http://www.w3.org/1999/xhtml"&gt;</pre>
<p>Element &lt;head&gt; amb les crides a fitxers CSS i JavaScript:</p> <pre>&lt;head runat="server"&gt;   &lt;meta http-equiv="Content-Type" content="text/html; charset="utf-8"/&gt;   &lt;title&gt;Open Atlas del Sistema de Sanitat Pública de Catalunya&lt;/title&gt;   &lt;link rel="stylesheet" [...]   &lt;script type="text/JavaScript" [...] &lt;/head&gt;</pre>
<p>Parts de la vista que no variaran i que corresponen a la distribució (header, menú, etc)</p> <pre>&lt;body&gt;   &lt;div id="menu"&gt; [...]</pre> <pre>  &lt;div id="main"&gt; [...]</pre>
<p>Elements &lt;asp:ContentPlaceholder&gt;. Aquests elements serveixen per marcar en quin punt s'insertarà el codi de cada vista particular. Es poden definir tants com siguin necessaris. S'han definit dos Place Holders: un per afegir contingut a l'element &lt;head&gt; i un altre per afegir contingut a l'element &lt;body&gt;.</p> <pre>&lt;asp:ContentPlaceholder ID="HeadContent" runat="server" /&gt; &lt;asp:ContentPlaceholder ID="MainContent" runat="server" /&gt;</pre>



## **[nom\_vista].aspx**

Elements `<asp:Content>`. Aquests elements serveixen per a inserir contingut als elements `ContentPlaceHolder` definits a la vista "mestra". Es fa servir l'atribut `ID` per a vincular una amb l'altra.

```
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
[...]
```

En aquests elements s'insereix tota la part variable de la vista.

Un cop vistes les tres carpetes corresponents al Model, Vista i Controlador es resumeixen la resta de carpetes de l'estructura.

- **Content:** Aquesta carpeta conté fitxers necessaris en la visualització. Aquest és l'espai reservat per a emmagatzemar fitxers CSS i imatges. També es podrien emmagatzemar altres tipus de fitxers com ara fitxers de text pla (JSON, XML, etc.) que s'utilitzin directament des de la Vista.
- **Properties:** Aquesta carpeta hi ha fitxers necessaris per a l'execució del *framework* MVC .NET. No s'utilitza en el desenvolupament directament.
- **Scripts:** Aquesta carpeta és la reservada a fitxers de tipus script com ara fitxers JavaScript que no provenguin d'un CDN. Al projecte s'ha creat un fitxer per a emmagatzemar algunes funcions que s'utilitzen a totes les vistes.
- **Utils:** Aquesta carpeta serveix per a emmagatzemar classes C# d'utilitats generals que s'utilitzin arreu de l'aplicació. En aquest cas s'ha inclòs una classe anomenada **Global** que defineix paràmetres generals de l'aplicació com la URL del servidor de mapes i alguns objectes enum que es fan servir en altres classes.
- **bin i obj:** Aquestes carpeta és on apareixen els resultats de la compilació (fitxers .dll i .pdb) i on es desen les llibreries .NET que s'importin a la solució.

## **7.2 La Vista – Càrrega de capes**

En aquest apartat s'entrarà en detall en com s'ha implementat la interacció amb l'usuari, és a dir, la Vista. Com s'ha comentat a l'apartat anterior la vista està composta d'una vista "mestra" i una altra vista de contingut. Si es realitza la petició <http://localhost:1635/Atles/Centres> s'està realitzant una petició al controlador **Atles**

per executar l'acció **Centres**, la qual retornarà una vista que mostra la ubicació de tots els centres del sistema de sanitat pública de Catalunya. Aquí es pot veure el resultat:

