

Infraestructuras de Datos Espaciales (SIG en la Web)

Tema 3.2. Clientes Web IDE

Programa

Modulo 1. Introducción

- Tema 1.1. Introducción a las IDE (SIG en la web)
- Tema 1.2. Construcción de los Modelos espaciales
- Tema 1.3. Medida de la componente temática de los datos espaciales

Modulo 2. Representación de la Información Geográfica

- Tema 2.1. Construcción de los Modelos de representación
- Tema 2.2. Formatos de estructuras de datos espaciales
- Tema 2.3. Explotación de un SIG
- Tema 2.4. Captura de datos espaciales
- **Modulo 3. Publicación de la Información espacial**
- Tema 3.1. Definiciones, componentes y Servicios Web de una IDE
- **Tema 3.2. Clientes IDE**
- **Modulo 4. Construcción de un servicio web**

Índice

1. Visores de la I.E.

2. Clientes Ligeros

3. Ejemplos de clientes ligeros:

- Open Layers
- Leaflet

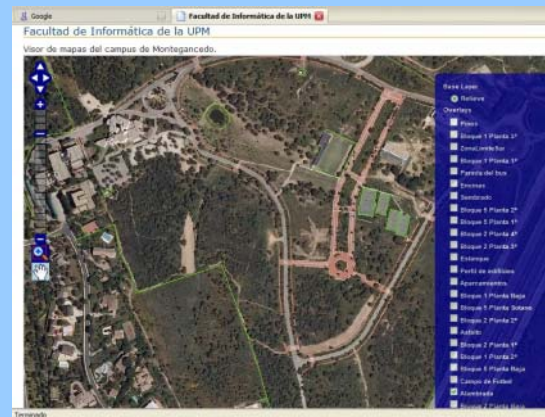
4. Clientes pesados

1. Visores de la I.E.

Practica 3.1

Un usuario puede acceder mediante un navegador web a una página que ofrece un servicio:

**http://servidor_wms?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1
&LAYERS=Campo_Futbol&STYLES=&SRS=EPSG:23030&BBOX=539840.23
797,4695887.53302,552440.23797,4706450.03302&WIDTH=640&HEIGHT=4
80&FORMAT=image/png**



Inconveniente: Hay que escribir todos estos parámetros para solicitar una sola capa

Solución:

- Clientes ligeros
- Clientes pesados.

Visores de la I. Espacial

Cientes ligeros y pesados: dos formas complementarias y distintas de acceder y visualizar servicios IDE

Están basadas en la arquitectura cliente-servidor

Se distinguen por el énfasis diferente que hacen sobre las dos partes que intervienen

Visores de la I. Espacial

Cientes ligeros

- Son visores web embebidos (incluyen en las paginas web los códigos de llamada a los servicios)
- No tienen capacidad de procesamiento de datos
- Es posible su utilización y acceso mediante un navegador web
- Forma muy simple y directa, pero a la vez limitada

Cientes pesados,

- Es necesaria una instalación en local
- Requieran cierto grado de análisis o procesamiento de los datos.
- No requiere de desarrollos, son los programas ya instalados (ArcGIS, gvSIG, Kosmo...) los que disponen de las capacidades para realizar las llamadas a los servicios y poseen capacidades de interacción estándares

Índice

1. Visores de la I.E.

2. Clientes Ligeros

3. Ejemplos de clientes ligeros:

- Open Layers
- Leaflet

4. Clientes pesados

Cientes ligeros de visualización (*thin client*)

- **Actúan como intermediario entre cliente y servidor**
- **No se utilizan las capacidades de proceso del cliente**
- **Los requisitos respecto a recursos hardware en el cliente son mínimos**
- **Aumenta la complejidad del servidor**
- **Gran control sobre los permisos en datos y operaciones**
- **Todas las demandas y respuestas van por la red**

Cientes ligeros para la visualización

Funcionan gracias a la interacción de los siguientes elementos:

- **Un navegador web en la parte cliente**
- **Una API en la parte servidora**
- **Un documento HTML contenedor de la visualización, a modo de página web**
- **Una programación JavaScript**

