



Treball fi de carrera

ENGINYERIA TÈCNICA EN  
INFORMÀTICA DE SISTEMES

Facultat de Matemàtiques  
Universitat de Barcelona

---

VISUALITZADOR DE MAPES WEB MAP SERVICE  
CROSSBROWSER EN JAVASCRIPT

---

Frederic Salgado Luque

Director: Sergio Escalera Guerrero

Realitzat a: Departament de Matemàtica  
Aplicada i Anàlisi. UB

Barcelona, 18 de febrer de 2009

## ***AGRAÏMENTS***

Voldria agrair a Slash-Solutions S.L. i més concretament als seus representants Ricard Gil Blasco i Miquel French Gonzalez, al cap del projecte del SICT/JP Xavier Santana Carceller i al tutor del TFC dins de l'empresa Miguel García Ortega pel suport i per permetrem presentar la planimetria desenvolupada en el SICT/JP com a TFC  
(Treball Final de Carrera).

També voldria agrair al Tutor del Projecte per part de la Universitat de Barcelona, Sergio Escalera Guerrero, pel suport i seguiment que ha realitzat del TFC.

## ***RESUM***

Aquesta documentació mostra el procés d'elaboració d'un sistema gestor de mapes, que hem anomenat visualitzador, programat en Javascript i que serà integrat en les aplicacions de Slash Solutions que requereixin d'un sistema de representació d'objectes sobre un mapa.

El document mostra l'especificació utilitzada, així com les diferents decisions de disseny que s'han dut a terme per tal de poder solucionar el problema de la manera més eficient.

El document mostrà diferents captures del visualitzador integrat i en funcionament. També disposem d'un annex amb el codi integrat en un Jboss per tal de poder executar-lo.

## ***RESUMEN***

Esta documentación muestra el proceso de elaboración de un gestor de mapas, al que hemos llamado visualizador de mapas, programado en JavaScript i que será integrado en las aplicaciones de Slash Solutions que requieren de un sistema de representación de objetos sobre un mapa.

El Documento muestra la especificación utilizada, así como las diferentes decisiones de diseño que se han llevado a cabo para poder solucionar el problema con la máxima eficiencia.

El documento también muestra diferentes capturas del componente integrado i en funcionamiento. También disponemos en un anexo que contiene el código integrado en un Jboss para poder ser ejecutado.

## ***ABSTRACT***

This project describes the design of a map controller, called map visualizer, programmed in Javascript. This software will be integrated in Slash Solutions applications which require from a map object representation procedure.

This documentation describes the used specification, and the design decisions taken into account in order to solve the defined problem with the highest reliability.

This text also shows different captures from the running integrated component. There is also an appendix including the project code integrated in a Jboss in order to be executed.

# ÍNDIX

|                            |          |           |
|----------------------------|----------|-----------|
| <b>INTRODUCCIÓ</b>         | <b>1</b> | <b>9</b>  |
| Motivació                  | 1.1      | 9         |
| Context                    | 1.2      | 9         |
| Estructura de la memòria   | 1.3      | 9         |
| <b>ANÀLISI</b>             | <b>2</b> | <b>10</b> |
| Problema                   | 2.1      | 10        |
| Objectius                  | 2.2      | 10        |
| Requeriments               | 2.3      | 10        |
| Requeriments funcionals    | 2.3.1    | 10        |
| Requeriments no funcionals | 2.3.2    | 11        |
| Casos d'ús                 | 2.4      | 11        |
| Diagrames                  | 2.5      | 16        |
| Diagrama de casos d'ús     | 2.5.1    | 16        |
| Diagrama Entitat - Relació | 2.5.2    | 17        |
| Tecnologies investigades   | 2.6      | 18        |
| WMS                        | 2.6.1    | 18        |
| GetCapabilities, Crida     | 2.6.1.1  | 19        |
| GetCapabilities, Resposta  | 2.6.1.2  | 19        |
| GetMap                     | 2.6.1.3  | 20        |
| GetMap, Crida              | 2.6.1.4  | 20        |
| GetMap, Resposta           | 2.6.1.5  | 22        |
| GetFeatureInfo             | 2.6.1.6  | 23        |
| GetFeatureInfo, crida      | 2.6.1.7  | 23        |
| GetFeatureInfo, Resposta   | 2.6.1.8  | 24        |
| Google Maps i Google Earth | 2.6.2    | 24        |
| API Google Maps            | 2.6.2.1  | 24        |
| Càrrega de la API          | 2.6.2.2  | 25        |
| Elements DOM del mapa      | 2.6.2.3  | 25        |
| Esdeveniments del mapa     | 2.6.2.4  | 25        |

|   |            |           |
|---|------------|-----------|
| Controls del mapa                       | 2.6.2.5    | 26        |
| Crear un POI                            | 2.6.2.6    | 26        |
| Generació de la API KEY                 | 2.6.2.7    | 27        |
| <b>Plataforma</b>                       | <b>2.7</b> | <b>27</b> |
| Javascript                              | 2.7.1      | 27        |
| Història                                | 2.7.1.1    | 27        |
| Sintaxi                                 | 2.7.1.2    | 28        |
| HTML                                    | 2.7.2      | 28        |
| Història                                | 2.7.2.1    | 29        |
| Elements bàsics                         | 2.7.2.2    | 29        |
| <b>Solució Proposada</b>                | <b>2.8</b> | <b>30</b> |
| <b>Planificació</b>                     | <b>2.9</b> | <b>30</b> |
| Temps de realització del projecte       | 2.9.1      | 30        |
| Valoració del cost del projecte         | 2.9.2      | 30        |
| <b>Disseny</b>                          | <b>3</b>   | <b>32</b> |
| <b>Disseny gràfic del visualitzador</b> | <b>3.1</b> | <b>32</b> |
| Barra d'eines                           | 3.1.1      | 32        |
| Eines Bàsiques                          | 3.1.1.1    | 32        |
| Eines no bàsiques                       | 3.1.1.2    | 32        |
| Menú lateral                            | 3.1.2      | 33        |
| Mapa de referència                      | 3.1.2.1    | 33        |
| Servidors WMS                           | 3.1.2.2    | 33        |
| Capes                                   | 3.1.2.2    | 33        |
| Menú inferior                           | 3.1.3      | 34        |
| Punts d'interès                         | 3.1.4      | 34        |
| Informació d'un punt d'interès          | 3.1.5      | 34        |
| Captura rectangular                     | 3.1.6      | 35        |
| Captura circular                        | 3.1.7      | 35        |
| <b>Diagrames</b>                        | <b>3.2</b> | <b>36</b> |
| Diagrama de Classes                     | 3.2.1      | 36        |
| Diagrames de Seqüència                  | 3.2.2      | 37        |
| Iniciar el mapa                         | 3.2.2.1    | 37        |
| Crear una captura rectangular           | 3.2.2.2    | 38        |

|   |            |           |
|---|------------|-----------|
| <b>Implementació</b>                                    | <b>4</b>   | <b>39</b> |
| Problema transformar una posició GPS píxel              | 4.1        | 39        |
| Problema transformar un píxel a una posició GPS         | 4.2        | 39        |
| Problema al realitzar una captura circular              | 4.3        | 40        |
| Problema per obtenir les imatges                        | 4.4        | 41        |
| Superposició de POI's                                   | 4.5        | 41        |
| Fitxer servers.xml                                      | 4.6        | 42        |
| Integració de Google Maps dins la informació dels POI's | 4.7        | 42        |
| Llibreries  | 4.8        | 43        |
| Integració del visualitzador en el SICT/JP              | 4.9        | 43        |
| Mòdul d'emplaçaments                                    | 4.9.1      | 43        |
| Mòdul de processos                                      | 4.9.2      | 44        |
| Mòdul de consultes                                      | 4.9.3      | 45        |
| Google Maps vs. Visualitzador de mapes                  | 4.9.4      | 46        |
| <b>Conclusions</b>                                      | <b>5</b>   | <b>47</b> |
| <b>Resultats</b>  | <b>5.1</b> | <b>47</b> |
| Inserir un nombre elevat de POI's                       | 5.1.1      | 47        |
| Test de Navegadors Web                                  | 5.1.2      | 47        |
| <b>Línies futures</b>                                   | <b>5.2</b> | <b>49</b> |
| WFS   | 5.2.1      | 49        |
| Operacions  | 5.2.1.1    | 49        |
| <b>Bibliografia</b>                                     | <b>6</b>   | <b>50</b> |
| <b>Annex</b>  | <b>7</b>   | <b>51</b> |
| <b>Vocabulari i Sigles</b>                              | <b>7.1</b> | <b>51</b> |
| <b>CD-ROM</b>   | <b>7.2</b> | <b>54</b> |
| Jboss   | 7.2.1      | 54        |
| Instal·lació del Jboss                                  | 7.2.1.1    | 54        |

|                                    |            |           |
|------------------------------------|------------|-----------|
| Execució del Jboss                 | 7.2.1.2    | 54        |
| Codi                               | 7.2.2      | 54        |
| Documentació                       | 7.2.3      | 54        |
| <b>Llistat de servidors WMS</b>    | <b>7.3</b> | <b>55</b> |
| <b>Implementació</b>               | <b>7.4</b> | <b>56</b> |
| Fitxer de Capabilities XML         | 7.4.1      | 56        |
| Funcions de conversió de UTM a GPS | 7.4.2      | 62        |
| Funcions de conversió de GPS a UTM | 7.4.3      | 63        |

# INTRODUCCIÓ

1

## Motivació

1.1

La raó que m'ha portat a desenvolupar aquest projecte neix d'un mòdul que era necessari implementar en una aplicació de l'empresa on treballa actualment. El mòdul consistia en l'elaboració d'un gestor de mapes similar a Google Maps, però gratuït i que utilitzés els mapes del IGN o del Cadastre.

La idea és aconseguir un visualitzador de mapes totalment funcional en la majoria de navegadors existents (Mozilla Firefox, Internet Explorer, Safari, Opera, Chrome,...) per tal que qualsevol usuari-client que utilitzi les aplicacions de Slash Solutions S.L., pugui utilitzar les funcionalitats que aportarà el mapa desenvolupat, sense que el navegador o l'aplicació en sí suposi una barrera pel seu perfecte funcionament.

## Context

1.2

Actualment existeixen diferents aplicacions i sistemes per poder visualitzar informació de mapes en línia. Destaquen per la seva popularitat les aplicacions de *Google*, Google Maps i Google Earth.

Google Maps és la solució via Interfície web, és a dir, la que més s'aproxima a la solució que caldrà implementar i Google Earth que correspon a la solució en aplicació client.

En aquests moments Yahoo, un dels competidors de Google, ha desenvolupat el seu sistema de mapes online, Yahoo mapes molt de l'estil de Google Maps, però encara amb menys popularitat que el citat anteriorment.

Cal destacar els visualitzadors de mapes WMS (Web Map Service) existents, que tot i que no disposen de la mateixa popularitat que els citats anteriorment, són bastant funcionals i usables ( propietat que marca quan una cosa és fàcil d'utilitzar per l'usuari).

## Estructura de la memòria

1.3

La memòria esta estructurada amb les fases bàsiques d'un Projecte: Anàlisi, Disseny, Implementació i Testeig(Prova) *veure figura 1*.

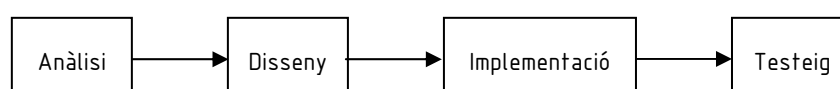
**Anàlisi:** En aquest apartat trobarem l'estudi de quin problema se'ns planteja, quines són les possibles vies per tal de solucionar-lo, quines seran les tecnologies que solucionaran el problema amb més eficiència, així com quina serà la solució lògica escollida.

Dins d'aquest apartat també trobarem la presa de requeriments que s'haurà realitzat amb diferents reunions amb el client.

**Disseny:** En aquest punt, un cop fet l'anàlisi del problema i emmarcats els punts conflictius, procedirem a buscar la solució tècnica al problema, és a dir, el disseny del codi que resoldrà el problema plantejat pel client.

**Implementació:** En aquesta fase, un cop disposem del disseny final, només caldrà crear el codi seguint com a patró el disseny assolit a la fase anterior.

**Testeig:** Una vegada disposem del codi, només caldrà provar-lo per tal de trobar els possibles *bugs* (errors de codi que provoquen un mal funcionament de l'aplicació), així com també es realitzaran els diferents tests de funcionalitat i *usability* (usabilitat de l'aplicació desenvolupada)



*Figura 1:* El gràfic mostra el procés d'elaboració del projecte amb les fases detallades anteriorment.



# ANÀLISI

2

## Problema

2.1

Slash Solutions ha hagut que desenvolupar una aplicació pel **MITyC** (*Ministerio de Industria Turismo y Comercio*), el **SICT/JP** (*Sistema de Información de Comprobación Técnica para las Efectores Provinciales*) que interactua amb Sistemes Centrals (RECOM i RENAPE) i amb **SAM** (*Sistema Automático de Medidas*).

SAM és un conjunt de hardware mòbil dissenyat per tal de poder realitzar captures de Ràdio, Televisió, Telefonía a camp obert, tractar i adequar aquestes captures per tal de realitzar estudis de les dades adquirides.

SICT/JP funciona en Interfície Web, més concretament i per contracte, en Internet Explorer 6.0 i com es comenta més amunt, és una interfície de comunicació entre dos sistemes, SAM (la captura de dades) i Sistemes Centrals (on el Ministeri desa i valida les dades). Aquesta interfície permet la gestió, modificació i explotació de les dades.

En ser la Terra esfèrica cal disposar d'un sistema de projeccions per tal de passar de l'esfera al pla. El problema és encara més gran doncs la terra no és perfectament esfèrica. Per això necessitem un sistema cartogràfic, per tal de poder representar la superfície del planeta (en el cas concret de l'aplicació que ens ocuparà, només haurem de representar el territori espanyol), capaç de ser integrat en una plana web i capaç de mostrar i representar ubicacions.

Cal dir que aquest problema existeix des de temps molt llunyans i que a hores d'ara ha estat solucionat amb força solvència.

Un cop solucionat el problema de com plasmar la Terra sobre un pla, ens cal integrar aquesta representació dins del SICT/JP per tal d'aconseguir mostrar punts de mesura realitzats, estacions, transmissors,...

## Objectius

2.2

Crear un visualitzador de mapes integrable a totes les aplicacions que funcionin via web independentment del navegador que es pugui utilitzar, el que en anglès es coneix com a Crossbrowser: i que sigui integrable en qualsevol de les aplicacions de l'empresa. El funcionament d'aquestes ha de poder ser via interfície web incorporant, a més, les funcionalitats dels *nomenclàtors* existents.

## Requeriments

2.3

### Requeriments funcionals

2.3.1

Requeriments Generals:

- Disponibilitat online de la planimetria. Això implicarà que l'aplicació on estigui integrat ell mapa, haurà de disposar de connexió a Internet.
- Possibilitat de combinar múltiples mapes. El que significa que haurem de ser capaços d'implementar un visualitzador capaç de treballar en múltiples servidors de mapes, i també, intentant buscar al màxim un sistema prou comú i estès possible.
- Configuració d'accés a dades WMS. És un requisit derivat de l'anterior, amb el que ens marca quin haurà de ser el protocol comú dels mapes que es podran utilitzar.

Requeriments del visualitzador:

- Visualitzador de mapes WMS modificable. El mapa ens haurà de facilitar d'una manera senzilla el canvi de mapa, sense que suposi un sobreesforç per l'usuari que utilitzi l'aplicació.
- Visualitzador de mapes multicapa. Els diferents servidors de mapes treballen en multicapa., Haurem de ser capaços de canviar de capes en qualsevol moment.
- Control total de POI-Mapa. Tenir present el moviment del mapa amb el què representa. A més s'haurà de re-ubicar els POI's a la nova situació.
- Caldrà disposar d'un segon mapa (que anomenarem minimapa) que mostrarà l'àrea que s'està visualitzant en el mapa principal.
- Visualització i edició de POI's (crear, moure i esborrar).
- Controls bàsics del mapa:
  - Moure el mapa: El visualitzador de mapes haurà de ser capaç d'implementar la funcionalitat de moure la imatge del mapa, i recarregar la nova posició.
  - Zoom In: Caldrà desenvolupar la funcionalitat que permeti al usuari poder acostar-se al mapa.
  - Zoom Out: Caldrà realitzar el moviment invers al anterior, l'usuari haurà de disposar d'una eina que li permeti allunyar-se del mapa.
  - Realitzar una captura de Zoom: L'usuari hauria de poder disposar d'una eina que li permeti realitzar una selecció d'una zona del mapa i aplicar-hi el màxim zoom possible, mostrant tota l'àrea seleccionada.
  - Realitzar captures d'àrea: Pel tipus de dades que s'utilitzen en el SICT/JP caldrà realitzar diferents captures d'àrea per tal de seleccionar punts i realitzar diferents estudis.
  - Mostrar la informació referent als punts que es mostren al mapa: És necessari mostrar la informació de cada punt, amb un element informatiu que visualitzi aquestes dades.
- Enllaçar els punts del mapa a Google Maps: Si és possible seria interessant poder mostrar un punt seleccionat dins de Google Maps.

Requeriments de la WebApp:

- Integrar el visor en el client.

## Requeriments no funcionals

### 2.3.2

- Utilització d'estàndards OGC (WMS) en el client. Cal utilitzar aquest estàndard degut a que des del client se'ns demana utilitzar els mapes del IGN o del cadastre.
- El visor ha de ser Crossbrowser. Això vol dir que el visualitzador ha de ser funcional en qualsevol navegador web.
- Client lleuger on només hagi de carregar les imatges i gestionar les funcionalitats amb la ràpides.
- Intentar evitar els popups, amb el que el mapa ha d'estar integrat dins del web-site.
- Cal que el mòdul sigui el més independent possible de l'aplicació on corre per tal de convertir-se en un estàndard per les altres aplicacions de l'empresa.

## Casos d'ús

### 2.4

Actors:

- Usuari: Serà el que interactuarà amb l'aplicació, el que realitzarà els diferents moviments que ens subministrarà el mòdul.
- Servidor de Mapes: Considerarem el servidor de mapes com un actor, que ens subministrarà totes

les captures de mapes que l'usuari o el sistema li sol·licitin.

Escenari:

- Interfície web de l'aplicació: Donat que el mapa pot ser integrat en múltiples programes que funcionin en interfície web, l'escenari sempre serà el navegador web (Mozilla Firefox, Internet Explorer,...)

### **Cas d'ús Iniciar el visualitzador.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa i el servidor de mapes.

**Pre-condicions:** L'aplicació on està integrat el mapa haurà d'estar en funcionament, haurem de disposar de connexió a Internet i el servidor de mapes haurà d'estar operatiu.

**Post-condicions:** El visor s'haurà carregat correctament.

**Descripció:** L'usuari iniciarà un mòdul on aparegui o s'utilitzi el visor de mapes. Això significa que es carregarà la configuració que més convingui al usuari. Es mostrarà el WMS per defecte, es carregarà el menú amb les opcions que pertoquin a cada mòdul i finalment el Servidor de mapes ens subministrarà la captura inicial del mapa.

### **Cas d'ús Iniciar el visualitzador en una zona concreta del mapa.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa i el servidor de mapes.