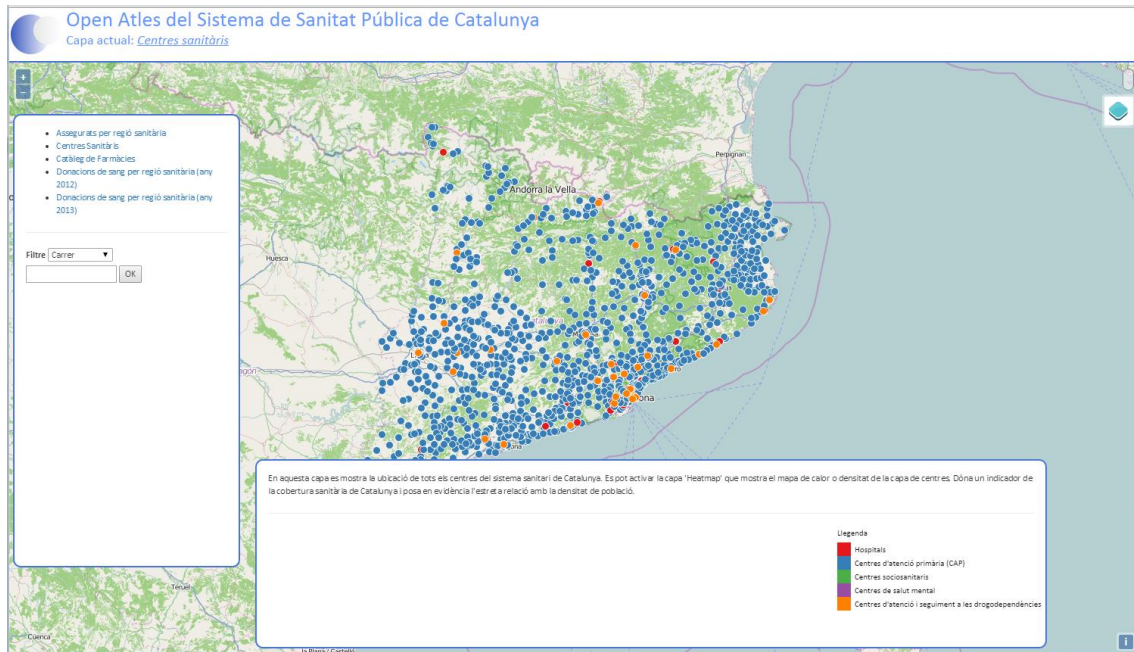


Fig. 30 – Vista Centres

S'observa que la distribució esta formada de diferents panells que contenen informació diversa amb el mapa com a fons. Hi ha certs elements que són propis de la vista resultat de l'acció **Centres** mentre que d'altres provenen de la vista "mestra". A la següent captura es mostren en vermell aquells elements que provenen de la crida a l'acció **Centres**.

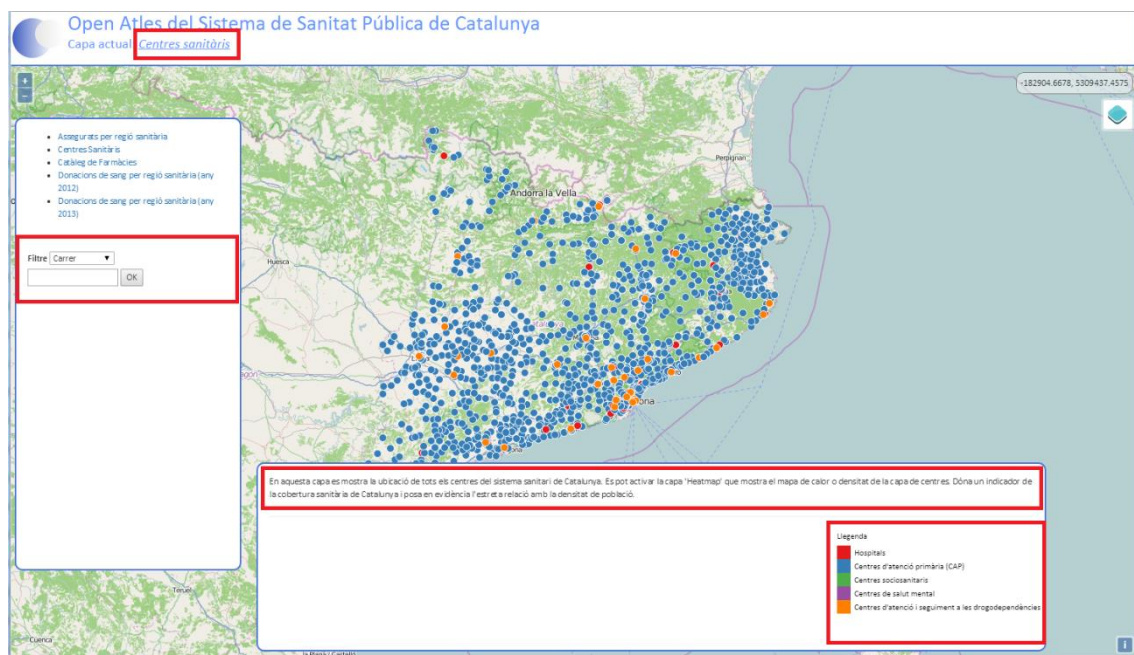


Fig. 31 – Vista Centres – Elements que es defineixen a la vista

En concret:

- Una part de la capçalera que indica el nom de la capa que es visualitza.
- Una part del menú lateral que permet fer cerques en aquesta capa.
- Un paràgraf d'informació sobre la capa al panell inferior.
- Una llegenda al panell inferior.