Clientes ligeros

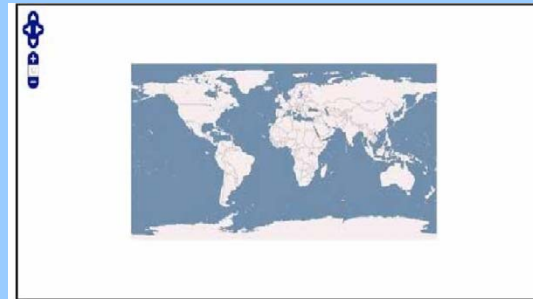
- Permiten consultar datos, servicios web, usar funcionalidades implementadas ,
- La palabra ligero se refiere a **lo pequeña que es la imagen de arranque**, no más grande que la requerida para conectar a la red y arrancar un navegador Web.

Ejemplos:

- Openlayers,
- Leaflet,
- Cartociudad,
- MapBender,
- MapFish,
- ArcGIS Viewer for Flex....

Cientes ligeros

Open Layers



```

1 <html>
2
3 <head>
4
5 <title>Prueba cliente libro con OpenLayers</title>
6
7 <meta name="description" content="CURSO DE OPENLAYERS HTML" />
8
9 <meta name="viewport" content="width=device-width, height=device-height" />
10
11 <meta name="author" content="GABRIEL DE MIRANDA DEL VILLALBA" />
12
13 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
14
15 <script src="http://openlayers.org/api/OpenLayers.js" type="text/javascript"></script>
16
17 <link href="style.css" rel="stylesheet" type="text/css" />
18
19 </head>
20
21 <body>
22
23 <div id="map">
24
25 <div id="EJEMPLO DE USO DE LA API DE OPENLAYERS" />
26
27 <div id="map">
28
29 <div id="map">
30
31 <div id="map">
32
33 <div id="map">
34
35 <div id="map">
36
37 <div id="map">
38
39 <div id="map">
40
41 <div id="map">
42
43 <div id="map">
44
45 <div id="map">
46
47 <div id="map">
48
49 <div id="map">
50
51 <div id="map">
52
53 <div id="map">
54
55 <div id="map">
56
57 <div id="map">
58
59 <div id="map">
60
61 <div id="map">
62
63 <div id="map">
64
65 <div id="map">
66
67 <div id="map">
68
69 <div id="map">
70
71 <div id="map">
72
73 <div id="map">
74
75 <div id="map">
76
77 <div id="map">
78
79 <div id="map">
80
81 <div id="map">
82
83 <div id="map">
84
85 <div id="map">
86
87 <div id="map">
88
89 <div id="map">
90
91 <div id="map">
92
93 <div id="map">
94
95 <div id="map">
96
97 <div id="map">
98
99 <div id="map">
100
101 <div id="map">
102
103 <div id="map">
104
105 <div id="map">
106
107 <div id="map">
108
109 <div id="map">
110
111 <div id="map">
112
113 <div id="map">
114
115 <div id="map">
116
117 <div id="map">
118
119 <div id="map">
120
121 <div id="map">
122
123 <div id="map">
124
125 <div id="map">
126
127 <div id="map">
128
129 <div id="map">
130
131 <div id="map">
132
133 <div id="map">
134
135 <div id="map">
136
137 <div id="map">
138
139 <div id="map">
140
141 <div id="map">
142
143 <div id="map">
144
145 <div id="map">
146
147 <div id="map">
148
149 <div id="map">
150
151 <div id="map">
152
153 <div id="map">
154
155 <div id="map">
156
157 <div id="map">
158
159 <div id="map">
160
161 <div id="map">
162
163 <div id="map">
164
165 <div id="map">
166
167 <div id="map">
168
169 <div id="map">
170
171 <div id="map">
172
173 <div id="map">
174
175 <div id="map">
176
177 <div id="map">
178
179 <div id="map">
180
181 <div id="map">
182
183 <div id="map">
184
185 <div id="map">
186
187 <div id="map">
188
189 <div id="map">
190
191 <div id="map">
192
193 <div id="map">
194
195 <div id="map">
196
197 <div id="map">
198
199 <div id="map">
200
201 <div id="map">
202
203 <div id="map">
204
205 <div id="map">
206
207 <div id="map">
208
209 <div id="map">
210
211 <div id="map">
212
213 <div id="map">
214
215 <div id="map">
216
217 <div id="map">
218