**Pre-condicions:** L'aplicació on està integrat el mapa haurà d'estar en funcionament, haurem de disposar de connexió a Internet i el servidor de mapes haurà d'estar en funcionament

**Post-condicions:** El visor s'haurà carregat correctament.

**Descripció:** L'usuari iniciarà un mòdul on aparegui o s'utilitzi el visor de mapes. Això significa que es carregarà la configuració que més convingui al usuari. Es mostrarà el WMS per defecte, es carregarà el menú amb les opcions que pertoquin a cada mòdul i finalment es mostrarà la captura del mapa en una zona concreta, una província, una comarca, una ciutat,...

### **Cas d'ús Moure el mapa.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa i el servidor de mapes.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament, que seguim disposant de connexió a Internet i que el servidor de mapes també segueixi operatiu.

**Post-condicions:** El mapa s'haurà mogut a la nova posició.

**Descripció:** Un cop inicialitzat el mapa, la configuració inicial que es carregarà per defecte és la de moure el mapa, tot i que l'usuari en tot moment podrà seleccionar l'opció del menú. L'usuari mitjançant un moviment de *drag & drop* arrossegà el mapa per tal de poder visualitzar una zona desitjada. Al igual com es faria en els Sistemes Operatius (en el moviment d'arrossegament d'arxius), l'usuari clicarà sobre el mapa i el desplaçarà cap a on més desitgi o convingui per tal de poder veure noves extensions del mapa. Un cop finalitzat aquest moviment el servidor de mapes subministrarà la nova captura del mapa.

### **Cas d'ús Zoom in.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa i el servidor de mapes.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament, que seguim disposant de connexió a Internet i que el servidor de mapes també segueixi operatiu.

**Post-condicions:** El mapa passarà a la nova captura, més ampliació, el que és el mateix, més endins de la zona que s'està visualitzant abans de realitzar l'acció.

**Descripció:** L'usuari seleccionarà la icona del menú designada a realitzar la funcionalitat de zoom in (captures que s'endinsen cap el mapa). Un cop seleccionada aquesta opció, el component realitzarà la crida al servidor de mapes, per tal d'obtenir la nova captura sol·licitada.

### **Cas d'ús Zoom out.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa i el servidor de mapes.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament, que seguim disposant de connexió a Internet i que el servidor de mapes també segueixi operatiu.

**Post-condicions:** El mapa passarà a la nova captura, menys ampliació, més allunyada de la zona que s'està visualitzant abans de realitzar l'acció.

**Descripció:** L'usuari seleccionarà la icona del menú designada a realitzar la funcionalitat de zoom out (captures que s'allunyen del mapa). Un cop seleccionada aquesta opció, el component realitzarà la crida al servidor de mapes, per tal d'obtenir la nova captura sol·licitada.

### **Cas d'ús Tornar a la posició inicial.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa i el servidor de mapes.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament, que seguim disposant de connexió a Internet i que el servidor de mapes també segueixi operatiu.

**Post-condicions:** El mapa retornarà a la posició de visualització per defecte.

**Descripció:** L'usuari seleccionarà la icona del menú designada a realitzar la funcionalitat de PAN (El component retornarà a la captura inicial per defecte). Un cop seleccionada aquesta opció el component farà la mateixa crida que esta programada per defecte, per tal de reiniciar la posició i obtenir la nova captura del mapa.

### **Cas d'ús Carregar punts al mapa.**

**Escenari:** Interfície web de l'aplicació

**Actors:** Usuari que interactuarà amb el mapa

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament.

**Post-condicions:** Es mostraran al mapa els punts afegits

**Descripció:** L'usuari disposarà d'una funcionalitat que li permetrà carregar n punts al mapa, el component rebrà aquesta col·lecció de punts i els visualitzarà a la posició GPS corresponent

### **Cas d'ús Moure un punt del mapa.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament.

**Post-condicions:** Els punts moguts quedaran fixats a les noves posicions i la resta d'operacions restaran de nou operatives.

**Descripció:** L'usuari seleccionarà l'opció del menú designada a aquesta funcionalitat, moure els punts que es veuen al mapa.

Un cop seleccionada aquesta opció, l'usuari ja podrà re-situar tants punts del mapa com desitgi. Per fer això només caldrà fer el mateix moviment de drag & drop que en la funció de desplaçar el mapa. Caldrà clicar sobre un punt i bellugar-lo fins a la nova posició.

Un cop finalitzats tots els moviments desitjats només caldrà clicar al boto d'acceptar els moviments realitzats.

### **Cas d'ús Afegir un nou punt al mapa.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament.

**Post-condicions:** El nou punt passarà a formar part de la col·lecció de punts existents i la resta d'operacions restaran de nou operatives.

**Descripció:** L'usuari seleccionarà l'opció del menú dissenyada per aquesta funcionalitat, afegir un nou punt al mapa.

Un cop seleccionada aquest opció, apareixerà al mig de la zona de captura del mapa, un punt, l'usuari per mitja del sistema de drag & drop, ja explicat anteriorment i al apartat de vocabulari, podrà desplaçar aquest punt a la zona corresponent del mapa.

Quan l'usuari confirmi la selecció, el POI passarà a formar part de la col·lecció de punts existents.

**Excepcions:** Es podran afegir punts al mapa per paràmetre, sense necessitat de la interacció de l'usuari

### **Cas d'ús Mostrar la informació del punt.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament

**Post-condicions:** Es podrà veure la informació del punt.

**Descripció:** L'usuari podrà veure la informació que conte cada punt del mapa, per fer-ho només haurà de clicar sobre el punt en qüestió, en aquest moment, el component farà aparèixer una bafarada amb la informació del punt al seu interior.

**Excepcions:** En el cas que més d'un punt estiguin a la mateixa posició o se sobreposin degut al nivell de profunditat del mapa, apareixerà un menú dins de la informació que permetrà pàginar la informació de tots els punts sobreposats.

### **Cas d'ús Tancar la informació del punt.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament i es mostri la informació d'un punt.

**Post-condicions:** Desapareixerà la informació del punt.

**Descripció:** L'usuari podrà tancar la informació del punt en qualsevol moment, per fer-ho, només haurà de clicar sobre la informació del punt, en aquest moment, aquesta desapareixerà.

### **Cas d'ús Captura d'àrea circular.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament.

**Post-condicions:** Disposarem del resultat de la captura circular

**Descripció:** L'usuari seleccionarà l'opció del menú que permet realitzar captures circulars, seguidament clicarà sobre el mapa i estendrà la posició del punter del ratolí. Quan deixi anar el ratolí, es crearà una captura circular amb el radi seleccionat.

Un cop satisfet amb la captura, només caldrà acceptar i el component retornarà les dades de la captura per tal que les dades obtingudes siguin tractades per l'aplicació on el mapa es troba integrat.

**Excepcions:** Es podrà realitzar aquesta captura per paràmetres, sense necessitat de la interacció de l'usuari.

### **Cas d'ús Captura d'àrea rectangular.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament.

**Post-condicions:** Disposarem del resultat de la captura rectangular.

**Descripció:** L'usuari seleccionarà l'opció del menú que permet realitzar captures rectangulars, seguidament clicarà sobre el mapa i estendrà la posició del punter del ratolí. Quan deixi anar el ratolí, es crearà una captura rectangular amb l'àrea que s'haurà creat entre clic i clic.

Un cop satisfet amb la captura, només caldrà acceptar i el component retornarà les dades de la captura per tal que les dades obtingudes siguin tractades per l'aplicació on el mapa es troba integrat.

**Excepcions:** Es podrà realitzar aquesta captura per paràmetres, sense necessitat de la interacció de l'usuari.

### **Cas d'ús Tancar el visor.**

**Escenari:** Interfície web de l'aplicació.

**Actors:** Usuari que interactuarà amb el mapa.

**Pre-condicions:** Cal que el visualitzador de mapes s'hagi inicialitzat correctament.

**Post-condicions:** El component es tancarà

**Descripció:** El usuari disposarà d'un boto al menú que permetrà tancar el visor quan ho desitgi. Un cop clicat el botó el visor desapareixerà i deixarà de funcionar.

**Excepcions:** Es podrà realitzar aquesta captura per paràmetres, sense necessitat de la interacció de l'usuari.



Figura 2: Aquesta figura mostrà el diagrama de casos d'ús del sistema amb la interactuació dels diferents usuaris.

## Diagrama Entitat - Relació

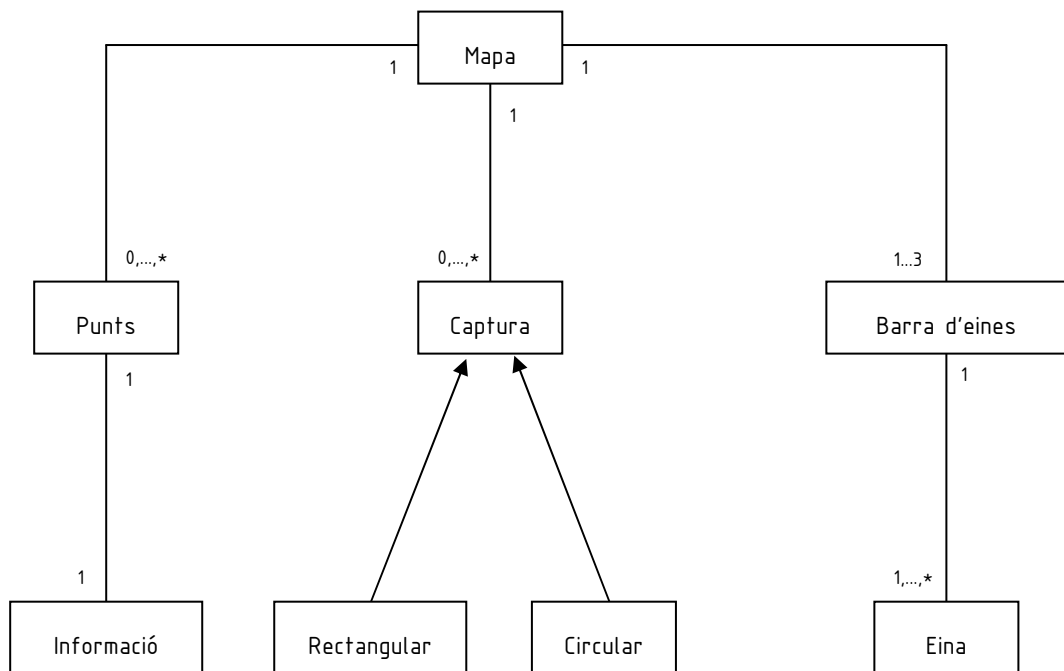
### 2.5.2

Disposem d'un mapa que estarà format des d'una a tres barres d'eines, on cadascuna d'aquestes barres pot tenir d'una a moltes eines incloses.

El mapa pot contenir de cap a molts punts d'informació, coneguts també com a POI's. Aquests POI's estaran definits per la seva informació.

El mapa també pot estar format per una captura, que bàsicament és una selecció d'una zona del mapa. Cal tenir present que aquesta captura pot adoptar dues formes:

- Forma rectangular.
- Forma de cercle perfecte.



*Figura 3:* Aquesta figura mostrà el diagrama entitat-relació del sistema



## Tecnologies investigades

## 2.6

En aquest apartat obtindrem informació de les tecnologies investigades que ens permetran desenvolupar el projecte, informació referent al funcionament d'aquestes, que ens ajudaran a entendre i a desenvolupar d'una manera més senzilla el visualitzador de mapes sol·licitat.

### WMS

### 2.6.1

Un Web Map Service, en català, un Servei Web de Mapes, produeix mapes de dades geo-referenciades (amb referències geogràfiques) tal i com s'explica en l'apartat de vocabulari, on es disposa d'una descripció del que és en essència un WMS.

WMS defineix un mapa com una representació visual de dades geogràfiques on el mapa no és les dades en si mateix.

En aquest punt explicarem amb més profunditat quin és el seu funcionament, de quina API es disposa, com s'interactua amb el servei o quines són les millors aplicacions del producte.

Bàsicament WMS defineix una sèrie de normes i protocols d'utilització del sistema, a fi de crear un estàndard internacional de servei de mapes via web. Aquesta especificació defineix tres operacions:

- **GetCapabilities:** Operació que retornarà meta-dades a nivell de servei, que seran una descripció del servei amb informació del contingut amb els paràmetres acceptables.
- **GetMap:** Operació que retornarà un mapa d'imatges geo-espacials amb les seves dimensions ben definides com a paràmetres.
- **GetFeatureInfo:** Operació opcional que retornarà informació sobre les característiques particulars del mapa mostrat.

Aquesta especificació defineix una sintaxi per a ser utilitzada via web per mitjà de la construcció d'una URL per a invocar cadascuna d'aquestes operacions. Per a la representació de les meta-dades s'utilitzarà XML (eXtensible Markup Language).

Quan se sol·licita un mapa, un client pot especificar la informació que es mostra (una capa, dues, ...), també hi ha la possibilitat d'especificar els estils de les capes, l'àrea de la terra que es desitja mostrar (bounding box), el sistema de coordenades geogràfiques que utilitzarà el sistema com a referència (el "Sistema de Referència Espacial" o SRS), el format de sortida, les dimensions de la sortida i la transparència i color del fons.

Quan dos o mes mapes es produeixen en el mateix bounding box, en el mateix sistema de referència espacial, la mateixa dimensió de sortida, els resultats poden ser tractats en capes per així poder produir un mapa compost. L'ús de formats d'imatge que suporten transparència permet que la capa menor sigui visible. Això permet que es pugui realitzar una composició d'imatges per mitja de diferents WMS, això és possible perquè l'especificació de creació ho permet.

Exemple d'utilització d'un WMS (Oficina del cadastre):

*[http://ovc.catastro.meh.es/Cartografia/WMS/ServidorWMS.aspx?VERSION=1.1.1&REQUEST=GetMap&SRS=EPSG:4230&LAYERS=Catastro&EXCEPTIONS=application/vnd.ogc.se\\_inimage&STYLES=&BBOX=-10,34,3.5,47.5&FORMAT=image/jpg&width=300&height=300](http://ovc.catastro.meh.es/Cartografia/WMS/ServidorWMS.aspx?VERSION=1.1.1&REQUEST=GetMap&SRS=EPSG:4230&LAYERS=Catastro&EXCEPTIONS=application/vnd.ogc.se_inimage&STYLES=&BBOX=-10,34,3.5,47.5&FORMAT=image/jpg&width=300&height=300)*



*Figura 4:* La figura mostra el resultat d'executar l'anterior URL d'exemple en un navegador web

## GetCapabilities, Crida

### 2.6.1.1

Tal i com es diu en l'apartat anterior, GetCapabilities retornarà meta-dades a nivell de servei, que seran una descripció del servei amb informació del contingut amb els paràmetres acceptables.

Per això caldrà construir la URL per tal de sol·licitar aquestes meta-dades.

| Paràmetre               | Requerit/Opcional |
|-------------------------|-------------------|
| VERSION=versió          | O                 |
| SERVICE=WMS             | R                 |
| REQUEST=GetCapabilities | R                 |
| UPDATESEQUENCE=text     | O                 |

*Figura 5:* Taula amb els atributs vàlids en una crida GetCapabilities

Exemple d'una crida a l'operació GetCapabilities:

<http://www.idee.es/wms/IDEE-Base/IDEE-Base?VERSION=1.1.1&REQUEST=GetCapabilities&SERVICE=WMS>

**VERSION:** El paràmetre versió, és un camp opcional a partir de la versió 1.0.0, tot i que si el WMS fos una versió anterior si que seria necessari posar el paràmetre dins de la crida. En les versions superiors automàticament el servei utilitzaria la versió més actual.

**SERVICE:** Aquest paràmetre indica quin dels tipus de serveis disponibles en el servei haurem d'utilitzar. Aquest paràmetre permet obtenir les diferents capes en un document en format XML de múltiples serveis web de OGC.

**REQUEST:** La naturalesa d'aquest paràmetre és indicar quina operació caldrà realitzar, dins de les Operacions bàsiques del servei.

**UPDATESEQUENCE:** Aquest paràmetre, opcional, serà utilitzat per tal de mantenir la consistència de la cache.

## GetCapabilities, Resposta

### 2.6.1.2

La resposta a la crida de GetCapabilities és un document en format XML.

**Name i Titles:** Dins del fitxer XML disposarem d'una sèrie d'elements que tenen un **<Name>** i un **<Title>**. Normalment el nom acostuma a ser una sola paraula (acostuma a ser un nom de màquina), mentrestant que el títol acostuma a ser un text més descriptiu.

**Meta-dades Generals:** A la primera part del fitxer XML és **<Service>** un element que aporta meta-dades

referents al servei. Aquest inclourà un nom, un títol i una URL de recursos en línia. Opcionalment també pot incloure un resum, la llista de paraules clau, informació de contacte, les taxes i les limitacions d'accés.

**Capas:** És la part més crítica del fitxer XML, cada mapa disposa d'un TAG **<Layer>** dins d'aquest trobarem més etiquetes que ens donaran informació referents a la capa.

- **<Name>**: Com el mateix nom indica, aquesta etiqueta indicarà el nom de la capa.
- **<Abstract>** i **<KeywordList>**: Etiqueta opcional, tot i que recomanada, és una descripció abstracta de la capa del mapa.

**Estil:** Podem tenir de cap a moltes etiquetes d'estil **<Style>** dins d'una capa. Dins de l'etiqueta estil també trobarem altres etiquetes que ens donaran informació d'aquest estil.

- **<Name>** i **<Title>**: Indican el nom i el títol del estil.
- **<Abstract>**: Ens donarà una descripció del estil.
- **<LegendURL>**: Conté la localització d'una imatge de la llegenda del mapa.
  - **<Format>**: Indica el MIME type del logotip de la imatge.

**SRS:** Cada capa conte una o moltes etiquetes **<SRS>**, SRS fa referència a Sistema de Referència Espaiat. Cal destacar que el SRS que correspon a Espanya és ED50, que en SRS equival a **<SRS>EPSG:4230</SRS>**.

**LatLonBoundingBox:** Cada capa té exactament un element **<LatLonBoundingBox>** i que estableix el mínim rectangle a partir de dades geogràfiques. Els atributs de LatLonBoundingBox minX, minY, maxX, maxY, indiquen els límits d'un rectangle en graus (admet decimals).

**BoundingBox:** Les capas disposaran de zero o més etiquetes **<BoundingBox>**. Es disposaran de tantes etiquetes com elements SRS. Aquesta etiqueta indicarà les dimensions de treball en un rectangle tal i com passa en el LatLonBoundingBox.

**ScaleHint:** Aquesta etiqueta ens indicarà quines son les escales de treball amb les que la capa podrà treballar. **<ScaleHint>**

**Dimensió i Extensió:** Dins de l'etiqueta de la capa també trobarem les etiquetes **<Dimension>** que ens indicarà la dimensió de la capa i **<Extension>** que ens indicarà l'extensió de la capa.

## GetMap

### 2.6.1.3

L'operació està dissenyada per a produir un mapa definit com una imatge en format digital. En rebre la sol·licitud el Servidor de mapes haurà de satisfer la petició retornant la captura que el client ha sol·licitat per mitjà d'una sèrie de paràmetres.