Els altres elements visible es carreguen a la pàgina “mestra”:

- La capçalera amb el logotip i el títol de l'aplicació.
- La llista d'enllaços del menú lateral.
- La maquetació en objectes `<div>` de tot el visor.

Per a la creació del mapa es parteix d'un bloc `<div>` que es maqueta tal i com es mostra a les figures anteriors fent servir CSS. La resta d'implementació es fa amb les llibreries JavaScript disponibles, principalment amb OpenLayers 3.

La major part de codi JavaScript es troba al fitxer **atles.js** en forma de funcions que són cridades des de la vista.

Per a la creació d'un mapa sobre el `<div>` anterior es realitza la crida a la funció `init()` definida en aquest fitxer. Aquesta funció fa el següent:

```
var projection = ol.proj.get('EPSG:3857');
var view = new ol.View({
  center: [195000, 5100000],
  projection: projection,
  zoom: 8
});
var map = new ol.Map({ target: 'map', view: view });
```

Es crea un objecte `ol.View` per a definir el centre, projecció i zoom del mapa en el moment de la seva creació i a continuació es crea l'objecte `ol.Map`, referenciant-hi l'objecte anterior així com l'identificador del `<div>` on es vol plasmar aquest mapa.

En referència a la projecció cal notar que és necessari que totes les dades siguin consistents. La cartografia base, com es comentava en capítols anteriors, està projectada sobre el EPSG:3857 que correspon a una projecció de Mercator molt habitual en cartografia de visors web com la que es fa servir en el projecte. És per això que totes les dades que s'incorporin al visor han d'estar en aquesta mateixa projecció i s'haurà d'explicitar sempre per a que OpenLayers faci la transformació adient en cas que aquesta no concordi.

Si només es fa això no es veurà res. Hi haurà alguns controls bàsics però el mapa estarà blanc. Cal doncs afegir-hi capes d'informació.

La primera que s'hi afegeix és una de les capes cartogràfiques que serviran de base que, com ja s'ha comentat en capítols anteriors, serà una capa d'OpenStreetMaps:

```
// Capa OSM
var osmSource = new ol.source.OSM();
var osmLayer = new ol.layer.Tile({source: osmSource, title: 'OSM'});
map.addLayer(osmLayer);
```

Es crea un objecte `ol.source.OSM`. Els objectes `ol.source`, dels quals hi ha diversos subtipus, defineixen l'origen de les dades. En aquest cas tot es fa de manera automàtica per OpenLayers ja que les capes d'OSM estan integrades a la llibreria.

Tot seguit es crea l'objecte `ol.layer.Tile` tot passant-li l'origen de dades i donant-li un nom. D'objectes `ol.layer` també hi ha diversos subtipus. En aquest cas es fa servir el tipus `Tile` que generarà una capa basada en imatges "trossejades" de manera que es carreguin només aquelles que siguin visibles en aquell moment. Finalment s'afegeix la capa al mapa i en aquest punt ja seria visible.

A més de la capa d'OpenStreetMaps s'han afegit algunes capes cartogràfiques més, com s'ha descrit al capítol 4. Les capes Stamen i MapQuest estan també integrades en OpenLayers així que la seva activació es realitza de la mateixa manera que amb OSM. Pel que fa a la capa de cartografia de l'ICGC no és així i s'han de definir els paràmetres de forma més manual:

```
var icgcLayer = new ol.layer.Tile({
  source: new ol.source.TileWMS({
    url: 'http://geoserveis.icc.cat/icc_mapesmultibase/noutm/wms/service?',
    params: { 'LAYERS': 'topo' }
  }),
  title: 'ICGC'
});
```

Un cop implementades les capes de cartografia de base i seguint l'exemple de la capa Centres cal carregar les capes d'informació. Totes aquestes capes d'informació vindran d'una petició WFS que el controlador retornarà a la vista en forma de cadena JSON.

Com s'ha comentat anteriorment el controlador passa una llista de capes en un objecte `List<Layer>`. Aquest objecte és accessible des de la vista amb l'objecte `Model`. A continuació es mostra com recórrer aquest objecte i realitzar la importació de cada capa al mapa:

```
// Càrrega de capes
<% foreach (CatSalutOpenAtlas.Models.Layer layer in Model) {
%>
    var json = <%= layer.Json %>;
```

Aquestes línies contenen “scriptlets”. Els scriptlets permeten executar codi C# dins de les pàgines HTML. En primer lloc s’itera l’objecte Model (una llista d’objectes Layer) i per cada una es crea una variable JavaScript anomenada json que contindrà el contingut de l’atribut json de l’objecte layer. S’assumeix que hi haurà una cadena en format JSON vàlid i no es realitza cap comprovació ja que d’això s’encarrega el controlador.