## GetMap, Crida

### 2.6.1.4

| Paràmetres de crida         | Requerit/Opcional |
|-----------------------------|-------------------|
| VERSION=versió              | R                 |
| REQUEST=GetMap              | R                 |
| LAYERS=llista_de_capas      | R                 |
| STYLES=llista_de_estils     | R                 |
| SRS=namespace:identificador | R                 |
| BBOX=minx,miny,maxx,maxy    | R                 |
| WIDTH=amplada_sortida       | R                 |

|   |   |
|---|---|
| HEIGHT=amplada_altura                       | R |
| FORMAT=format_sortida                       | R |
| TRANSPARENT=TRUE FALSE                      | R |
| BGCOLOR=valor_del_color                     | 0 |
| EXCEPTIONS=format_excepció                  | 0 |
| TIME=time                                   | 0 |
| ELEVATION=elevació                          | 0 |
| Altres dimensions                           | 0 |
| Paràmetres específics d'un Servidor concret | 0 |
| SLD=descripció_d'estil_de_capa_URL          | 0 |
| WFS=web_feature_service_URL                 | 0 |

**Figura 6:** Taula amb els atributs vàlids en una crida GetMap

Exemple d'una crida a GetMap:

*http://ovc.catastro.meh.es/Cartografia/WMS/ServidorWMS.aspx?VERSION=1.1.1&REQUEST=GetMap&SRS=EPSG:4230&LAYERS=Catastro&EXCEPTIONS=application/vnd.ogc.se\_inimage&STYLES=&BBOX=-10,34,3.5,47.5&FORMAT=image/jpg&width=120&height=120*

**VERSION:** Versió de resposta que volem obtenir. VERSION=1.1.1

**REQUEST:** Atribut que ens permetrà indicar al servei de mapes que volem obtenir una imatge concreta amb uns paràmetres concrets, definits per la resta d'atributs.

**LAYERS:** Llistat de capes separades per coma, d'una o més capes. Les capes hauran d'existir dins del llistat de capes que haurem obtingut dels *capabilities.xml* obtingudes prèviament amb el GetCapabilities. Caldrà utilitzar el contingut de l'etiqueta *<name>* de l'apartat *<layer>*. Un exemple de llista vàlida seria: **LAYER=capa1,capa2,capa3**. Cal tenir present que l'ordre en que inscrivim les capes influeix en el resultat final, ja que podem tenir problemes amb les transparències d'aquestes capes.

**STYLES:** Llistat d'estils separats per coma, d'un o més estils. Al igual que el cas anterior, caldrà obtenir els estils de *capabilities.xml* utilitzant el contingut de l'etiqueta *<name>* de l'apartat de *<style>*. Un exemple d'utilització seria: **STYLE=style1,style2**. En cas de no conèixer cap estil, seria vàlid declarar-lo com a null (sense valor), però el l'atribut sempre haurà d'estar present en la URL ("style=" o "style=,,,").

**SRS:** Paràmetre que estableix el sistema de referència espacial que s'aplica als valors del Bounding Box. Cal destacar que el valor de SRS haurà d'aparèixer dins del fitxer *capabilities.xml* subministrat pel servidor, etiquetat com a *<SRS>*. Si aquest valor es defineix com a **NONE**, pot provocar problemes amb les capes ja que l'espai de referència pot no ser correcte. Un exemple d'utilització: **SRS=EPSG:4230**, que correspon a un dels espais de referència d'Espanya i subministrats pel WMS del cadastre.

**BBOX:** Paràmetre determinat pel usuari i que sol·licita quina és la porció del mapa que aquest desitja veure. És un paràmetre format per 4 valors, minx, miny, maxx i maxy, que s'introduiran a la URL separats per comes. D'aquesta manera es pot formar un rectangle, amb el que el servidor pot obtenir la porció del mapa que ha de subministrar al client. Cal tenir present que aquests valors son posicions GPS i que s'introduiran en graus (accepta decimals). Exemple d'utilització: **BBOX:-10,34,3.5,47.5**

**FORMAT:** El paràmetre format indica amb quin format desitgem rebre la resposta, el que és el mateix, amb quin MIME-type desitgem rebre la imatge. Disposem des de **GIF** (image/gif), **PNG** (image/png) i **JPEG** (image/jpeg) dins dels formats més comuns.

**WIDTH i HEIGHT:** Com el nom indica, aquests camps fan referència a la dimensió de la imatge que el client vol obtenir, amplada (Width) i altura (Height). Una dimensió que estarà expressada en píxels. Cal anar amb compte amb aquest paràmetre, ja que un mal ús de les dimensions combinat amb un mal ús del Bounding box, pot provocar que obtinguem una imatge distorsionada.

**TRANSPARENT:** Paràmetre opcional que ens permet donar transparència al fons de la imatge, per defecte aquesta propietat restarà com a fals (False) però si es desitja que estigui operativa només caldrà posar el paràmetre a Cert (True). Aquest és un paràmetre que també anirà molt lligat amb el format de la imatge, ja que hi ha formats que no permeten la transparència, com pot ser jpeg.

**BGCOLOR:** Atribut que especificarà quin ha de ser el color de fons. Per la seva naturalesa i importància, és un paràmetre opcional i que s'expressarà en hexadecimal de la forma **0xRRGGBB**. **RR (Red - vermell)**, **GG (Green - Verd)** i **BB (Blue - Blau)**.

**EXCEPTIONS:** Aquest paràmetre indica la manera en la que el client vol rebre els errors que s'han produït. Un fitxer xml (*application/vnd.ogc.se\_xml*), una imatge (*application/vnd.ogc.se\_inimage*) o una captura en blanc (*application/vnd.ogc.se\_blank*).

## GetMap, Resposta

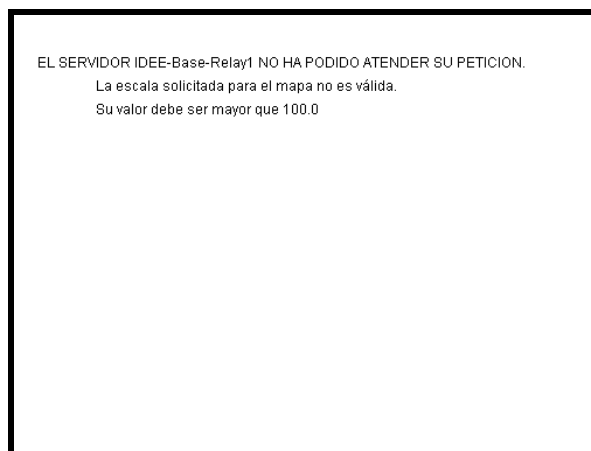
### 2.6.15

La resposta que obtindrem després de llançar la URL amb el paràmetre de REQUEST GetMap serà una imatge amb els paràmetres especificats de dimensió, BBOX, capes, estils, format i transparència.



*Figura 7:* Retorn d'una captura correcte del WMS del IGN

En cas que la sol·licitud no s'hagi pogut resoldre satisfactòriament el Servidor ens informarà de quin ha estat el possible error, tal i com podem veure a la Figura 8.



*Figura 8:* Retorn d'una excepció

## GetFeatureInfo

2.6.1.6

Aquesta és una operació opcional i suportada per aquelles capes que tinguin l'atribut de *queryable="1"*. Aquesta operació esta dissenyada per subministrar als clients d'un WMS més informació referent a les possibilitats que ofereix el servei respecte a les imatges que haurà retornat prèviament la funció GetMap.

## GetFeatureInfo, crida

2.6.1.7

| Paràmetres de crida                         | Requerit/Opcional |
|---|-------------------|
| VERSION=versió                              | R                 |
| REQUEST=GetFeatureInfo                      | R                 |
| <copia de la crida a GetMap>                | R                 |
| QUERY_LAYERS=llista_de_capes                | R                 |
| INFO_FORMAT=format_de_sortida               | O                 |
| FEATURE_COUNT=número                        | O                 |
| X=píxel_columna                             | R                 |
| Y=píxel_fila                                | R                 |
| EXCEPTIONS                                  | O                 |
| Paràmetres específics d'un Servidor concret | O                 |

Figura 9: Taula amb els atributs vàlids en una crida GetFeatureInfo

Exemple d'una crida a GetFeatureInfo:

```
http://ovc.catastro.meh.es/Cartografia/WMS/ServidorWMS.aspx?REQUEST=GetFeatureInfo&VERSION=1.1.1&SRS=EPSG:4230&LAYERS=Catastro&EXCEPTIONS=application/vnd.ogc.se_inimage&STYLES=&BBOX=-10,34,3.5,47.5&FORMAT=image/jpg&width=120&height=120
```

**VERSION:** Paràmetre que especifica la versió de WMS que hem d'utilitzar, aquest és un paràmetre requerit.

**REQUEST:** Paràmetre que hem d'utilitzar si volem utilitzar la funcionalitat de GetFeatureInfo.

**Copia de la crida a GetMap:** No és un paràmetre del estil nom=valor com els altres paràmetres, sinó que són els mateixos paràmetres passats per obtenir el mapa, per tal de poder obtenir el mapa original. Cal tenir present que els camps de **VERSION** i **REQUEST** s'ometran, per tal de poder obtenir la informació pròpia del GetFeatureInfo.

**QUERY\_LAYERS:** Paràmetre que estableix les característiques de les capes del mapa. És una llista separada per comes. Si el paràmetre és buit o una de les capes de la llista no apareix al llistat de capes del fitxer *capabilities.xml*, obtindrem una excepció.

**INFO\_FORMAT:** atribut que definirà el format en el que desitgem obtenir la informació que el GetFeatureInfo ens ha de subministrar. Aquest paràmetre estarà definit al fitxer xml obtingut al fer GetCapabilities dins de l'etiqueta *<request><featureInfo><format>*. Aquest valor acostumarà a ser un MIME-type. Un exemple d'utilització *INFO\_FORMAT=application/vnd.ogc.gml*.

**FEATURE\_COUNT:** atribut que estableix el número màxim de dades s'obre una informació que serà retornada. Aquest valor haurà de ser un enter positiu més gran que 0 i per defecte s'agafarà el 1.

**X i Y:** Paràmetres que indiquen el punt d'interès del mapa identificat un punt dins dels marges de amplada i altura definits en la crida al GetMap.

**EXCEPTIONS:** Atribut que indicarà la manera en la que desitgem rebre un possible error en la crida al servei. Els formats acostumaran a ser els mateixos que estan definits en la crida de GetMap, com per exemple: *application/vnd.ogc.se\_xml*.

## GetFeatureInfo, Resposta

2.6.1.8

El WMS retornarà una resposta d'acord a la consulta, sempre que la consulta hagi estat correcta i no es produeixi una excepció per qualsevol altre qüestió.

## Google Maps i Google Earth

2.6.2

Google Maps és el nom d'un servei gratuït de Google. Bàsicament és un servidor d'aplicacions de mapes Web, que ofereix imatges de mapes desplaçables, així com de fotografies de satèl·lits del món sencer, inclús permet la creació de rutes entre diferents ubicacions.

Google Maps és similar a Google Earth, que és una aplicació disponible en Windows, Mac i Linux que ofereix vistes del globus terraquí, però Google Maps esta pensat per a ser integrat en una pàgina web.

### API Google Maps

2.6.2.1

L'element fonamental a qualsevol aplicació del API de Google Maps és el propi mapa. Explicarem l'ús del objecte fonamental que és GMap2 i els aspectes bàsics de les operacions amb el mapa.

A partir d'un exemple, explicarem el funcionament bàsic del codi:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Exemple de la API Javascript de Google Maps</title>
    <script src="http://maps.google.com/maps?file=api&v=2&key=abcdefg" type="text/javascript"></script> (1)
    <script type="text/javascript">
      Function initialize() { (3)
        if (GBrowserIsCompatible()) {
          var map = new GMap2(document.getElementById("map_canvas"));
          map.setCenter(new GLatLng(37.4419, -122.1419), 13); (4)
        }
      }
    </script>
  </head>
  <body onload="initialize()" onunload="GUnload()"> (5)
    <div id="map_canvas" style="width: 500px; height: 300px"></div> (2)
  </body>
</html>
```

Amb la base d'aquest codi, pràcticament es poden integrar totes les funcionalitats que la API aporta.

Hem de tenir en compte els aspectes següents:

1. Incloem el codi Javascript del API de Google Maps mitjançant l'etiqueta script.
2. Creem un element *div*, de nom *"map\_canvas"* que allotja el mapa.
3. Escrivim una funció Javascript per crear un objecte de mapa.
4. Centrem el mapa en un punt geogràfic concret.
5. Inicialitzem l'objecte del mapa des del esdeveniment *onLoad* del *<body>*

## Càrrega de la API

### 2.6.2.2

La direcció URL `http://maps.google.es/map?file=api&v=2&key=abcdefg` dirigeix a la ubicació del fitxer Javascript que inclou tots els símbols i definicions necessaris per a poder utilitzar la API de Google Maps. Per accedir a la API, és necessari disposar de l'etiqueta `<script>` dirigint el *source (src)* a aquesta direcció.

El paràmetre *file*, indica que utilitzem la API de Google Maps, la *v* indica la versió de la API que utilitzarem i com hem comentat anteriorment *key* és la clau pública per poder accedir al recurs.

La *v*, ens indica quina versió de la API de Google Maps volem d'utilitzar.

## Elements DOM del mapa

### 2.6.2.3

**GMap2:** La classe de Javascript que representa els mapes és GMap2. Cada objecte d'aquesta classe defineix un únic mapa a una pàgina. Per crear la instància d'aquesta classe haurem de fer-ho mitjançant l'operador *new* de Javascript.

Aquest codi defineix una variable anomenada *map* i assigna la variable a un nou objecte GMap2. La funció *GMap2()* és un constructor i la seva definició.

```
var map = new GMap2("recipient", [paràmetres opcionals])
```

**Inicialitzador del Mapa:** Una vegada creat el mapa mitjançant GMap2, caldrà inicialitzar-lo. Aquesta inicialització es realitza amb la funció *setCenter()* que ens aporta el mapa.

*setCenter()* requereix d'una coordenada *GLatLng* i un nivell d'aproximació. Per poder realitzar qualsevol altre acció serà necessari haver realitzat aquesta inicialització.

```
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
```

**Càrrega del mapa:** Mentrestant s'està processant la pàgina HTML, s'externalitza el model d'objectes de documents (DOM) i les imatges, es reben seqüències de comandes externes que incorporen l'objecte document. Per garantir que el mapa de Google només s'afegeixi quan la pàgina s'hagi carregat per complet, per fer-ho haurem de crear el mapa una vegada el cos de la pàgina s'hagi carregat. Per fer-ho caldrà incloure la crida a la funció *initialize()* que ho inicialitza tot de la següent manera:

```
<body onload="initialize()" onunload="GUnload()">
```

Cal tenir en compte que el mètode *initialize()* és un mètode desenvolupat per l'usuari, no forma part de la API de Google, però el contingut d'aquest mètode sí que és part de la API.

La funció *GUnload()* és una utilitat dissenyada per evitar les fuites de memòria.

**Atributs del mapa:** De forma predeterminada, els mapes es mostren a la API de Google Maps mitjançant mosaics d'aspecte estàndard. No obstant també admet altres tipologies de mapes. Els estàndard són:

- *G\_NORMAL\_MAP:* Vista predeterminada.
- *G\_SATELLITE\_MAP:* Imatges satèl·lit de Google Earth.
- *G\_HYBRID\_MAP:* Barreja les vistes normals i de satèl·lit.
- *G\_DEFAULT\_MAP\_TYPES:* Una barreja d'aquest tres tipus de mapes, útil per un processament repetitiu.

Fixarem la tipologia del mapa utilitzant la funció *setMapType(tipologia)*

La API de Google Maps utilitza l'anglès com a forma predeterminada per mostrar la informació textual (noms de controls, drets d'autor i rutes). És possible canviar el llenguatge del mapa mitjançant la crida a un mètode que definirà el nou idioma.

## Esdeveniments del mapa

### 2.6.2.4

Els esdeveniments de la API de Google Maps es gestionen mitjançant *GEvent* que registrarà els detectors d'esdeveniments. Cada objecte de la API exporta una determinada quantitat d'esdeveniments amb nom. Per exemple l'objecte GMap2 exporta els esdeveniments *click*, *dblclick*, *move*,...



Per registrar les notificacions d'aquests esdeveniments utilitzarem el mètode estàtic `GEvent.addListener()`, que agafarà un objecte, un esdeveniment i una funció que ha de cridar quan l'esdeveniment especificat es produeixi.

Exemple de la creació d'un esdeveniment:

```
GEvent.addListener(map, "click", function() {  
    alert("Funció que hem d'executar.");  
});
```

## Controls del mapa

### 2.6.2.5

Els mapes disposen d'uns elements anomenats controls que permeten interactuar amb el mapa. Aquest controls poden ser modificats al gust del usuari i inclús poden ser personalitzats mitjançant la subclasse de la classe *GControl*.

L'API de Google Maps incorpora diferents controls integrats:

- **GLargeMapControl:** Control de desplaçament i zoom de grans dimensió. Es mostra a la part superior esquerra de forma predeterminada.
- **GSmallMapControl:** Control de desplaçament i zoom de petita dimensió. Es mostra a la part superior esquerra de forma predeterminada.
- **GSmallZoomControl:** Control de zoom (sense desplaçament) de petita dimensió.
- **GScaleControl:** Control de l'escala del mapa.
- **GMapTypeControl:** Botons que permeten canviar el tipus de mapa (satèl·lit, mapa, híbrid).
- **GHierarchicalMapTypeControl:** Selecció de botons nidats i elements de menú per col·locar molts selectors de tipus de mapes.
- **GOverviewMapControl:** Mapa de visió general a la cantonada de la pantalla que es pot contraure.

Exemple de la creació d'un control:

```
var control = new GMapTypeControl();
```

El mètode *addControl*, ens permet afegir els controls al mapa, en una posició concreta, tot i que per defecte els col·loca a la part superior esquerra. Si volem col·locar els control en una posició diferent utilitzarem *GControlPosition* que permet especificar una posició:

- **G\_ANCHOR\_TOP\_RIGHT:** Part superior dreta.
- **G\_ANCHOR\_TOP\_LEFT:** Part superior esquerra.
- **G\_ANCHOR\_BOTTOM\_RIGHT:** Part inferior dreta.
- **G\_ANCHOR\_BOTTOM\_LEFT:** Part inferior esquerra.

Exemple de posicionament d'un control:

```
var pos = new GControlPosition(G_ANCHOR_BOTTOM_RIGHT, new GSize(10,10));  
map.addControl(control, pos);
```

## Crear un POI

### 2.6.2.6

Els marcadors identifiquen els punts dins del mapa. Per defecte s'utilitzen els identificadors de punts *G\_DEFAULT\_ICON*, tot i que es pot personalitzar i utilitzar icones personalitzades. El constructor de *GMarker* utilitza *GLatLng* i l'objecte opcional *GMarkerOptions* com arguments.

Els marcadors estan dissenyats per ser interactius, el que significa que també podem assignar esdeveniments de tipus click, per tal de ensenyar un quadre d'informació.

Exemple de la creació d'un marcador:

```
var point = new GLatLng(latitud, longitud);
```

```
map.addOverlay(new GMarker(point));
```

## Generació de la API KEY

2.6.2.7

Per tal de poder executar Google Maps dins de les nostres pàgines web caldrà disposar d'una clau d'accés, coneguda com API KEY, que ens serà proporcionada per Google Maps, podem veureu en aquesta enllaç <http://code.google.com/intl/es/apis/maps/signup.html>

Google Maps ens sol·licitarà l'enllaç al nostre web, per tal que en la inicialització del mapa, pugui comprovar que realment esta tot correcte.

Cal tenir en compte que si tenim previst superar els 500.000 accessos caldrà adreçar-se amb Google, ja que el volum de consultes pot provocar que s'hagi d'adquirir el pack Google Enterprise.

## Plataforma

2.7

Tenint en compte on s'haurà d'integrar el visualitzador de mapes, el SICT/JP, ja ens obliga a restringir la plataforma en la que haurà de ser desenvolupat el component.

Com s'ha esmentat anteriorment, el SICT/JP, és una aplicació que funciona en interfície web. Per mitjà d'un Framework (propi de Slash Solutions), uns "Workers" programats en Java, i les pàgines on es renderitzaran les diferents pantalles de l'aplicació en format JSP.

La idea inicial, consistiria en que el servidor gestiones tota interacció amb els servidors WMS, instal·lats o predefinitos pel client. El servidor de l'aplicació gestionaria les crides, la gestió i el display dels punts,... Això implicarà doncs que la plataforma en la que serà implementat és Java, però aquesta solució no serà seleccionada perquè provocaria masses transaccions amb el servidor de l'aplicació, i faria l'aplicació massa lenta.

Tenint en compte que les pàgines estan construïdes en JSP's, ens fa pensar que una bona solució per a gestionar les crides i les funcionalitats, seria utilitzar Javascript.

## Javascript

2.7.1

És un llenguatge script basat en el prototip desenvolupat per Netscape, i que va derivar en l'estàndard ECMAScript. La seva principal utilització és en pàgines web tot i que es pot integrar en altres aplicacions.

Malgrat el seu nom, Javascript no deriva de Java, tot i que els seus components també tenen una sintaxi similar. Bàsicament Javascript és més proper al llenguatge *Self* i *ActionScript* (basat en ECMAScript).

## Història

2.7.1.1

Al 1995 Netscape va desenvolupar la primera versió d'aquest llenguatge, però amb el nom de *Mocha*, reanomenat a *LiveScript* i finalment com a Javascript. Aquest últim canvi va coincidir amb la inclusió de la tecnologia Java per part del navegador.

En els inicis nomes s'utilitzava en pàgines web HTML, per realitzar feines i operacions de marc de l'aplicació client servidor. Amb l'aparició de la Web 2.0, Javascript s'ha popularitzat ja que aporta potència de càlcul al navegador per augmentar-ne la usabilitat d'aplicacions web amb tècniques avançades.

Al any 1997 els autors van proposar Javascript com l'estàndard del *European Computer Manufactures Association (ECMA)*, on finalment va ser adoptat com estàndard *ECMA*, conegut amb el nom de *ECMAScript*. Poc després també va esdevenir estàndard *ISO*.

*JScript* és la implementació ECMAScript desenvolupada per Microsoft, molt similar al Javascript de Netscape, però amb certes diferències que provocava incompatibilitats de versions.

Per evitar quest problema, *World Wide Web Consortium* va dissenyar l'estàndard DOM (*Document Object Model* o en català *Model d'objectes del Document*), incorporat en Konqueror, Internet Explorer 6 i superior, Netscape Navigator, Opera des de la versió 7 i Mozilla des de la primera versió

**Tipus de dades:** Javascript reconeix sis tipus de dades de valors:

- **Cadenes:** Aquest tipus de dades van entre cometes simples (') o dobles (").
- **Números:** No es fa distinció.
- **Booleans o lògiques (boolean):** true o false.
- **Nul (null):** Té valor nul.
- **Indefinit (undefined):** És el que correspon a una variable que ha estat declarada però que no se li ha assignat cap valor.

**Variables:** Les variables en Javascript no disposen d'un tipus concret en la seva definició, la declaració d'aquestes es fa mitjançant var. Si no utilitzem var, aquesta variable passarà a ser una variable global.

Exemples de declaracions de variables:

```
var x = valor;
```

```
y = valor;
```

Quan es declara una variable no s'ha d'indicar de quin tipus és, a diferència d'altres llenguatges

**Constants:** Per declarar una constant cal seguir les mateixes normes que en el cas de les variables, però serà obligatori utilitzar la paraula cons, en cas que no es faci, s'interpretarà com a variable global. No es pot declarar una constant amb el mateix nom que una funció o que una variable dins del mateix àmbit.

Si la constant es declara dins d'una funció serà considerada d'àmbit local, en el cas contrari serà considerada d'àmbit global.

Exemple de declaracions de constants:

```
cons GLOBAL = valor;
```

**Array:** Una matriu o vector o *array*, és una col·lecció de variables del mateix tipus i amb el mateix nom. Farem referència a un element concret d'aquesta col·lecció indicant el seu índex.

A diferència de molts llenguatges de programació que limiten l'ús del índexs a un enter, en Javascript, aquest índex pot ser tant un numero com una cadena de caràcters.

Exemples d'utilització d'arrays:

```
var nom_array = new Array()
```

```
nom_array[índex] = valor;
```

**Operadors:** Els operadors de Javascript són els mateixos que per exemple es disposen en Java.

**Funcions:** en Javascript per declarar les funcions només caldrà utilitzar la paraula clau function.

Un exemple de declaració d'una funció:

```
function funcioExemple() { ... <Contingut de la funció> ... }
```

Per a més informació és pot consultar la web de Javascript per tal de conèixer tota l'especificació:

<http://www.ecmascript.org/docs.php>

## HTML

## 2.7.2

HTML correspon al acrònim de *Hyper Text Markup Language* o el que és el mateix en català Llenguatge de marcat d'hipertext. És un llenguatge de marcat que deriva de SGML dissenyat per estructurar textos i relacionar-los en forma d'hipertext. Gràcies a Internet i als navegadors web, s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents.

## Història

### 2.7.2.1

En els seu origen, HTML era un llenguatge dissenyat per a compartir informació científica entre científics de tot el món. Era purament un llenguatge estructural, en què no hi havia forma de descriure l'aparença de les pàgines (ni tan sols la possibilitat de posar un text en negreta o cursiva). Més endavant s'hi van afegir nombroses opcions per tal de poder presentar el text i els grafismes.

A mitjans de la dècada de 1990 van començar a aparèixer les aplicacions HTML per aconseguir la presentació desitjada, però sempre des de diferents perspectives de diferents desenvolupadors, que van acabar amb diverses solucions no estàndards per a diferents navegadors. Això va provocar l'aparició d'un consorci que controla l'evolució de HTML, *W3C (World Wide Web Consortium)*.

Aquesta evolució tenia un punt clau: La separació del contingut i l'aparença. Amb la versió 4 de HTML es recomanava un altre mecanisme per controlar la visualització del nostre contingut HTML: *CSS (Cascading Style Sheets* o en català *Fulls d'estils*).

W3C recomana l'ús de XHTML que manté la mateixa sintaxi i els mateixos mecanismes que HTML, però reformulat amb les normes d'un XML, preparant-se així per aprofitar les avantatges d'aquest llenguatge.

*HTML 5* intenta millorar la part d'aplicació web amb l'especificació *Web Forms 2.0*. Aquest grup surt com a reacció pel canvi bruscat del pas de HTML a XHTML que es pot utilitzar amb navegadors que no suporten *MIME-type application/xhtml+xml* gracies a l'apèndix de C de l'especificació XHTML 1.0.

La última especificació vigent és l'XHTML 1.1 que ja no contempla cap compatibilitat amb versions anteriors i, per tant, només es pot servir com a *application/xhtml+xml* exclouent qualsevol navegador antic.

El punt més polèmic actualment és la proposta d'especificació (en estat d'esborrany) XHTML 2.0 que deixa de ser compatible amb versions anteriors, no només a nivell de MIME-type sinó que l'estructura de document i els elements estructurals també canvien.

## Elements bàsics

### 2.7.2.2

Les etiquetes bàsiques i d'obligada presència en tot document són:

- **<!DOCTYPE>**: És l'etiqueta que permet definir el tipus de document HTML que s'empra. Existeixen tres tipus bàsics: l'estricta (Strict), el transicional (Transitional) i el de marcs (Frameset).
- **<HTML>**: És l'etiqueta arrel de qualsevol document HTML o XHTML.
- **<HEAD>**: Defineix la capçalera del document HTML. Permet declarar meta informació del document i que no es mostra directament en el navegador. Aquesta informació és d'especial rellevància pels indexadors i cercadors automàtics.
- **<BODY>**: Defineix el cos del document. Aquesta és la part del document HTML que es mostra en el navegador.

Dins de la capçalera **<HEAD>** podem trobar:

- **<TITLE>**: Permet definir el títol de la pàgina. En navegadors gràfics el contingut del *title* apareix a la barra del títol a sobre de la finestra.
- **<META>**: Permet definir meta informacions del document tals com l'autor, la data de realització, la codificació del document (UTF, ISO, etc.), les paraules clau i la descripció del mateix.
- **<LINK>**: Permet definir metadades complementàries a les del meta tals com el document anterior, el següent, el capítol al qual pertany el document, la pàgina glossari, etc.

Dins del cos **<BODY>** hi podem trobar:

- **<A>**: Crea un enllaç a un altre document o a una altra zona del mateix, segons els atributs.
- **<H1>**, **<H2>**, ..., **<H6>**: Capçaleres o títols del document, acostumen a distingir-se per mida.
- **<DIV>**: Divisió estructural de la pàgina.
- **<P>**: Paràgraf.



| <b>Resum de costos de desenvolupament</b>   |                   |
|---|-------------------|
| <b>Costos laborals</b>  |                   |
| <i>Sou d'un analista-programador, 300 hores de feina</i>  | <i>4.500,00 €</i> |
| <i>Despeses: S.S. (23,60% sobre el sou), Atur (1,60%), FP (0,10%), 300 hores de feina</i>   | <i>1.138,50 €</i> |
| <b>Costos tècnics</b>   |                   |
| <i>Consum: serveis (electricitat, etc) i consumibles (discos, paper, tòner, enquadernació) directament imputables al projecte</i> | <i>180,00 €</i>   |
| <i>Documentació i formació: Llibres, manuals, etc.</i>  | <i>120,00 €</i>   |
| <b>Total</b>  | <b>5.938,50 €</b> |

Un preu de venda adequat del software, després d'aplicar el marge de venda de l'ordre del cinquanta per cent, seria de **8.907,70 €**. Aquest preu està calculat com a preu per a una venda en exclusiva, és a dir, un desenvolupament a mida per a una persona física o jurídica en concret.

L'aplicació d'un marge tant elevat és per tal d'evitar un retràs en l'entrega del projecte, així d'aquesta manera, en cas de possibles problemes l'empresa sempre disposarà d'un benefici.

\* Nota: Part de la informació que es mostra a continuació forma part del pressupost del projecte SICT/JP assignat a la planimetria (nom amb el que es coneix al visualitzador de mapes dins del projecte).

En aquest apartat intentarem plasmar tota la informació que hem recollit en l'apartat d'anàlisi i començar a desenvolupar el que serà el visualitzador de mapes WMS.

## Disseny gràfic del visualitzador

3.1

En aquest apartat intentarem descriure l'aspecte gràfic que tindrà el visualitzador de mapes, amb els apartats i les eines que ha d'incloure.

### Barra d'eines

3.1.1

Per poder realitzar les operacions amb el mapa serà necessari disposar d'una barra d'eines.



Figura 11: Imatge del que serà la barra d'eines del mapa.

### Eines Bàsiques

3.1.1.1



**Vista Completa:** Aquesta icona ens permetrà en qualsevol moment retornar a la posició inicial del mapa. Funcionalitat molt útil si després de fer una sèrie de zooms i moviments volem tornar al punt inicial per tal de realitzar una nova operació amb el mapa.



**Desplaçar el mapa (PAN):** Amb aquesta icona podrem realitzar desplaçaments del mapa. Funcionalitat considerada bàsica, ja que si no podem desplaçar la imatge, el visualitzador perd tota utilitat.



**Zoom:** Amb aquesta eina podrem realitzar captures d'una àrea de Zoom. Utilitat bàsica i molt útil si volem realitzar un zoom sobre una zona concreta del mapa.



**Zoom In:** Amb aquesta utilitat ens permetrà fer un zoom, endinsant-nos dins el mapa, agafant com a centre el centre de la imatge del mapa que s'està mostrant en aquell moment.



**Zoom Out:** Amb aquesta utilitat podrem realitzar l'operació contrària a la esmentada anteriorment, és a dir, podrem fer un zoom allunyar-nos del mapa, agafant com a centre el centre de la imatge del mapa que s'està mostrant en aquell instant.

### Eines no bàsiques

3.1.1.2



**Desplaçar POI's:** Eina que ens permet desplaçar tots els punts d'interès que es mostren en el mapa en aquell moment. Podrem desplaçar qualsevol punt i inclús fer-ho simultàniament (un darrera l'altre però dins de la mateixa operació) Si disposem dels permisos pertinents podrem utilitzar aquesta eina.



**Crear un POI:** Aquesta eina ens permetrà afegir un punt d'interès al mapa. Aquest punt apareixerà inicialment al centre de la imatge que s'està mostrant actualment.



**Captura rectangular:** Amb aquesta utilitat podrem fer captures d'àrea amb forma rectangular.



**Captura circular:** Amb aquesta icona dispondrem de la funcionalitat que ens permetrà crear una captura circular (Aquesta serà un cercle perfecte).



**Acceptar:** Aquest boto ens permetrà acceptar una acció que s'estigui realitzant en aquell moment. Si el boto es de color gris significarà que en aquell moment no hi ha cap operació activa.



**Cancel·lar:** Aquest boto ens permetrà realitzar l'operació contrària, ens permetrà cancel·lar qualsevol operació que s'estigui realitzant en aquell moment. Com passava abans, si el boto es de color gris no hi haurà cap operació en curs.



**Tancar el mapa:** Aquest boto servirà per poder tancar el mapa en aquelles zones de les nostres aplicacions on aquest s'hagi obert com a *popup* o com a *div* flotant.

## Menú lateral

3.1.2

El mapa disposarà d'un menú lateral amb una sèrie de funcionalitats bàsiques que ens permetran variar i modificar l'aparença del mapa. Cal tenir en compte que si ho desitgem aquesta part del mapa podrà ser no visible.

### Mapa de referència

3.1.2.1

Menú lateral que contindrà una captura del mapa amb unes dimensions molt més reduïdes i que contindrà un requadre que mostrarà la captura que s'està mostrant en aquell instant en el mapa gran.

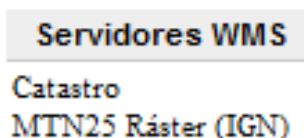


*Figura 12:* Mostra el mapa de referència del menú lateral.

### Servidors WMS

3.1.2.2

En aquest punt es mostraran els servidors amb els que el client desitja interactuar, són els que es llegiran d'un arxiu XML i que es carregaran en aquest punt. Al clicar sobre els noms dels servidors el mapa, automàticament recollirà la imatge del nou mapa i carregarà les noves capes disponibles per aquest.



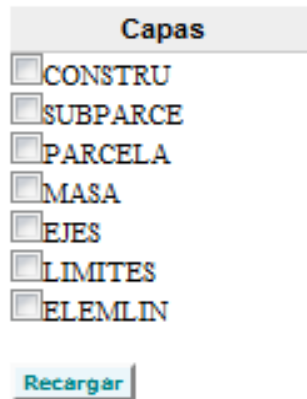
*Figura 13:* Mostra els servidors WMS disponibles.

### Capes

3.1.2.2

En aquest apartat podrem visualitzar les capes que el servidor WMS ens haurà retornat i que estaran ordenades per ordre de transparència, a fi que tot es pugui visualitzar correctament.





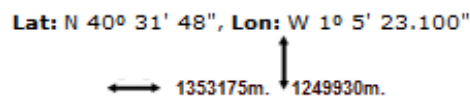
**Figura 14:** Mostra el llistat de capes que ens haurà retornat el servidor WMS seleccionat en aquell moment. Si cliquem al boto de recarregar, s'obindrà el nou mapa amb les capes seleccionades.

### Menú inferior

### 3.1.3

A la part inferior de la imatge del mapa disposarem de la posició GPS del punter dins del mapa, aquesta posició s'anirà refrescant a cada mínim moviment del ratolí dins d'aquest. Si el ratolí surt del àmbit de treball del mapa, el comptador de posició s'aturarà.

També disposarem d'un indicador que ens informarà del espai que representa la imatge en metres, es a dir, que representa l'altura del mapa en metres a la realitat i que representa l'amplada en metres a la realitat.



**Figura 15:** Menú inferior que mostrarà la posició GPS del punter i l'àmbit de treball en metres del mapa.

### Punts d'interès

### 3.1.4

En el mapa haurem de representar d'alguna manera els punts d'interès, hem de tenir present que existiran dos tipus de punts d'interès, els nous i els ja creats (existents). Cal tenir en compte que la representació gràfica d'aquests es podrà modificar en el moment que es vulgui, ja que estarà implementat de tal manera que puguem canviar la representació gràfica de qualsevol punt d'interès.



**Nou punt:** Aquesta serà la representació gràfica d'un nou punt d'interès, serà la selecció que s'ha fet en el cas que no s'especifiqui un altre tipus d'imatge.

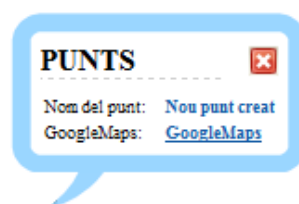


**Punt existent:** Aquesta serà la representació gràfica per defecte d'un punt existent.

### Informació d'un punt d'interès

### 3.1.5

Quan disposem d'un punt d'interès ja creat, aquest disposa de la possibilitat de mostrar la informació que conté, per això només farà falta pitjar sobre el punt d'interès i es mostrarà una "bafarada" que contindrà dades informatives referents al punt.



**Figura 16:** Mostra la informació que podria aparèixer al pitjar sobre un POI.

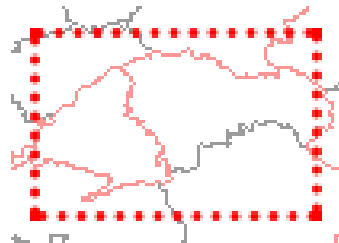


Aquest botó ens permetrà tancar la informació, tot i que si cliquem sobre qualsevol posició (sense cap enllaç) també ho tancarà.

## Captura rectangular

3.1.6

Com a funcionalitat serà necessari crear o pensar com ha de ser el resultat de realitzar una captura rectangular. Després de crear mitjançant el clic del ratolí en un origen i un destí obtindrem una imatge dins del mapa.

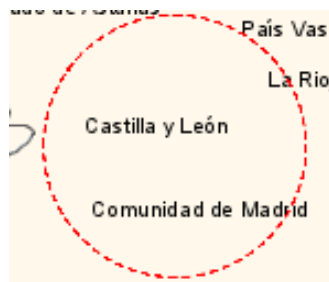


*Figura 17:* Resultat de realitzar una captura rectangular.

## Captura circular

3.1.7

També haurem de crear o pensar com ha de ser el resultat de realitzar una captura circular. Després de crear mitjançant el clic del ratolí en un origen i un destí obtindrem una imatge dins del mapa.



*Figura 18:* Resultat de realitzar una captura circular.

## Diagrames

3.2

### Diagrama de Classes

3.2.1

Partim de la interfície web, que serà on s'haurà d'integrar el visualitzador.

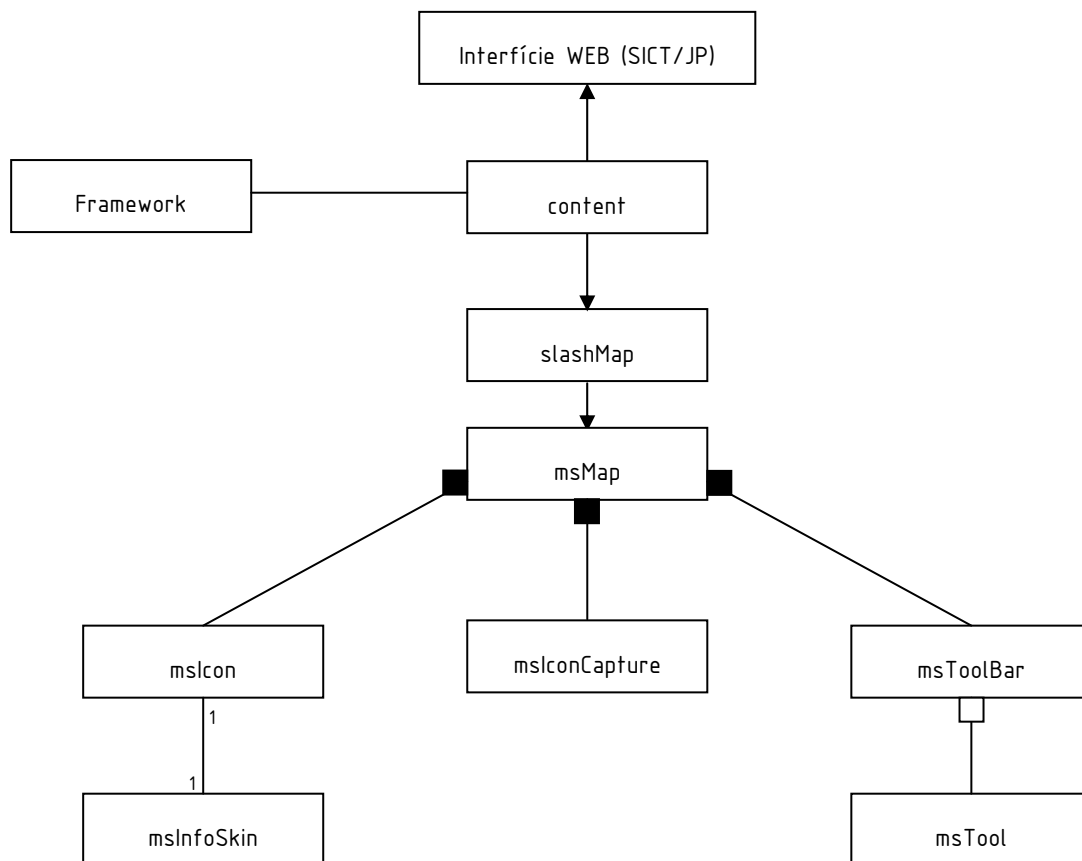
Per tal de poder comunicar el component amb la interfície web i amb el Framework, ja que és d'on obtindrem els diferents documents (imatges i fitxers XML), disposarem de Content que serà l'encarregat de realitzar aquesta comunicació.

Des de la interfície web s'iniciarà el slashMap que serà l'encarregat de gestionar els esdeveniments entre els moviments de l'usuari i el control del mapa, el msMap.

MsMap contindrà els mslcon, que són els punts, mslconCapture que són les captures, i msToolBar que serà les barres de eines.

El mslcon estarà format per la informació d'on està ubicat el POI i pel msInfoSkin que serà la seva representació gràfica de la informació.

Per l'altre banda msToolBar estarà format per les msTool que seran les eines pròpiament dites.



*Figura 19:* La figura mostra el diagrama de classes.

## Diagrames de Seqüència

## 3.2.2

### Iniciar el mapa

### 3.2.2.1

L'usuari iniciarà el mapa i la classe Javascript *slashMap.js*, que serà la interfície de comunicació entre l'usuari i el mapa, *msCross.js*.

Seguidament la interfície *slashMap.js* inicialitzarà els paràmetres bàsics del visualitzador, carregarà els servidors disponibles mitjançant la lectura d'un arxiu XML contingut dins de la WebApp.

A continuació s'obtidran les *capabilities* del servidor WMS que es carregarà per defecte o al inici.

Un cop carregades les *capabilities*, es procedirà a inicialitzar la classe *msCross.js*, que carregats els paràmetres inicials i fetes totes les inicialitzacions corresponents, procedirà a obtenir la primera imatge del mapa.

Aquesta imatge que s'haurà de carregar per primera vegada vindrà establerta per un array de dades i passat com a paràmetres en la inicialització del *msCross.js*

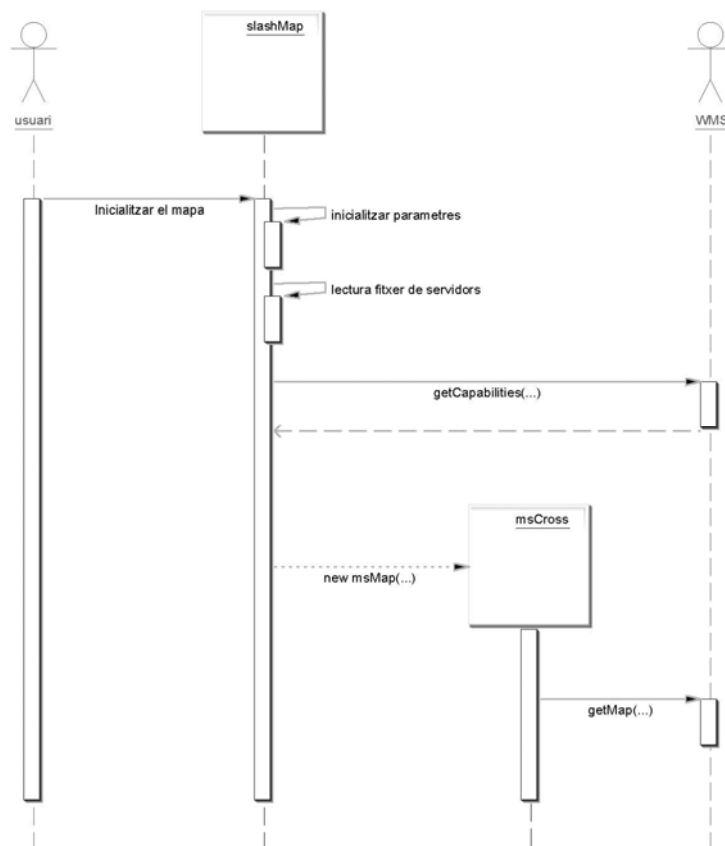


Figura 20: La figura mostra el diagrama de seqüència d'iniciar el mapa.

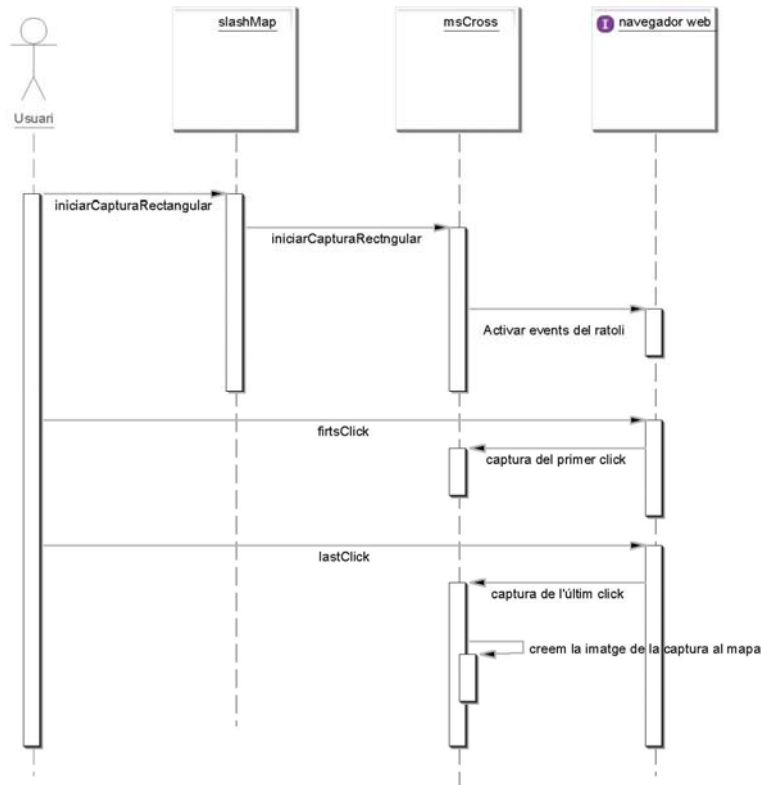
## Crear una captura rectangular

### 3.2.2.2

L'usuari decidirà iniciar la captura rectangular, si no existeix cap operació en curs podrem realitzar l'operació, amb el que l'usuari iniciarà la captura a la interfície de comunicació amb el msCross i l'slashMap.

SlashMap realitzarà una crida a msCross per tal de comprovar que es pugui realitzar la captura Rectangular. En cas de poder dur a terme l'operació msCross activarà la captura dels esdeveniments del ratolí en la interfície web.

En aquest punt msCross restarà a l'espera que l'usuari faci click en dos zones del mapa. Un cop msCross capturi aquests dos esdeveniments es mostrarà el resultat de la captura en una imatge sobreposada al mapa.



*Figura 21:* La figura mostra el diagrama de seqüència de crear una captura rectangular.

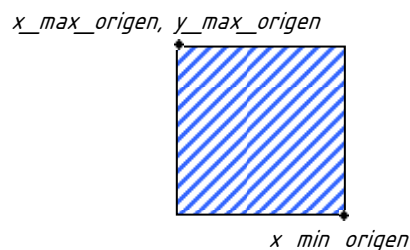
## Problema transformar una posició GPS píxel

## 4.1

Un dels problemes amb els que ens trobarem serà el de transformar una posició GPS a un píxel de la imatge. Hem de tenir present que el càlcul ha de tenir en compte la funcionalitat del zoom que provoca l'alteració de la imatge, el que significa que el càlcul serà el mateix però cal tenir en compte que els paràmetres si que variaran i es un fet que cal tenir present.

Hem de tenir en compte que no és el mateix la transformació dels punts si ho fem en l'eix d'abscisses(x) o en l'eix d'ordenades (y).

Quan creem el mapa passem 2 posicions GPS, tot i que només en farem servir 3 de les 4 components, ja que l'altre la calcularem, per tal de mantenir les proporcions del mapa:



**Figura 20:** La imatge mostra l'espai contingut en un píxel.

$puntX = longitudDelaPosicioGPSaConvertir$  i  $puntY = latitutDelaPosicioGPSaConvertir$

$y\_max\_origen = ( alturaDelMapa/ampladaDelMapa ) * ( x\_max\_origen - x\_min\_origen )$   
 $+ y\_min\_origen;$

$x\_max\_origen$  i  $x\_min\_origen$  correspon a la longitud i  $y\_max\_origen$  i  $y\_min\_origen$  correspon a la latitud

$x \text{ píxel} = Math.round ( ampladaDelMapa * ( puntX - x\_min\_origen ) / ( x\_max\_origen - x\_min\_origen ) );$

$y \text{ píxel} = Math.round( alturaDelMapa * ( y\_min\_origen - puntY ) / ( y\_max\_origen - y\_min\_origen ) );$

Caldrà utilitzar el mètode round per arrodonir el resultat a un nombre enter ja que la posició d'un píxel correspon a un enter.

Al apartat de vocabulari disposarem de més informació referent a UTM i GPS.

## Problema transformar un píxel a una posició GPS

## 4.2

El problema anterior també està relacionat amb un altre problema, i es el fet que també hem de ser capaços de transformar un píxel a una posició GPS. Aquest fet no seria tant complicat sinó fos perquè el visualitzador de mapes disposa de la funcionalitat del zoom. Per tant a cada moviment de zoom necessitem recalculer tots els valors que apareixen sobre del visualitzador.

Primer de tot hem d'obtenir la representació espacial de l'amplada del píxel:

$ampladaDelPíxelX = ( x\_max\_origen - x\_min\_origen ) / ampladaDelMapa$

$alturaDelPíxelY = ( y\_max\_origen - y\_min\_origen ) / alturaDelMapa$

Els valors de  $x\_max\_origen$  i  $x\_min\_origen$  corresponen a les dades actuals, no fan referència als valors de la inicialització

*ampladaDelPixelX* = píxel\_x \* ampladaDelPíxelX

*alturaDelPixelY* = píxel\_y \* alturaDelPíxelY

*longitud* = ampladaDelPixelX + x\_min\_origen

*latitud* = alturaDelPixelX + y\_min\_origen

## Problema al realitzar una captura circular

## 4.3

El fet de crear una captura circular suposa un càlcul una mica especial ja que, hem de crear la representació gràfica del cercle sobre el mapa i també hem d'obtenir dades del radi de la circumferència en metres així com obtenir el centre de la circumferència.

L'usuari realitzarà un primer click en un punt del mapa i s'allunyarà formant un radi, així d'aquesta manera tindrem dos posicions GPS, la del punt central (primer click) i un punt del cercle (l'últim punt).

Per poder crear la representació gràfica del cercle, el que farem serà obtenir la distància entre els dos punts, i ho farem en valor de píxels, ja que així d'aquesta manera ens estalviem de realitzar càlculs.

$$dist = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

*dist* = *Math.sqrt*(

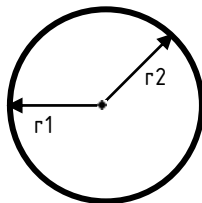
*Math.pow*(*\_capture\_x\_first* - *\_capture\_x\_last*, 2) +

*Math.pow*(*\_capture\_y\_first* - *\_capture\_y\_last*, 2)

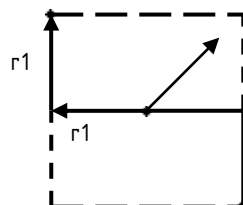
);

A continuació el que farem serà convertir el centre del radi a coordenades UTM, el motiu pèl que fem aquesta conversió és perquè en UTM enlloc de treballar amb graus minuts i segons, es treballa en metres, cosa que ens ajudarà alhora de poder obtenir un rectangle que serà la representació del cercle.

La distància del centre al punt final es el radi:



Com podem comprovar a la imatge anterior r1 i r2 tenen la mateixa longitud, amb el que si fem la distància del centre al punt exterior de la circumferència obtenim el radi. Gràcies al radi podem construir el quadrat que conté el cercle en qüestió. Necessitem aquest rectangle per tal de poder formar la circumferència, ja que disposem d'una imatge que representarà la circumferència, però aquesta imatge serà un quadrat, d'aquí la necessitat de formar aquest quadrat.



Seguidament cald transformar r1+r1 (altura i amplada del quadrat) en píxels. Amb el valor de l'amplada i de l'altura podrem construir una imatge amb les dimensions sol·licitades per l'usuari.

La necessitat d'utilitzar UTM es que tal i com hem implementat el sistema de representar punts a la pantalla, necessitem obtenir el punt concret on ubicar la imatge de la representació del cercle.

## Problema per obtenir les imatges

### 4.4

Un dels problemes amb els que ens trobarem serà obtenir el fitxer xml de les *capabilities* i les imatges del mapa, de fet el problema és com obtenir un objecte extern a la nostra aplicació, per això, i gràcies a que utilitzarem el Framework de l'empresa, crearem una JSP per tal d'obtenir i ficar el contingut obtingut a la *request* per tal de poder-los utilitzar.

Per això el que farem serà des de l'aplicació (*slashMap* i *msCross*) construir la URL on haurem d'anar a sol·licitar l'objecte en qüestió.

Dins del projecte disposarem d'una JSP, anomenada **content.jsp** que serà l'encarregada de realitzar aquestes operacions.

Crearem la URL que sens passarà com a paràmetre:

```
url = new URL(request.getParameter("url"));
InputStream in = (InputStream)url.getContent();
OutputStream outr = response.getOutputStream();
FileUtils.pipe(in, outr);
```

*FileUtils.pipe(InputStream i, OutputStream o)* és una funció del Framework que s'encarregà de col·locar fitxers dins de la request del sistema per a que les JSP's puguin obtenir aquestes dades.

## Superposició de POI's

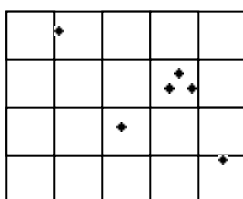
### 4.5

Un dels problemes amb els que també hem hagut de fer front és que el mapa ha d'estar capacitat per carregar fins a 5000 punts, sense que això suposi una limitació per al sistema.

El problema apareix quan estem treballant amb una captura de zoom molt allunyada de la terra i amb molts punts sobre el mapa. Com que hem de representar els punts, arriba un moment en el que els POI's comencen a ajuntar-se i degut al fet que cal representar-los dins d'un píxel, aquests queden dins del mateix píxel, situació provocada perquè un píxel representa un territori molt extens.

Com a resultat d'aquest defecte, provocat pel zoom, la superposició de POI's fa que només es pugui mostrar la informació del POI superior (aquell que restarà sobre de tots els de la pila).

La solució que es va pensar va ser tessellar el mapa. Tessellar el mapa significa crear una sèrie de requadres (tessel·les). El motiu pel que farem això és per tal de poder construir una estructura que ens permeti determinar quins punts estan dins d'una mateixa zona i poder crear informació de punts paginats.



**Figura 20:** La figura mostra una representació del mapa tessellat amb els punts que quedarien ubicats dins d'aquestes tessell·les.

Cal tenir present que la subdivisió del mapa en requadres no implica que un requadre equivalgui a un píxel, just al contrari, per tal que el sistema pugui mostrar la informació de tots els punts, un requadre equivaldrà a uns 8 píxels. Amb aquest tamany tenim la seguretat que es podrà visualitzar la informació dels punts correctament.

Com a resultat d'haver creat aquest procés obtindrem uns nous marcs d'informació:



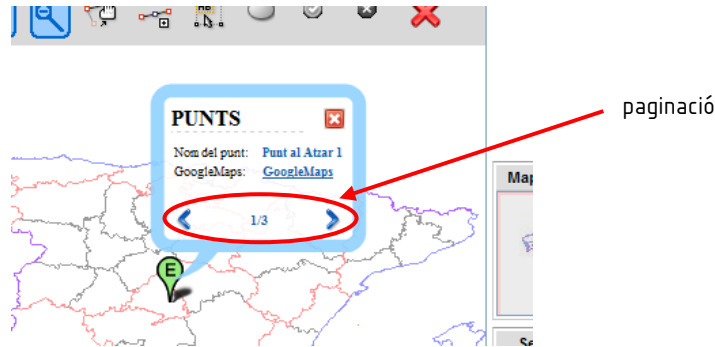


Figura 21: Resultat de la paginació dins dels marcs d'informació.

## Fitxer servers.xml

4.6

Per tal de poder carregar un llistat de servidors, el component inclou un fitxer anomenat *servers.xml*

Cal destacar que el sistema esta pensat per treballar en el fus horari d'Espanya i la utilització d'un mapa que treballi en un fus horari diferent pot provocar una lleugera alteració en els resultats obtinguts.

Format del fitxer *servers.xml*:

- Server Etiqueta que mostrarà les dades bàsiques d'un servidor WMS.

```
<server> ..... data ... </server>
```

- Descriptor: Nom que apareixerà en el component.

```
<descriptor> Cadastre </descriptor>
```

- Baseurl: URL del servidor WMS.

```
<baseurl> http://ovc.catastro.meh.es/Cartografia/WMS/ServidorWMS.aspx </baseurl>
```

- Defaultlayer: Paràmetre que indicarà quina es la capa a carregar per defecte.

```
<defaultlayer> Todas </defaultlayer>
```

- Minimapa: Paràmetre que ens indicarà si es o no necessari carregar el minimapa.