Tot seguit es carreguen la resta de membres de la capa en variables JavaScript per a utilitzar-les en la creació de la capa amb OpenLayers fent la crida a:

```
var vectorlayer = creaCapa(json, map, colorSpacing, min, layerColors, parameter,
title, layertype, startVisible);
```

### 7.3 La Vista – Enriquant el contingut

Un cop carregada una capa sobre la base cartogràfica s’obté un resultat com aquest:

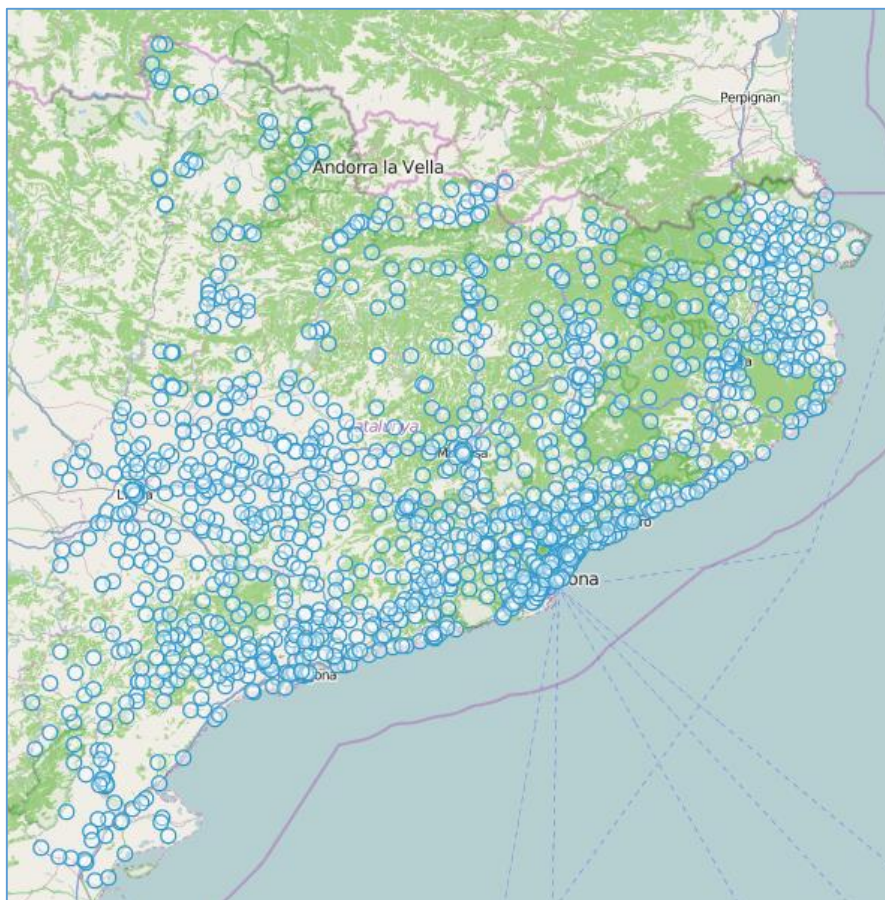


Fig. 32 – Capa de centres carregada per primera vegada

Com es pot veure tots els punts carregats són idèntics i es fa difícil extreure informació. Per a enriquir el contingut es realitzen diverses tasques:

- Realçar els elements representats segons algun criteri. En el cas que s'ha pres com a exemple, la capa de Centres, s'opta per realçar segons el camp que determina el tipus de centre. Per aconseguir això la vista disposa d'algunes variables que provenen del controlador:
  - o Una paleta de colors a fer servir.
  - o Atribut que determinarà el canvi de color. Pot ser un atribut categòric, com en aquest cas, però també pot ser numèric i en aquest cas:
    - Valors mínim i màxim de l'atribut a representar.
    - Valor numèric de l'espaiat entre colors de la paleta. Determina cada quantes unitats de l'atribut numèric es fa un salt de color. S'ha optat per un espaiat equidistant a totes les capes tot i es podria haver fet servir una escala logarítmica en algunes de les capes per a veure millor la diferència entre els elements.
- A continuació s'afegeix interacció als elements representats. En concret es fa servir la llibreria Bootstrap, que incorpora controls de tipus finestra emergent. En aquesta finestra es mostren alguns dels camps més rellevants:

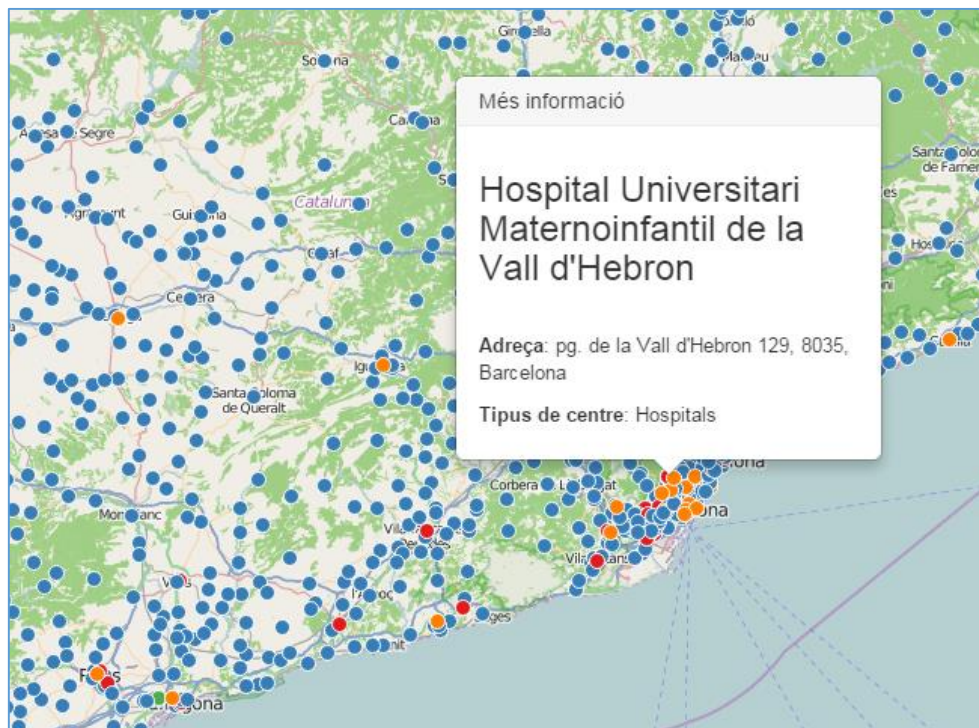


Fig. 33 – Finestres emergents amb Bootstrap

- En aquesta capa (i d'altres) s'incorpora també una funció de cerca. Aquesta està disponible a la part inferior del menú lateral. Aquesta funció permet seleccionar un dels atributs de la capa, per exemple el Nom, l'Adreça o el Codi Postal. L'aplicació crea de forma dinàmica el llistat d'atributs llegint-ho directament de l'objecte GeoJSON fent servir jQuery:

```
var fields = Object.keys(jsondata.features[0].properties);
for (var key in fields){
    $("#selCerca").append("<option
    val="+fields[key]+">" +fields[key]+"</option>");
}