```
<minimapa> true </minimapa>
```

## Integració de Google Maps dins la informació dels POI's 4.7

El client va sol·licitar que en la mesura de lo possible intentéssim integrar Google Maps en el visualitzador i es va optar per incloure en tots els punts, dins de l'apartat d'informació que apareix quan un POI es seleccionat.

Cal tenir present que des de l'empresa ja es disposa de la llicència *Google Enterprise* que permet la utilització de les aplicacions de Google en aplicacions que ven l'empresa.

Dins d'aquestes zones d'informació apareix un enllaç que ens dirigirà a Google Maps, però per poder-ho fer va ser necessari crear una altre JSP.

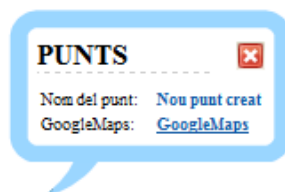


Figura 22: La Figura mostra l'enllaç a Google Maps.

Aquesta JSP, anomenada *gmaps.jsp* fa aparèixer en un popup del navegador, un Bounding box amb el POI

centrat.



Figura 23: Mostra el popup que apareixerà al clicar l'enllaç de Google Maps anterior

## Llibreries

## 4.8

Cal destacar que l'aplicació del visualitzador de mapes utilitzarà una sèrie de llibreries, per tal de millorar-ne l'eficiència així com de ajudar-nos en l'elaboració del component.

- **wz-dragdrop**: Llibreria que ens permet arrossegar imatges per la pantalla, el mapa, els POI's,...
- **dom-drag**: Llibreria que també ens ajuda alhora de realitzar els moviments de les imatges del visualitzador.
- **jkl-parsexml** i **jkl-dumper**: Aquestes llibreries ens permetran crear objectes a partir de fitxers xml, això ens serà molt útil alhora de poder interpretar els fitxers de *capabilities* que obtindrem dels diferents servidors WMS.

## Integració del visualitzador en el SICT/JP

## 4.9

Un cop desenvolupat el component arriba el moment d'integrar el visualitzador en l'aplicació mare, SICT/JP.

SICT/JP necessita de la utilitat del mapa en diferents apartats de l'aplicació i en cadascuna d'elles ha de realitzar diferents funcionalitats de les que hem hagut de desenvolupar.

SICT/JP és una aplicació que disposa d'un potent mòdul de ACL, amb el que els permisos formen part de la integració del mapa. Aquest és un dels motius pels que en la carrega del mapa es pot passar l'array de botons que volem que apareguin en el component. Així d'aquesta manera els usuaris amb menys permisos carregaran una versió de mapa amb les utilitats bàsiques, com són el zoom i els moviments de la imatge del mapa.

Durant el procés de desenvolupament, cap el final per ser més concrets, es va sol·licitar si podien coexistir dos instàncies del mapa en la mateixa pàgina. Aquest fet tot i que a priori no estava contemplat en l'anàlisi, va provocar que tinguéssim que realitzar una sèrie de canvis per tal d'evitar que els esdeveniments d'un mapa no saltessin al altre mapa.

Dutes a terme les modificacions, el component pot tenir no una ni dos sinó múltiples instàncies en una mateixa plana i funcionant totes elles correctament. Tot i que sigui possible carregar múltiples instàncies no es recomana sobrepasar les 3 ja que sinó el sistema és massa pesat i deixa de ser funcional.

A continuació podem veure alguns dels apartats on el mapa ha estat integrat satisfactòriament.

### Mòdul d'emplaçaments

### 4.9.1

En aquest apartat es gestiona l'alta i la modificació d'un emplaçament. En aquesta zona el component ha de ser capaç de permetre l'edició d'un sol punt dins del mapa, permetent la creació del punt quan el mòdul estigui en el moment de l'alta, i gestionar un punt ja creat quan estiguem en l'apartat d'edició.

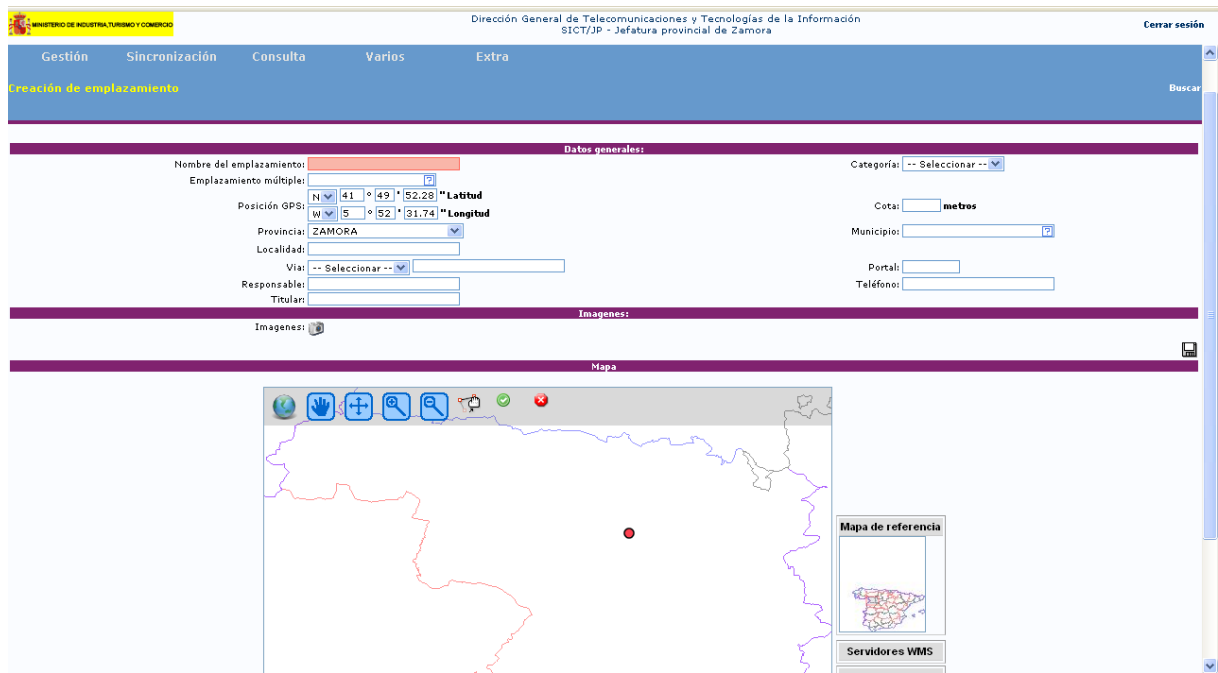


Figura 24: La figura mostra la integració del component en el mòdul d'alta d'un emplaçament.

Quan el punt estigui creat, es mostrarà la informació del emplaçament dins del marc d'informació del punt d'interès.

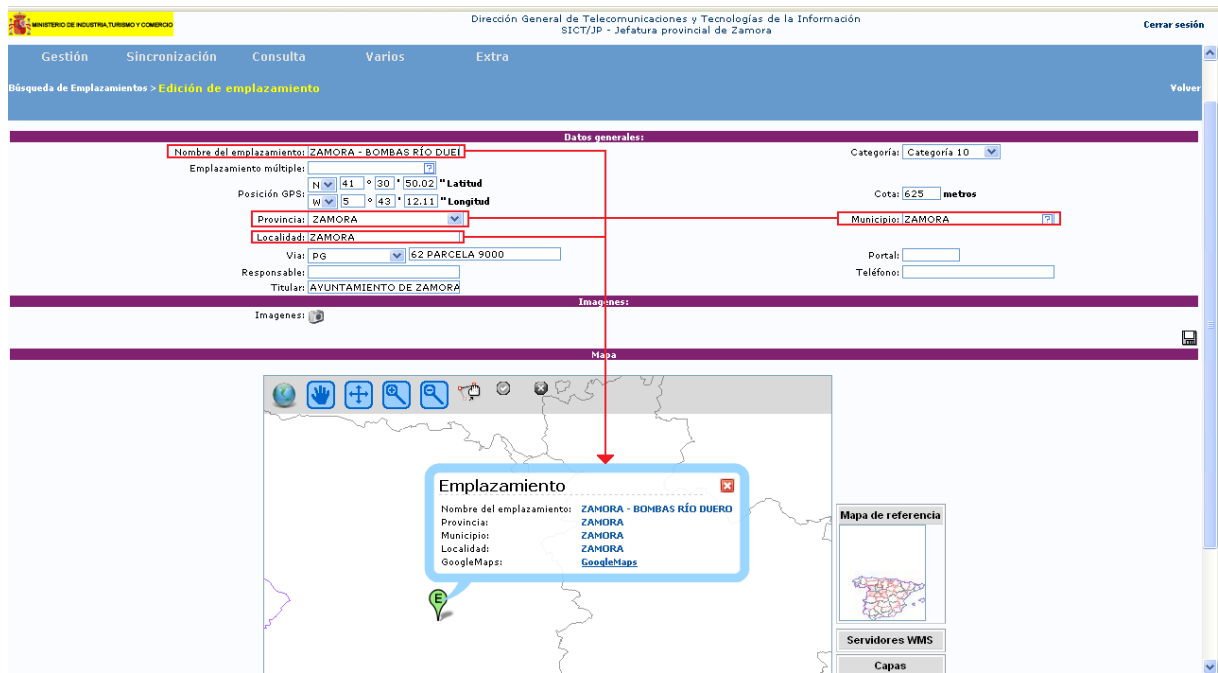


Figura 24: La figura mostra la integració del component en el mòdul d'edició d'un emplaçament.

## Mòdul de processos

## 4.9.2

El component també estarà integrat en els diferents apartats del mòdul de processos.

En aquest apartat el component se li demana que sigui capaç de carregar un nombre indeterminat de punts, un nombre que pot arribar a ser elevat.

Se li demana al component que sigui capaç de seleccionar una sèrie de punts mitjançant el mètode de selecció d'una zona rectangular.

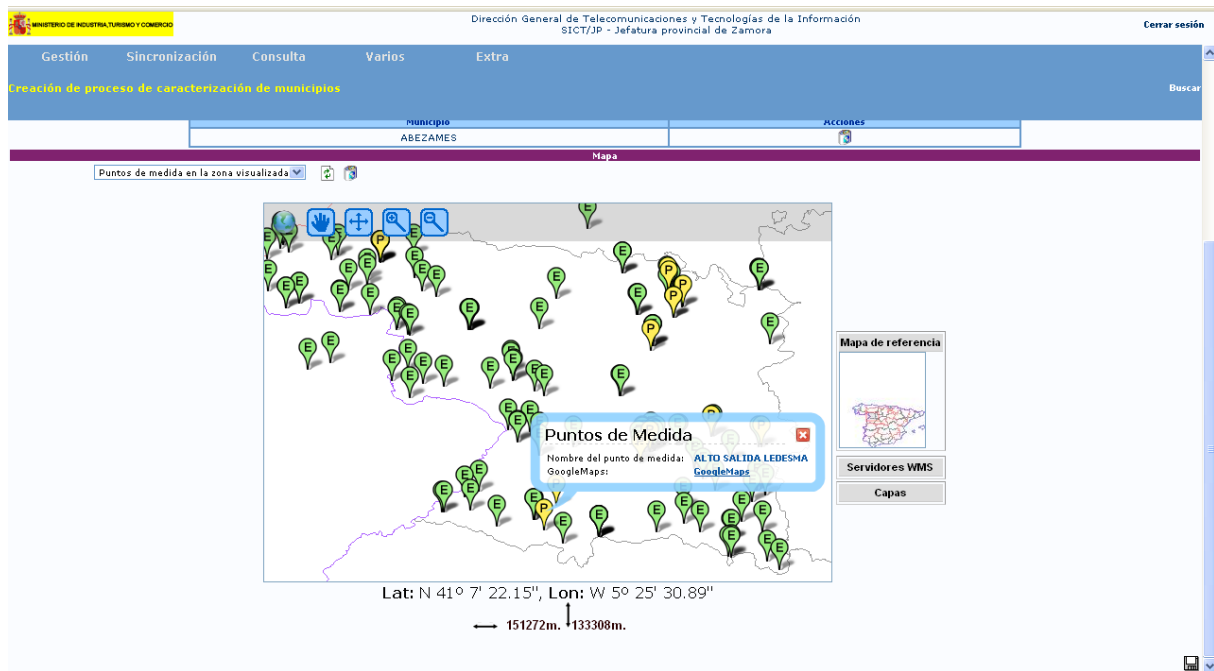


Figura 25: La figura mostra la integració del component en el mòdul de caracterització de municipis.

## Mòdul de consultes

## 4.9.3

En el mòdul de consultes també era necessari integrar el component, i més concretament, era necessari disposar-ne de dos instàncies del mapa.

El motiu principal de la necessitat de disposar de dos mapes era perquè calia fer la selecció d'una zona del mapa i que els resultats es poguessin traslladar al segon mapa.

Com podem veure en la imatge, el segon mapa (el sobreposat) s'ha aplicat l'opció de no mostrar el menú lateral ni mostrar el menú inferior, així d'aquesta manera el segon mapa fa la sensació d'aparèixer com si d'un popup es tractes.

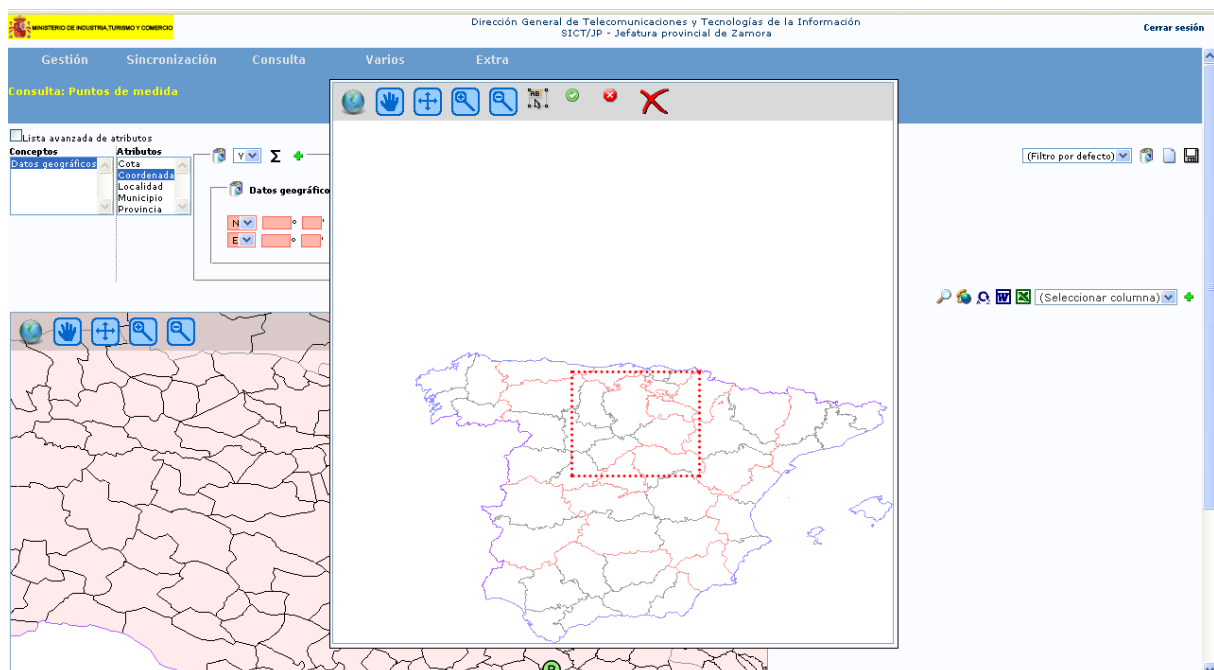
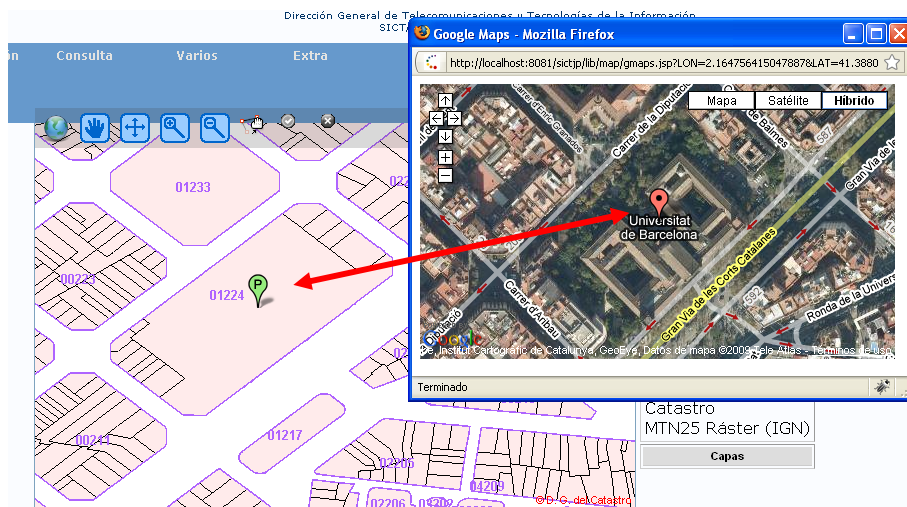


Figura 26: La figura mostra la integració del component en el mòdul de consultes.

## Google Maps vs. Visualitzador de mapes

## 4.9.4

En aquest punt podem comprovar quin es l'error entre ambdós sistemes utilitzant la integració del component en el SICT/JP i la visualització del mateix punt en Google Maps.



**Figura 27:** Comparativa dels resultats en el nostre component amb els resultats de Google Maps.

# Conclusions

5

## Resultats

5.1

En aquest apartat inclourem els resultats de les proves que hem realitzat durant la integració del component en el SICT/JP (l'única aplicació que fins ara utilitza el component).

Els test de rendiment que s'han dut a terme ens han demostrat la robustesa i eficiència del codi. Per tal d'arribar a aquestes conclusions es van realitzar diferents proves.

### Inserir un nombre elevat de POI's

5.1.1

Una de les proves de rendiment que es va realitzar va ser la de intentar inserir un nombre elevat de POI's al component.

El test es va iniciar inserint 50 punts, a continuació se'n van inserir 500 i finalment 5000. El resultat obtingut va ser sorprenent ja que el component va ser capaç de crear i ubicar en cadascun dels test, tots els POI's en un espai de temps molt curt (en el cas dels 5000 punts en un temps raonable i acceptable).

L'únic inconvenient que detectat va ser que alguns navegadors el procés s'alentia alhora d'intentar renderitzar el volum d'imatges que suposaven 5000 punts (un punt dos imatges, el POI i la ombra).

### Test de Navegadors Web

5.1.2

Un dels tests que calia dur a terme era el *crossbrowser*, és a dir el que en català significaria que el component sigui 100% funcional en la majoria de navegadors web.

Per comprovar si el component complia la propietat *crossbrowser* varem investigar quins eren els navegadors més utilitzats avui dia per la població que navega per Internet.

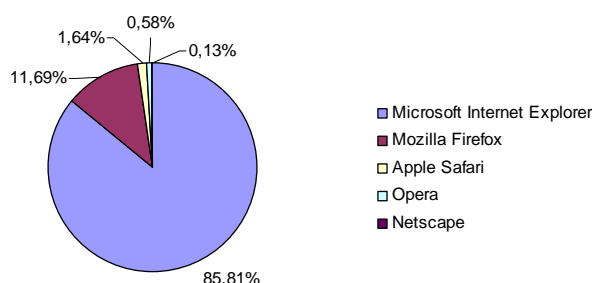


Figura 28: La figura mostra la utilització dels diferents navegadors a nivell mundial.

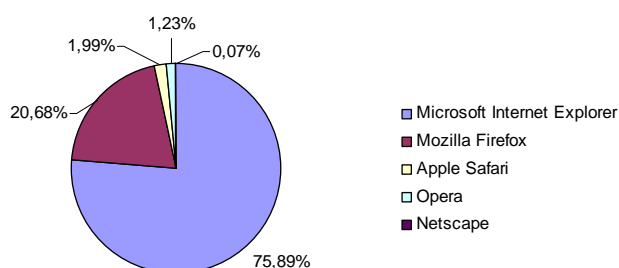


Figura 29: La figura mostra la utilització dels diferents navegadors a Espanya.

Si mirem les dades dels gràfics, entre Microsoft Internet Explorer i Mozilla Firefox s'assoleix un 97,5% d'utilització, és a dir, pràcticament 97 persones de cada 100 utilitzen un d'aquests navegadors, això a nivell mundial però pel que fa referència a nivell espanyol es un 96,5% pràcticament les mateixes dades.

Analitzades aquestes dades es va decidir testejar el component en Microsoft Internet Explorer i en Mozilla Firefox tenint en compte, que si sobrava temps al final del projecte s'intentaria provar-ho amb algun dels altres navegadors.

Realitzades diferents proves, executar totes les funcionalitats del mapa en ambdós navegadors, es va constatar que Mozilla Firefox era el navegador que aconseguia gestionar millor els esdeveniments que el mapa llençava però que per contra, Internet Explorer era qui renderitzava millor les imatges.

També es va poder comprovar com Internet Explorer presentava més dificultats en llegir els fitxers XML obtinguts pel component en les crides al mètode *getCapabilities*.

Acabades les proves es va determinar que el component complia amb l'especificació inicial, que el component fos *crossbrowser*.

Per demostrar l'afirmació anterior, seguidament podrem veure una sèrie de captures en diferents navegadors:

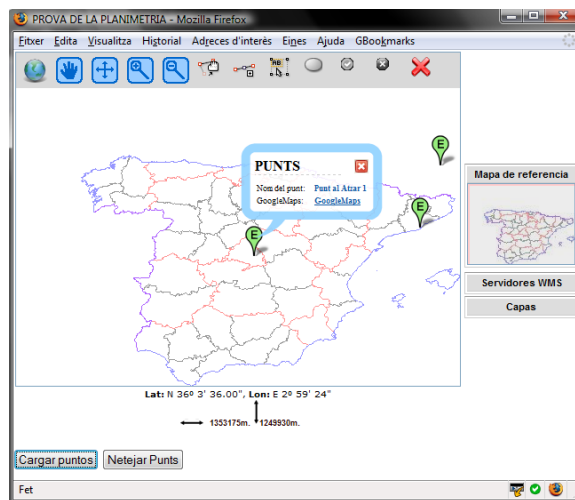


Figura 30: Captura del mapa funcionant en Mozilla Firefox 3.0

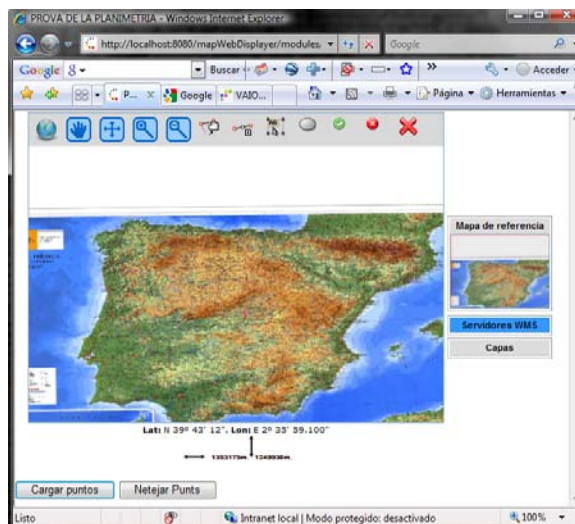


Figura 31: Captura del mapa funcionant en Microsoft Internet Explorer 7

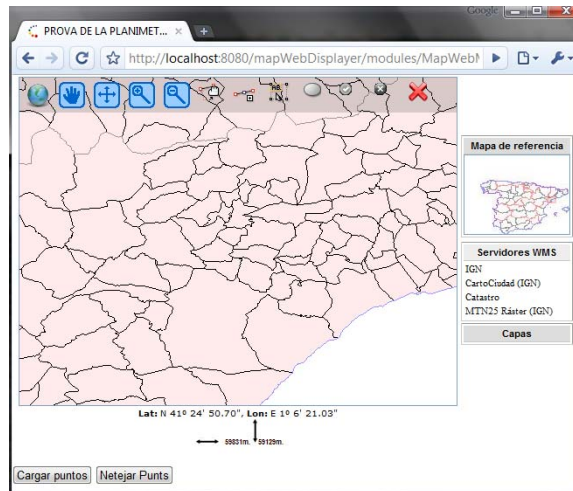


Figura 32: Captura del mapa funcionant en Google Chrome

## Línies futures

## 5.2

Una de les línies futures que queda pendent és la de intentar incloure el servei nomenclàtor dins del component.

S'ha fet una prèvia investigació de que existeix i que es el que es podria aplicar, però la falta de temps i hores ha provocat que al final es descartes seguir amb aquesta línia de treball, tot i que el component es bastant adaptable a afegir aquest apartat.

Tot hi haver estat descartat en aquest apartat intentarem descriure el protocol que hem trobat i explicar una mica que és i com funcionaria.

## WFS

## 5.2.1

WFS són les sigles de Web Feature Service o el que és el mateix en català servei de publicació d'objectes, ens permet consultar dades espacials en format vectorial codificats en *GML*. Cada servei pot controlar un o més tipologies de fenòmens, cadascun dels quals té associat un XML que descriu la seva estructura.

Per accedir i manipular aquests fenòmens geogràfic, l'estàndard WFS defineix interfícies que operen mitjançant la utilització de HTTP com a plataforma de treball distribuïda. Gràcies a aquestes interfícies, un usuari o servei web pot combinar, utilitzar i gestionar informació referent als fenòmens que constitueixen els mapes, sempre que compleixin amb els següents requisits:

1. Les interfícies han de definir-se en XML
2. S'ha d'utilitzar GML per a servir la informació sobre els fenòmens
3. Un WFS ha de ser capaç d'oferir un mínim d'un fenomen utilitzant GML

## Operacions

## 5.2.1.1

Podem definir tres operacions bàsiques de WFS:

- *GetCapabilities*.
- *DescribeFeatureType*
- *GetFeature*



## *Bibliografia*

6

- Open Geospatial Consortium.
- OpenGIS Web Map Service (WMS) Implementation Specification.
- Global Spatial Data Infrastructure.
- Infraestructura de Datos Espaciales de España.
- Oficina Virtual del Catastre.
- Google Maps API.
- Google Maps API KEY.
- Wikipedia.
- David Flanagan, JavaScript: The Definitive Guide.
- Documentació de Projectes d'Slash Solution.

## Vocabulari i Sigles

## 7.1

**API (Application Programming Interface):** La traducció literal del terme anglès és Interfície de Programació d'Aplicacions. Són doncs un conjunt de declaracions que defineixen el funcionament d'un component informàtic, per tal que qui hagi de fer ús dels seus serveis sàpiga que fa cada part del component.

Al moment de construir un sistema informàtic o llibreria de programació, per donar suport a les invocacions a serveis fetes per un altre programa, cal oferir una API, tant als programes externs (que podran usar els serveis oferts), com al programador (que disposa del manual indispensable per poder treure el màxim partit del component que ha adquirit).

Molt sovint, una API pot esdevenir part del *SDK*. L'API en sí mateixa és abstracta ja que especifica una interfície i no està lligada als detalls de implementació. Quan un programari proveeix la funcionalitat descrita per una API, llavors diem que és una implementació de l'API.

En aquesta abstracció apareixen la llista de variables públiques, funcions i procediments (o mètodes, si parlem de programació orientada a objectes), que el component informàtic ofereix.

**Crossbrowser:** Terme anglès que fa referència a la possibilitat d'una aplicació o sistema de funcionar en múltiples navegadors web, una traducció literal del terme del anglès al català seria que el sistema és capaç de saltar entre navegadors.

**Bug:** Terme anglès utilitzat en software i que fa referència a un error informàtic.

**DOM (Document Object Model):** En català Model d'objectes del Documents, es essencialment un model computacional a través del que els programes i *scripts* poden accedir i modificar el contingut, estructura i estil dels documents HTML i XML.

**Drag & Drop:** Expressió informàtica (arrossegar i deixar) que es refereix a l'acció de moure amb el ratolí objectes d'una finestra a una altre o entre parts de la mateixa finestra. Els objectes arrossegats són habitualment arxius, però també poden ser arrossegats altres elements en funció del programa.

**Framework:** És una estructura de suport definida en la qual un altre projecte de software pot ser organitzat i desenvolupat. Típicament, un Framework pot incloure suport de programes, biblioteques i un llenguatge interpretat entre altres software per ajudar a desenvolupar i unir diferents components d'un projectes.

Un Framework representa doncs una arquitectura de software que modela les relacions generals de les entitats del domini. Proveeix una estructura i una metodologia de treball la qual estén i utilitza les aplicacions del domini.

**GIF (Graphics Interchange Format):** És un format gràfic utilitzat àmpliament a la web, tant per imatges com per animacions. GIF és un format sense pèrdua de qualitat per a imatges fins a 256 colors, limitats per una paleta restringida a aquest número de colors (profunditat de color superior a 8 bits), la imatge ha d'adaptar-se reduint els seus colors, produint la conseqüent disminució de la qualitat.

**Google Inc:** Corporació pública nord americana propietària de la marca **Google** i que té com a principal producte el motor de cerca del mateix nom. Va ser fundada el 7 de setembre de 1998 per dos estudiants de l'escola d'enginyeria de la Universitat d'Stanford.

Encara que el seu principal producte és el cercador, l'empresa ofereix també entre altres serveis, com una plataforma de publicitat en línia i a través de mòbil (*Google AdSense, Google AdWords...*), webmail i missatgeria instantània (*Gmail i Google Talk*), mapes en línia (*Google Maps*), ofimàtica en línia (*Google Docs...*), xarxes socials, vídeos en línia (*Youtube, Google Video...*) així com moltes més solucions online.

**GPS (Global Positioning System):** És un Sistema Global de navegació per satèl·lit que permet determinar en tot el món la posició d'una persona, un vehicle, un objecte,... amb una desviació aproximada de quatre metres. El sistema va ser desenvolupat i instal·lat pel departament de defensa dels Estats Units d'Amèrica. El GPS funciona mitjançant una xarxa de satèl·lits que orbiten al voltant de la terra. Quan es desitja determinar la

posició, l'aparell que s'utilitza per això localitza automàticament un mínim de quatre satèl·lits, dels quals rep una senyal indicant la posició i el rellotge de cadascun d'ells. Sobre la base d'aquests senyals, l'aparell sincronitza el rellotge del GPS i calcula el retard de les senyals, és a dir, la distància al satèl·lit. D'aquesta manera s'aconsegueix realitzar una triangulació i per tant, podem determinar la posició en la que ens trobem.

**JPEG (Joint Photographic Experts Group):** És un algoritme dissenyat per a comprimir imatges amb 24 bits de profunditat o en escala de grisos. És doncs, per a imatges amb més qualitat, un algoritme de compressió amb pèrdua de dades, el que significa que al descomprimir les imatges no s'obté exactament la mateixa imatge que teníem abans de la compressió.

**Llenguatge script o Llenguatge de scripting:** És un llenguatge de programació que controla aplicacions. Els scripts són executats directament des del seu codi font, que generalment són fitxers de text que contenen llenguatges de marcatge específics.

**Nomenclàtor:** Un nomenclàtor és una relació de totes les unitats de poblacions inferiors a municipi (nuclis de població, viles, parròquies barris, urbanitzacions,...). Aquest llistat conté les entitats singulars de població, nuclis i disseminats de cada un dels municipis a fi de permetre conèixer de quina manera s'assenta la població en els mateixos. Per a cada una d'aquestes unitats conté la població i dades de caràcter geogràfic.

**OGC (Open Geospatial Consortium):** Va ser creat al 1994 i agrupa a més de 250 organitzacions públiques i privades. El seu fi és la definició de estàndards oberts i interoperables dins dels Sistemes d'Informació Geogràfica. Aconsegueix acords entre les diferents empreses del sector que possibiliten la interpolació dels seus sistemes geo-processament i facilitar l'intercanvi de la informació geogràfica en benefici dels usuaris. Antigament era conegut com a **Open GIS Consortium**.

**PNG (Portable Network Graphics):** És un format gràfic basat en un algoritme de compressió amb pèrdua per a bitmaps no subjecte a cap patent. Aquest format va ser desenvolupat en bona part per a solucionar les deficiències del format GIF i permet emmagatzemar imatges amb una major profunditat de contrastos.

**POI:** És al que ens referirem quan parlem d'una representació geo-espacial en el mapa, en anglès correspon a les sigles de *Point of Interest* o el que és el mateix en català, Punt d'interès.

**SDK (Source Development Kit):** Terme que en català significa Kit de Desenvolupament de Codi, i que generalment és un conjunt d'eines de desenvolupament que permeten a un programador crear aplicacions per a un sistema bastant concret, per exemple certs paquets de software, Frameworks o plataformes de hardware, ordinadors, consoles, etc.

**SVG (Scalable Vector Graphics):** És un llenguatge per descriure gràfics vectorials bidimensionals, ja siguin estàtics o animats.

**Testeig:** Terme anglès que utilitzarem en aquesta memòria i que significarà o tindrà relació amb temes de prova de l'aplicació, sinònims del català serien proves, reacció, assaig, etc.

**Usability:** Terme anglès que significa usabilitat i és la característica d'un sistema que pretén ser utilitzat per:

- El tipus específics de usuari/s,
- La tasca o tasques per les quals el sistema ha estat desenvolupat.
- El context en el que es dona la interacció.

El concepte d'usabilitat es refereix a una aplicació informàtica (software) o un aparell (hardware), tot i que també pot ser aplicat a qualsevol sistema fet amb un objectiu particular.

**UTM (Universal Transverse Mercator):** En català Sistema de coordenades universal transversal de Mercator. Com el seu nom indica és un sistema de coordenades basat en la projecció geogràfica transversal de Mercator, que es constitueix com la projecció de Mercator normal, però enlloc de fer-se amb la tangent respecte l'equador, es fa la tangent respecte un meridià. A diferència del sistema de coordenades tradicional, expressades en longitud i latitud (hores minuts segons), les magnituds en el sistema UTM s'expressen en metres amb els valors de *Easting* i *Northing*.

**WMS (Web Map Service):** Com el seu nom indica significa, Servei de mapes Web. Servei definit per **OGC** (Open Geospatial Consortium) i produeix mapes de dades espacials referits de forma dinàmica a partir d'una informació geogràfica. Aquest estàndard defineix un mapa com una representació de la informació geogràfica en forma d'arxiu d'imatge digital convenient per a la exhibició en una pantalla d'ordinador. Els mapes produïts per WMS

es generen normalment en un format d'imatge *PNG*, *GIF* o *JPEG* i ocasionalment com gràfics vectorials en format *SVG*.

## CD-ROM

## 7.2

Aquest document contindrà un CD-ROM amb diferents continguts:

- Jboss.
- Codi.
- Document.

### Jboss

### 7.2.1

En el CD adjuntat disposarem d'un directori anomenat *jboss-4.0.4* que contindrà un Jboss per poder avaluar o provar el visualitzador.

Per tal de poder executar-ho correctament caldrà realitzar els següents passos, ja que el Jboss no es pot executar dins d'un CD.

#### Instal·lació del Jboss

#### 7.2.1.1

Per poder instal·lar correctament el Jboss caldrà realitzar els següents passos:

En Windows:

1. Extreure del CD el directori *jboss-4.0.4* a *c:/ \**
2. Crear un accés directe al fitxer *c:/jboss-4.0.4/bin/run.bat*.
3. Configurar l'accés directe afegint el paràmetre *-c mapWebDispalyer*.
4. Executar l'accés directe.

En Linux:

1. Extreure del CD el directori *jboss-4.0.4* a */ \* \*\**
2. Executar el Script *run.sh* passant-hi com a paràmetres *-c mapWebDispalyer*.

\* Cal tenir present que aquest directori pot ser el que nosaltres vulguem, però caldrà modificar el següent arxiu *path\_desat/jboss-4.0.4/server/mapWebDisplayer/deploy/mapWebDisplayer.war/web.xml* per tal de poder carregar les propietats del Framework. Caldrà afegir doncs un *<param-value>* amb la ubicació de les propietats *path\_desat/jboss-4.0.4/server/mapWebDisplayer/deploy/mapWebDisplayer.war/properties/mapwebdisplayer.properties*

\*\* En Linux aquest procés s'haurà de realitzar obligatòriament, ja que esta pensat per ser utilitzat des de Windows.

#### Execució del Jboss

#### 7.2.1.2

Per executar el visualitzador caldrà arrancar el Jboss com s'explica anteriorment i introduir la següent URL: <http://localhost:8080/mapWebDisplayer>. Si tot ha funcionat correctament el mapa apareixerà al nostre navegador.

Existeix un error en les traces del Jboss, provocat pel Framework, però que no afecten en el funcionament.

### Codi

### 7.2.2

Dins del CD disposem d'una carpeta amb el codi desenvolupat. Només es disposa del codi del projecte pròpiament, ja que el codi del Framework es privat.

### Documentació

### 7.2.3

També disposarem d'un directori que contindrà la documentació que esteu llegint en aquests moments.

## Llistat de servidors WMS

## 7.3

En aquest apartat disposarem d'un llistat dels diferents WMS que hem trobat a la xarxa:

- Web de la Infraestructura de Dades Espacials de Catalunya (IDEC)
- ExploreOurPla.net - Buscadors i visionats de servidors WMS
- Web de la Infraestructura de Dades Espacials de Espanya (IDEE)
- Web de la Infraestructura de Dades Espacials del Govern de La Rioja (IDERIOJA)
- Visualitzador Humboldt OGC WMS Viewer
- Visualitzador de la ciutat d'Osaka.
- OGC-Services.NET
- Cadastre

# Implementació

7.4

## Fitxer de Capabilities XML

7.4.1

```
<?xml version='1.0' encoding="UTF-8" standalone="no" ?>
<!-- El DTD (Document Type Definition) hauria d'indicar aquí la versió numèrica del WMT_MS_Capabilities -->
<!DOCTYPE WMT_MS_Capabilities SYSTEM
    http://www.digitalearth.gov/wmt/xml/capabilities\_1\_1\_1.dtd
    [
        <!-- Elements específics del servidor, estaran definits aquí. -->
        <!-- Si no és necessari pot ometre aquest aparat. -->
        <!ELEMENT VendorSpecificCapabilities EMPTY>
    ]>
<!-- Fi de la declaració DOCTYPE -->
<!-- Nota: Aquest XML és només un exemple, en un intent de mostrar totes les possibilitats. -->
<!-- El número de la versió que figura a la llista del WMT_MS_Capabilities ha de correspondre amb el del DTD declarat anteriorment -->
<WMT_MS_Capabilities version="1.1.1" updateSequence="0">
    <!-- Service Metadata -->
    <Service>
        <!-- Nom que defineix amb el servei -->
        <Name>OGC:WMS</Name>
        <!-- Títol descriptiu -->
        <Title>Acme Corp. Map Server</Title>
        <!-- Descripció del Servei-->
        <Abstract>
            WMT Map Server maintained by Acme Corporation. Contact: webmaster@wmt.acme.com. High-quality maps showing
            roadrunner nests and possible ambush locations.
        </Abstract>
        <KeywordList>
            <Keyword>bird</Keyword>
            <Keyword>roadrunner</Keyword>
            <Keyword>ambush</Keyword>
        </KeywordList>
        <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="http://hostname/" />
        <!-- Informació de contacte-->
        <ContactInformation>
            <ContactPersonPrimary>
                <ContactPerson>Jeff deLaBeaujardiere</ContactPerson>
                <ContactOrganization>NASA</ContactOrganization>
            </ContactPersonPrimary>
            <ContactPosition>Computer Scientist</ContactPosition>
            <ContactAddress>
                <AddressType>postal</AddressType>
                <Address>NASA Goddard Space Flight Center, Code 933</Address>
            </ContactAddress>
        </ContactInformation>
    </Service>
</WMT_MS_Capabilities>
```