```

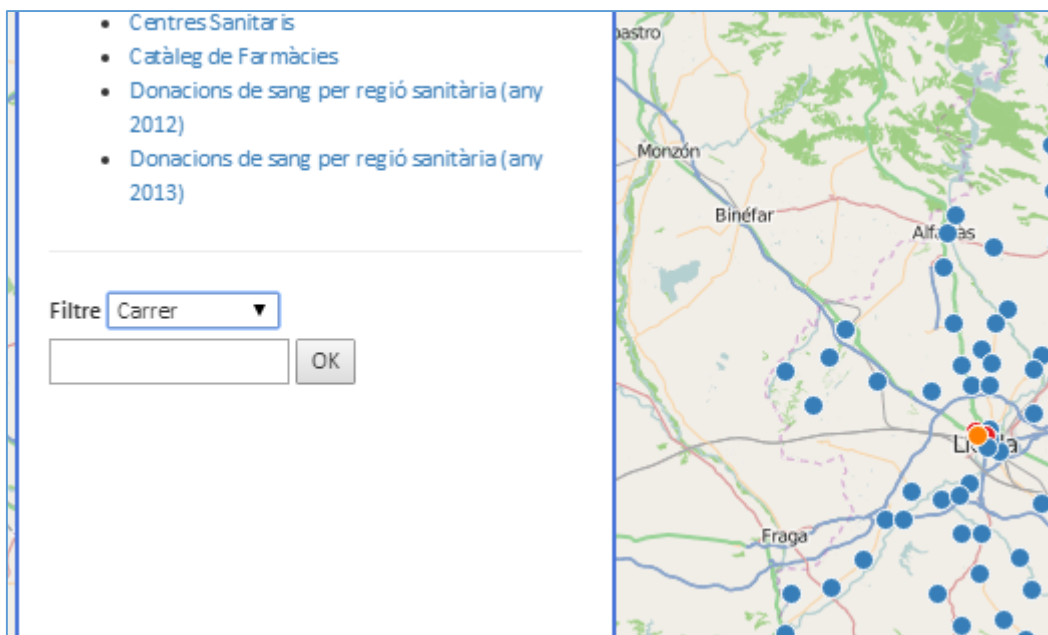


Fig. 34 – Cercador d'elements de la capa

- Finalment s'incorpora informació sobre la capa al panell inferior. En primer lloc hi apareix una descripció de la capa, que ve directament del controlador.
- Tot seguit es genera una llegenda que expliqui els colors que es representen al mapa fent servir la llibreria D3. Aquesta llibreria permet manipular el DOM segons les dades que hi hagi a l'objecte GeoJSON que s'hagi carregat.
- Amb D3 es crea un element <svg> de gràfics vectorials que permet (de la mateixa manera que amb les capes vectorials) treballar de forma matemàtica amb representacions gràfiques. Es creen rectangles que prenen el color dels elements del mapa i s'afegeixen títols a cada un segons el tipus de centre al qual pertanyin.
- En altres capes on l'atribut sigui numèric i no categòric la llegenda es crea de forma similar, però utilitzant els valors de màxim, mínim i espaiat descrits anteriorment.
- A continuació es mostren dos exemples:

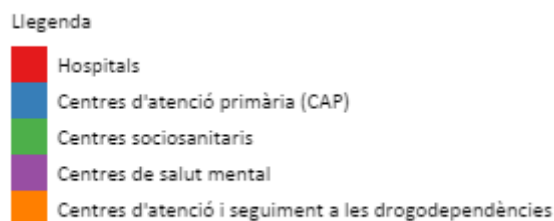


Fig. 35 – Llegenda d'una capa amb atributs categòrics

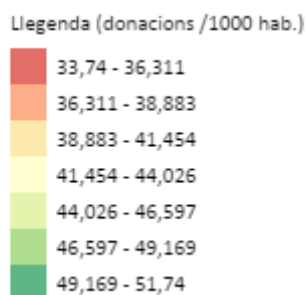


Fig. 36 – Llegenda d'una capa amb atributs numèrics

- En algunes capes on les dades ho permeten s'ha intentat incorporar més elements visuals realitzats en D3, ja que el potencial d'aquesta llibreria és molt gran.
- A continuació es mostra una gràfica de barres generada de forma dinàmica en base a les dades carregades a la capa:

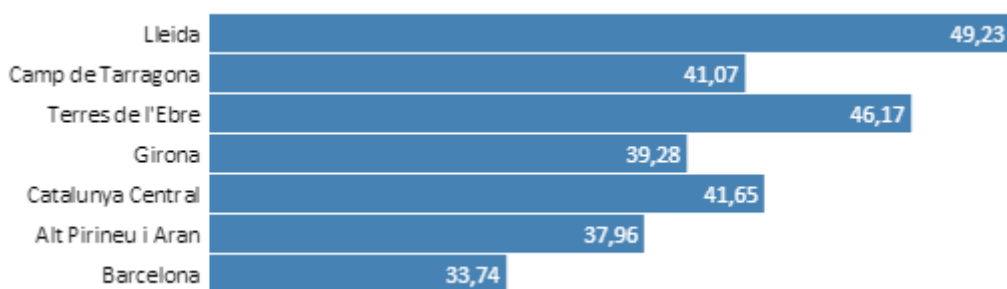


Fig. 37 – Gràfica dinàmica en D3

- D'igual manera que amb la llegenda, aquest gràfic s'ha realitzat creant un element `<svg>` al DOM. En aquest element s'hi creen rectangles, l'amplada dels quals va determinada pel valor que han de representar. Finalment s'hi inclouen els literals adjacents per a donar nom als elements de la sèrie i indicar-hi el seu valor.



## 8 Implementacions futures

---

La versió actual d'aquest Atlas Sanitari és tot just els inicis del que pot ser un projecte més gran.

El visor permet consultar algunes de les fonts de dades més importants, cercar-hi elements en alguna d'elles, navegar-hi i representar-les de diverses maneres.

D'altra banda hi ha un gran nombre de funcionalitats que no han vist la llum encara, en gran part degut a l'ajustat calendari i a la voluntat d'arribar a un mínim de contingut que fos raonable i que servís de *proof-of-concept*, per a posteriorment decidir si el projecte pot seguir endavant.

En una propera versió es preveuen un seguit de funcionalitats noves així com l'aprofundiment en algunes de les existents:

- En els requisits de l'anàlisi funcional es va establir que seria possible exportar la capa visible a un format GeoJSON (com a mínim). Això no ha estat possible però no és una implementació costosa. La informació arriba del servidor de mapes al client en aquest format i està disponible en variables JavaScript, per tant seria possible dur-ho a terme sense gaire esforç. Es preveuen varies opcions:
  - o Crear un control personalitzat d'OpenLayers en forma de botó que permeti la descàrrega de les capes en format GeoJSON.
  - o Expandir el control personalitzat de selecció de capes amb aquesta funcionalitat.
- Una altra opció d'exportació seria en format imatge. La documentació d'OpenLayers dóna un possible enfocament en un dels seus exemples<sup>24</sup>.
- Pel que fa a la visualització de les dades, amb la implementació actual es dóna poca flexibilitat a l'usuari. S'han triat paletes de color que siguin clares a l'hora de diferenciar els diferents elements, però no s'ha tingut en compte que l'usuari pugui tenir disfuncions cromàtiques.
- A l'aplicació, com es comenta al capítol 3, es fa servir ColorBrewer, una petita llibreria que conté un gran nombre de paletes de color. En una propera

---

<sup>24</sup> <http://openlayers.org/en/v3.0.0/examples/export-map.html>.

actualització es podria crear un control que permeti la tria d'altres paletes més enllà de la que apareix per defecte i així actualitzar la visualització de les dades.

- Permetre a l'usuari connectar amb altres servidors WMS. Es demanaria a l'usuari una sèrie de paràmetres i es carregaria la capa a la vista existent.
- El controlador de l'aplicació s'ha implementat escrivint una acció per cada capa (o conjunt de capes) que es vol representar. Aquesta metodologia de programació és molt estàtica i fa que qualsevol incorporació futura de noves capes s'hagi d'escriure al codi font de l'aplicació, compilar-ho i publicar-ho de nou al servidor web.

Si en el futur es volgués mantenir actualitzat el contingut amb una freqüència força elevada s'hauria de pensar en un altre sistema per a definir les capes i els seus paràmetres de visualització.

Es creu que una millor solució podria ser la definició de les capes en una taula de base de dades (es podria fer servir el propi PostGIS) i, mitjançant una interfície d'administració, es realitzessin les actualitzacions adients en aquesta taula.

- Es va contemplar la possibilitat que l'aplicació permetés analitzar el temps mínim de desplaçament a un centre sanitari des d'un punt donat. Aquesta funcionalitat és força complexa i per aquest motiu no s'ha plantejat com a objectiu d'aquesta versió. Es preveuen varies opcions:
  - Incorporar dades sobre la xarxa viària i realitzar els càlculs de ruta més curta i calcular-ne el temps.
  - Fer ús d'alguna API pública com ara la de Google Maps per a calcular-ne les millors rutes.
- Per últim però no menys important, el conjunt de les dades representades no ha estat l'esperat. En una propera versió es treballarien més fonts de dades per a incorporar-les al visor.

## 9 Conclusions

---

El camp de la geografia i la possibilitat d'estudiar el nostre planeta (o d'altres) sempre m'han apassionat. Per aquest motiu, la possibilitat de realitzar el Treball de Fi de Grau sobre tecnologia d'aquest àmbit era un repte personal.

Tot i haver tingut algun contacte previ amb aquest tipus de tecnologia, el camp dels SIG era força desconegut per a mi. Això, però, no ha impedit que l'afrontés amb moltes ganes d'aprendre.

Considero que s'ha arribat a fites importants: la implementació obtinguda funciona i té potencial, he conegut l'existència de llibreries per al desenvolupament web que m'eren desconegudes i les puc aplicar en aquest i altres projectes, i he realitzat tractament de dades provinents de fonts obertes del govern.

D'altra banda cal destacar també tot allò que podria millorar. Al capítol 8 es destaquen algunes de les mancances que considero que haurien d'haver-se incorporat en aquesta versió i que finalment no han pogut veure la llum. Tot i haver previst a la planificació inicial que hi hauria mancances pel que fa a temps disponible, no ha estat possible evitar-ho. La vida laboral així com la inversió de temps en altres assignatures han minvat molt el temps de dedicació a aquest projecte i ha calgut afrontar-ho obtenint temps d'on sigui i reduint, en la mesura del possible, l'abast del projecte.

La sensació final, doncs, és de satisfacció per la feina feta però amb una mica de recança pel fet d'haver hagut d'obviar alguns objectius.

## 10 Glossari

---

**SIG:** Acrònim que correspon a Sistemes d'Informació Geogràfica. Es descriu en detall al capítol 2.

**GIS:** Acrònim que correspon a Geographic Information Systems. Veure [SIG](#).

**Ràster:** Model de representació de dades utilitzat en SIG i que es basa en imatges. Es descriu en detall al capítol 2.

**Vectorial:** Model de representació de dades utilitzat en SIG i que es basa en vectors. Es descriu en detall al capítol 2.

**OSM:** Acrònim que correspon a Open Street Maps. És un servei de visualització de mapes en línia de codi obert i col·laboratiu molt utilitzat en la creació de visors web.