```

    <City>Greenbelt</City>
    <StateOrProvince>MD</StateOrProvince>
    <PostCode>20771</PostCode>
    <Country>USA</Country>
  </ContactAddress>
  <ContactVoiceTelephone>+1 301 286-1569</ContactVoiceTelephone>
  <ContactFacsimileTelephone>+1 301 286-1777</ContactFacsimileTelephone>
  <ContactElectronicMailAddress>delabeau@iniki.gsfc.nasa.gov</ContactElectronicMailAddress>
</ContactInformation>
<!-- Taxes d'accés o limitacions imposades. -->
<Fees>none</Fees>
<AccessConstraints>none</AccessConstraints>
</Service>
<Capability>
  <Request>
    <GetCapabilities>
      <Format>application/vnd.ogc.wms_xml</Format>
      <DCPType>
        <HTTP>
          <Get>
            <!-- Adreça URL per invocar utilitzant HTTP GET -->
            <OnlineResource xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="simple"
              xlink:href="http://hostname:port/path" />
          </Get>
          <Post>
            <!-- Adreça URL per invocar GetCapabilities utilitzant HTTP POST, inclou l'adreça completa a la qual
              una consulta s'enviarà en format XML. Apareix aquí per a una futura expansió. -->
            <OnlineResource xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="simple"
              xlink:href="http://hostname:port/path" />
          </Post>
        </HTTP>
      </DCPType>
    </GetCapabilities>
    <GetMap>
      <Format>image/gif</Format>
      <Format>image/png</Format>
      <Format>image/jpeg</Format>
      <DCPType>
        <HTTP>
          <Get>
            <!-- Adreça URL per invocar utilitzant HTTP GET -->
            <OnlineResource xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="simple"
              xlink:href="http://hostname:port/path" />
          </Get>

```



```

        </HTTP>
    </DCPType>
</GetMap>
<GetFeatureInfo>
    <Format>application/vnd.ogc.gml</Format>
    <Format>text/plain</Format>
    <Format>text/html</Format>
    <DCPType>
        <HTTP>
            <Get>
                <OnlineResource xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="simple"
                    xlink:href="http://hostname:port/path" />
            </Get>
        </HTTP>
    </DCPType>
</GetFeatureInfo>
<DescribeLayer> <!--Opcional; utilitzat només si el SLD del WMS està habilitat-->
    <Format>application/vnd.ogc.gml</Format>
    <DCPType>
        <HTTP>
            <Get>
                <OnlineResource xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="simple"
                    xlink:href="http://hostname:port/path" />
            </Get>
        </HTTP>
    </DCPType>
</DescribeLayer>
</Request>
<Exception>
    <Format>application/vnd.ogc.se_xml</Format>
    <Format>application/vnd.ogc.se_inimage</Format>
    <Format>application/vnd.ogc.se_blank</Format>
</Exception>
<!-- Qualsevol text o etiqueta esta permesa a <VendorSpecificCapabilities>, segons sigui necessari per descriure les capacitats
    específiques del seu proveïdor. -->
<!-- Aquest exemple esta buit -->
<VendorSpecificCapabilities />
<!-- Aquest element resta absent en els WMS bàsics. -->
<UserDefinedSymbolization SupportSLD="1" UserLayer="1" UserStyle="1" RemoteWFS="1" />
<Layer>
    <Title>Acme Corp. Map Server</Title>
    <SRS>EPSG:4326</SRS> <!-- Totes les capes estan disponibles per aquest SRS -->
    <AuthorityURL name="DIF_ID">
        <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"

```

```

        xlink:href="http://gcmd.gsfc.nasa.gov/difguide/whatisadif.html" />
</AuthorityURL>
<Layer>
  <!-- Aquesta capa te una capa pare, amb el que podem comprovar un sistema de capes. -->
  <Name>ROADS_RIVERS</Name>
  <Title>Roads and Rivers</Title>
  <!-- Veure les especificacions per les característiques de les capes heretades. -->
  <SRS>EPSG:26986</SRS> <!--SRS adicional per aquesta capa -->
  <LatLonBoundingBox minx="-71.63" miny="41.75" maxx="-70.78" maxy="42.90"/>
  <!-- resx i resy indiquen a continuació els atributs X i Y de la resolució espacial per aquest SRS-->
  <BoundingBox SRS="EPSG:4326" minx="-71.63" miny="41.75" maxx="-70.78" maxy="42.90" resx="0.01"
    resy="0.01"/>
  <BoundingBox SRS="EPSG:26986" minx="189000" miny="834000" maxx="285000" maxy="962000" resx="1"
    resy="1" />
  <!-- Valors opcionals: Title, URL i Logotip del proveïdor. -->
  <Attribution>
    <Title>State College University</Title>
    <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
      xlink:href="http://www.university.edu/" />
    <LogoURL width="100" height="100">
      <Format>image/gif</Format>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
        xlink:href="http://www.university.edu/icons/logo.gif" />
    </LogoURL>
  </Attribution>
  <!-- Identificador que es defineix a AuthorityURL -->
  <Identifier authority="DIF_ID">123456</Identifier>
  <FeatureListURL>
    <Format>application/vnd.ogc.se_xml</Format>
    <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
      xlink:href="http://www.university.edu/data/roads_rivers.gml" />
  </FeatureListURL>
  <Style>
    <Name>USGS</Name>
    <Title>USGS Topo Map Style</Title>
    <Abstract>Features are shown in a style like that used in USGS topographic maps.</Abstract>
    <!-- Imatge d'una llegenda d'un capa en aquest estil -->
    <LegendURL width="72" height="72">
      <Format>image/gif</Format>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
        xlink:href="http://www.university.edu/legends/usgs.gif" />
    </LegendURL>
    <!-- Un full d'estil XSL descriu les característiques GML dades prestats a crear un mapa d'aquesta capa. -->
    <StyleSheetURL>

```

```

        <Format>text/xsl</Format>
        <OnlineResource xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="simple"
            xlink:href="http://www.university.edu/stylesheets/usgs.xsl" />
    </StyleSheetURL>
</Style>
<ScaleHint min="4000" max="35000"></ScaleHint>
<Layer queryable="1">
    <Name>ROADS_1M</Name>
    <Title>Roads at 1:1M scale</Title>
    <Abstract>Roads at a scale of 1 to 1 million.</Abstract>
    <KeywordList>
        <Keyword>road</Keyword>
        <Keyword>transportation</Keyword>
        <Keyword>atlas</Keyword>
    </KeywordList>
    <Identifier authority="DIF_ID">123456</Identifier>
    <!-- Metadates especifics per aquesta capa. Metadades FGDC que s'ofereixen en dos formats. -->
    <MetadataURL type="FGDC">
        <Format>text/plain</Format>
        <OnlineResource xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="simple"
            xlink:href="http://www.university.edu/metadata/roads.txt" />
    </MetadataURL>
    <MetadataURL type="FGDC">
        <Format>text/xml</Format>
        <OnlineResource xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="simple"
            xlink:href="http://www.university.edu/metadata/roads.xml" />
    </MetadataURL>
    <!-- A més dels especificats en l'estil de la capa del pares, aquesta capa es troba disponible en aquest estil-->
    <Style>
        <Name>ATLAS</Name>
        <Title>Road atlas style</Title>
        <Abstract>Roads are shown in a style like that used in a commercial road atlas.</Abstract>
        <LegendURL width="72" height="72">
            <Format>image/gif</Format>
            <OnlineResource xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="simple"
                xlink:href="http://www.university.edu/legends/atlas.gif" />
        </LegendURL>
    </Style>
</Layer>
<Layer queryable="1">
    <Name>RIVERS_1M</Name>
    <Title>Rivers at 1:1M scale</Title>
    <Abstract>Rivers at a scale of 1 to 1 million.</Abstract>
    <KeywordList>
        <Keyword>river</Keyword>

```

```

        <Keyword>canal</Keyword>
        <Keyword>waterway</Keyword>
    </KeywordList>
</Layer>
</Layer>
<Layer queryable="1">
    <Title>Weather Forecast Data</Title>
    <SRS>EPSG:4326</SRS> <!--No afecta la repetició d'aquest SRS -->
    <LatLonBoundingBox minx="-180" miny="-90" maxx="180" maxy="90" />
    <!--Aquestes dades meteorològiques es troben disponibles tots els dies des de 01/01/1999 a 22/08/2000 -->
    <Dimension name="time" units="ISO8601" />
    <Extent name="time" default="2000-08-22">1999-01-01/2000-08-22/P1D</Extent>
    <Layer>
        <Name>Clouds</Name>
        <Title>Forecast cloud cover</Title>
    </Layer>
    <Layer>
        <Name>Temperature</Name>
        <Title>Forecast temperature</Title>
    </Layer>
    <Layer>
        <Name>Pressure</Name>
        <Title>Forecast barometric pressure</Title>
        <!--La pressió és variable a diferents altures. EPSG:5030 és WGS84 el·lipsoide i les unitats en metres -->
        <Dimension name="time" units="ISO8601" />
        <Dimension name="elevation" units="EPSG:5030" />
        <Extent name="time" default="2000-08-22">1999-01-01/2000-08-22/P1D</Extent>
        <Extent name="elevation" default="0" nearestValue="1">0,1000,3000,5000,10000</Extent>
    </Layer>
</Layer>
<!--Exemple d'una capa que és un mapa de mida fixa, que no pot ser transparent ni subconjunt-->
<Layer opaque="1" noSubsets="1" fixedWidth="512" fixedHeight="256">
    <Name>ozone_image</Name>
    <Title>Global ozone distribution (1992)</Title>
    <LatLonBoundingBox minx="-180" miny="-90" maxx="180" maxy="90" />
    <Extent name="time" default="1992">1992</Extent>
</Layer>
<!--Exemple d'una capa que es va originar a partir de WMS i un altre ha estat "en cascada" per aquest WMS. -->
<Layer cascaded="1">
    <Name>population</Name>
    <Title>World population, annual</Title>
    <LatLonBoundingBox minx="-180" miny="-90" maxx="180" maxy="90" />
    <Extent name="time" default="2000">1990/2000/P1Y</Extent>
</Layer>

```

```

    </Layer>
  </Capability>
</WMT_MS_Capabilities>

```

## Funcions de conversió de UTM a GPS

## 7.4.2

```

utm2gps = function(p_north, p_east, p_zone, p_utmletter) {
  var RAD2DEG = 180 / Math.PI;
  var DEG2RAD = Math.PI / 180;
  var POLAR_RADIUS = 6356911.9;
  var EARTH_RADIUS = 6378388;
  var EARTH_ECCENTRICITY = (1.0 - ((POLAR_RADIUS * POLAR_RADIUS) / (EARTH_RADIUS * EARTH_RADIUS)));
  var latitude = null, longitude = null;
  var N = parseFloat(p_north);
  var E = parseFloat(p_east);
  var zone = p_zone;
  var L = p_utmletter;
  var k0 = 0.9996; // escala
  var x = E - 500000;
  var y = N;
  zone = 30; // ED50 centrat en el FUS 30
  if(L=="A" || L=="B" || L=="C" || L=="D" || L=="E" || L=="F" || L=="G" || L=="H" || L=="I" || L=="J" || L=="K" || L=="L"){
    y -= 1000000.0;
  }
  var longorigin = (parseFloat(zone) - 1) * 6 - 180 + 3;
  var eccPrimeSquared = (EARTH_ECCENTRICITY) / (1 - EARTH_ECCENTRICITY);
  var M = y / k0;
  var mu = M / (EARTH_RADIUS * (1 - EARTH_ECCENTRICITY / 4 - 3 * EARTH_ECCENTRICITY * EARTH_ECCENTRICITY / 64 -
    5 * EARTH_ECCENTRICITY * EARTH_ECCENTRICITY * EARTH_ECCENTRICITY / 256));
  var e1 = (1 - Math.sqrt(1 - EARTH_ECCENTRICITY)) / (1 + Math.sqrt(1 - EARTH_ECCENTRICITY));
  var phi1rad = mu + (3 * e1 / 2 - 27 * e1 * e1 * e1 / 32) * Math.sin(2 * mu) + (21 * e1 * e1 / 16 - 55 * e1 * e1 * e1 / 32) * Math.sin(4 * mu) +
    (151 * e1 * e1 * e1 / 96) * Math.sin(6 * mu);
  var N1 = EARTH_RADIUS / Math.sqrt(1 - EARTH_ECCENTRICITY * Math.sin(phi1rad) * Math.sin(phi1rad));
  var T1 = Math.tan(phi1rad) * Math.tan(phi1rad);
  var C1 = EARTH_ECCENTRICITY * Math.cos(phi1rad) * Math.cos(phi1rad);
  var R1 = EARTH_RADIUS * (1 - EARTH_ECCENTRICITY) / (Math.pow((1 - EARTH_ECCENTRICITY * Math.sin(phi1rad) *
    Math.sin(phi1rad)), 1.5));
  var D = x / (N1 * k0);
  latitude = parseFloat(phi1rad - (N1 * Math.tan(phi1rad) / R1) * (D * D / 2 - (5 + 3 * T1 + 10 * C1 - 4 * C1 * C1 - 9 * eccPrimeSquared) * D * D *
    D * D / 24 + (61 + 90 * T1 + 298 * C1 + 45 * T1 * T1 - 252 * eccPrimeSquared - 3 * C1 * C1) * D * D * D * D * D / 720));
  latitude = parseFloat(parseFloat(latitude) * RAD2DEG);
  longitude = parseFloat((D - (1 + 2 * T1 + C1) * D * D * D / 6 + (5 - 2 * C1 + 28 * T1 - 3 * C1 * C1 + 8 * eccPrimeSquared +
    24 * T1 * T1) * D * D * D * D / 120) / Math.cos(phi1rad));
  longitude = parseFloat(longorigin + parseFloat(longitude) * RAD2DEG);
  return new Array(longitude, latitude); // retornem un array amb la longitud i la latitud obtingudes
}

```

}

## Funcions de conversió de GPS a UTM

## 7.4.3

```
gps2utm = function(p_x, p_y) {
  var DEG2RAD = Math.PI / 180;
  var POLAR_RADIUS = 6356911.9;
  var EARTH_RADIUS = 6378388;
  var EARTH_ECCENTRICITY = (1.0 - ((POLAR_RADIUS * POLAR_RADIUS) / (EARTH_RADIUS * EARTH_RADIUS)));
  var longitude = parseFloat(p_x);
  var latitude = parseFloat(p_y);
  var long2 = longitude - Math.floor((longitude + 180) / 360) * 360;
  var zone = 30; // ED50 centrat en el FUS 30
  var lat_radian = DEG2RAD * latitude;
  var long_radian = DEG2RAD * long2;
  var k0 = 0.9996; // escala
  var longorigin = (zone - 1) * 6 - 180 + 3;
  var longoriginradian = DEG2RAD * longorigin;
  var eccentprime = EARTH_ECCENTRICITY / (1 - EARTH_ECCENTRICITY);
  var N = EARTH_RADIUS / Math.sqrt(1 - EARTH_ECCENTRICITY * Math.sin(lat_radian) * Math.sin(lat_radian));
  var T = Math.tan(lat_radian) * Math.tan(lat_radian);
  var C = eccentprime * Math.cos(lat_radian) * Math.cos(lat_radian);
  var A = Math.cos(lat_radian) * (long_radian - longoriginradian);
  var M = EARTH_RADIUS * ( ( 1 - EARTH_ECCENTRICITY/4 - 3*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY/64
    - 5*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY/256 ) * lat_radian
    - ( 3*EARTH_ECCENTRICITY/8 + 3*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY/32
    + 45*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY/1024 ) * Math.sin(2 * lat_radian)
    + ( 15*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY/256 +
    45*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY/1024 ) * Math.sin(4 * lat_radian)
    - ( 35*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY*EARTH_ECCENTRICITY/3072 ) * Math.sin(6 * lat_radian) );
  var utm_easting = k0 * N * ( A + ( 1 - T + C ) * A * A * A / 6 + ( 5 - 18 * T + T * T + 72 * C - 58 * eccentprime ) * A * A * A * A * A / 120 ) + 500000.0;
  var utm_northing = k0 * ( M + N * Math.tan(lat_radian) * ( A * A / 2 + ( 5 - T + 9 * C + 4 * C * C ) * A * A * A * A * A / 24 +
    ( 61 - 58 * T + T * T + 600 * C - 330 * eccentprime ) * A * A * A * A * A * A / 720 ));
  if (latitude < 0) utm_northing += 10000000.0;
  var utm_letter;
  if ( 84 >= parseFloat(latitude) && parseFloat(latitude) >= 72) utm_letter = 'X';
  else if ( 72 > parseFloat(latitude) && parseFloat(latitude) >= 64) utm_letter = 'W';
  else if ( 64 > parseFloat(latitude) && parseFloat(latitude) >= 56) utm_letter = 'V';
  else if ( 56 > parseFloat(latitude) && parseFloat(latitude) >= 48) utm_letter = 'U';
  else if ( 48 > parseFloat(latitude) && parseFloat(latitude) >= 40) utm_letter = 'T';
  else if ( 40 > parseFloat(latitude) && parseFloat(latitude) >= 32) utm_letter = 'S';
  else if ( 32 > parseFloat(latitude) && parseFloat(latitude) >= 24) utm_letter = 'R';
  else if ( 24 > parseFloat(latitude) && parseFloat(latitude) >= 16) utm_letter = 'Q';
  else if ( 16 > parseFloat(latitude) && parseFloat(latitude) >= 8) utm_letter = 'P';
  else if ( 8 > parseFloat(latitude) && parseFloat(latitude) >= 0) utm_letter = 'N';
}
```

```

else if ( 0 > parseFloat(latitude) && parseFloat(latitude) >= -8)      utm_letter = 'M';
else if ( -8 > parseFloat(latitude) && parseFloat(latitude) >= -16)   utm_letter = 'L';
else if (-16 > parseFloat(latitude) && parseFloat(latitude) >= -24)   utm_letter = 'K';
else if (-24 > parseFloat(latitude) && parseFloat(latitude) >= -32)   utm_letter = 'J';
else if (-32 > parseFloat(latitude) && parseFloat(latitude) >= -40)   utm_letter = 'H';
else if (-40 > parseFloat(latitude) && parseFloat(latitude) >= -48)   utm_letter = 'G';
else if (-48 > parseFloat(latitude) && parseFloat(latitude) >= -56)   utm_letter = 'F';
else if (-56 > parseFloat(latitude) && parseFloat(latitude) >= -64)   utm_letter = 'E';
else if (-64 > parseFloat(latitude) && parseFloat(latitude) >= -72)   utm_letter = 'D';
else if (-72 > parseFloat(latitude) && parseFloat(latitude) >= -80)   utm_letter = 'C';
else      utm_letter = "\0";
utm_easting += (zone - 30)*500000.0;
return new Array(utm_northing, utm_easting, zone, utm_letter); // Creem una array amb el resultat de la conversió de les dades
}

```