**Framework:** Estructura ben definida basada en mòduls de software que serveix com a base per a noves implementacions de programari.

**Plugin:** Petita peça de software implementada exclusivament per a un software més gran que en fa ús.

**Stamen:** És una capa de representació de mapes OSM. Més informació a <http://maps.stamen.com/>.

**MapQuest:** És una capa de representació de mapes OSM. Més informació a <http://www.mapquest.com/>.

**JSON:** Acrònim que correspon a *JavaScript Object Notation*. És un format de fitxers per a l'intercanvi de dades basats en parelles clau-valor integrat en JavaScript que elimina la necessitat de la notació XML per aquesta finalitat.

**GeoJSON:** Basat en JSON, és un format estàndard per a l'intercanvi de dades espacials. Aquest format és suportat en infinitat de software SIG com ara OpenLayers, Leaflet, GeoServer o PostGIS.

DOM: Acrònim que correspon a *Document Object Model*. És la interfície de programació que permet l'accés mitjançant objectes als elements d'un fitxer HTML o XML per a la seva consulta o manipulació.

D3: Llibreria JavaScript per a la representació de dades utilitzada en aquest projecte. Més informació a <http://d3js.org/>.

OpenLayers: Llibreria JavaScript per a la creació de visors de mapes web utilitzada en aquest projecte. Més informació a <http://openlayers.org/>.

Leaflet: Llibreria JavaScript per a la creació de visors de mapes web estudiada en aquest projecte. Més informació a <http://leafletjs.com/>.

SVG: Acrònim que correspon a *Scalable Vector Graphics*. Format gràfic vectorial recomanat pel W3C i suportat pels principals navegadors. Veure Vectorial.

# 11 Bibliografia

---

- [1] Project Management Institute, Inc., Guia del PMBOK Ed. 4, Newtown Square, Pennsylvania: Project Management Institute, Inc., 2008.
- [2] «Map Server,» [En línia]. Available: <http://mapserver.org/>. [Últim accés: Maig 2015].
- [3] «Leaflet JS,» [En línia]. Available: <http://leafletjs.com/>. [Últim accés: Maig 2015].
- [4] «Open Layers,» [En línia]. Available: <http://openlayers.org/>. [Últim accés: Maig 2015].
- [5] «GeoServer,» [En línia]. Available: <http://geoserver.org/>. [Últim accés: Abril 2015].
- [6] Institut Cartogràfic i Geològic de Catalunya, «Institut Cartogràfic i Geològic de Catalunya,» [En línia]. Available: <http://www.icc.cat/>. [Últim accés: Maig 2015].
- [7] Generalitat de Catalunya, «Dades obertes de salut,» [En línia]. Available: [http://observatorisalut.gencat.cat/ca/demanar\\_dades/Dades-sanitaries/dades\\_obertes\\_salut/](http://observatorisalut.gencat.cat/ca/demanar_dades/Dades-sanitaries/dades_obertes_salut/). [Últim accés: Abril 2015].
- [8] L. M. Royo Pérez, «iniSIG,» [En línia]. Available: <http://inisig.com/postgis-geoserver-y-leaflet-ii/>. [Últim accés: Maig 2015].
- [9] M. Schmandt, «GIS Commons,» [En línia]. Available: <http://giscommons.org/>. [Últim accés: Març 2015].
- [10] Generalitat de Catalunya, «Regions Sanitàries,» [En línia]. Available: <http://catsalut.gencat.cat/ca/catsalut-territori/regions-sanitaries/>. [Últim accés: Abril 2015].
- [11] «D3 - Data Driven Documents,» [En línia]. Available: <http://d3js.org/>. [Últim accés: Maig 2015].
- [12] «Spatial Reference,» [En línia]. Available: <http://spatialreference.org/>. [Últim accés: Abril-Maig 2015].
- [13] «jQuery,» [En línia]. Available: <https://jquery.com/>. [Últim accés: Maig 2015].
- [14] «Color Brewer 2,» [En línia]. Available: <http://colorbrewer2.org/>. [Últim accés: Maig 2015].
- [15] «JSON .NET,» [En línia]. Available: <http://www.newtonsoft.com/json>. [Últim accés: Maig 2015].
- [16] «ASP .NET MVC,» Microsoft Corporation, [En línia]. Available: <http://www.asp.net/mvc>.

## 12 Annexos

---

En aquest capítol s'inclouria normalment el codi font de la solució implementada. S'ha optat per publicar-ho de forma oberta a GitHub per a la seva consulta. Es pot accedir al repositori seguint el següent enllaç:

<https://github.com/GabrielGMG/CatSalutOpenAtlas/>