219 <div id="map">
220
221 <div id="map">
222
223 <div id="map">
224
225 <div id="map">
226
227 <div id="map">
228
229 <div id="map">
230
231 <div id="map">
232
233 <div id="map">
234
235 <div id="map">
236
237 <div id="map">
238
239 <div id="map">
240
241 <div id="map">
242
243 <div id="map">
244
245 <div id="map">
246
247 <div id="map">
248
249 <div id="map">
250
251 <div id="map">
252
253 <div id="map">
254
255 <div id="map">
256
257 <div id="map">
258
259 <div id="map">
260
261 <div id="map">
262
263 <div id="map">
264
265 <div id="map">
266
267 <div id="map">
268
269 <div id="map">
270
271 <div id="map">
272
273 <div id="map">
274
275 <div id="map">
276
277 <div id="map">
278
279 <div id="map">
280
281 <div id="map">
282
283 <div id="map">
284
285 <div id="map">
286
287 <div id="map">
288
289 <div id="map">
290
291 <div id="map">
292
293 <div id="map">
294
295 <div id="map">
296
297 <div id="map">
298
299 <div id="map">
300
301 <div id="map">
302
303 <div id="map">
304
305 <div id="map">
306
307 <div id="map">
308
309 <div id="map">
310
311 <div id="map">
312
313 <div id="map">
314
315 <div id="map">
316
317 <div id="map">
318
319 <div id="map">
320
321 <div id="map">
322
323 <div id="map">
324
325 <div id="map">
326
327 <div id="map">
328
329 <div id="map">
330
331 <div id="map">
332
333 <div id="map">
334
335 <div id="map">
336
337 <div id="map">
338
339 <div id="map">
340
341 <div id="map">
342
343 <div id="map">
344
345 <div id="map">
346
347 <div id="map">
348
349 <div id="map">
350
351 <div id="map">
352
353 <div id="map">
354
355 <div id="map">
356
357 <div id="map">
358
359 <div id="map">
360
361 <div id="map">
362
363 <div id="map">
364
365 <div id="map">
366
367 <div id="map">
368
369 <div id="map">
370
371 <div id="map">
372
373 <div id="map">
374
375 <div id="map">
376
377 <div id="map">
378
379 <div id="map">
380
381 <div id="map">
382
383 <div id="map">
384
385 <div id="map">
386
387 <div id="map">
388
389 <div id="map">
390
391 <div id="map">
392
393 <div id="map">
394
395 <div id="map">
396
397 <div id="map">
398
399 <div id="map">
400
401 <div id="map">
402
403 <div id="map">
404
405 <div id="map">
406
407 <div id="map">
408
409 <div id="map">
410
411 <div id="map">
412
413 <div id="map">
414
415 <div id="map">
416
417 <div id="map">
418
419 <div id="map">
420
421 <div id="map">
422
423 <div id="map">
424
425 <div id="map">
426
427 <div id="map">
428
429 <div id="map">
430
431 <div id="map">
432
433 <div id="map">
434
435 <div id="map">
436
437 <div id="map">
438
439 <div id="map">
440
441 <div id="map">
442
443 <div id="map">
444
445 <div id="map">
446
447 <div id="map">
448
449 <div id="map">
450
451 <div id="map">
452
453 <div id="map">
454
455 <div id="map">
456
457 <
```

Leaflet



```
<html>
<head>
<link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.6.4/leaflet.css"/>
<link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.6.4/leaflet.ie.css"/>
<script src="http://cdn.leafletjs.com/leaflet-0.6.4/leaflet.js"></script>
</head>
<body>
<div style="width:100%; height:100%; id="map"></div>
<script>
var map = L.map('map',{crs:L.CRS.EPSG4326}).setView([0,0],3);
var cupul1 = L.tileLayer.wms('http://vmap.0.tiles.osgeo.org/wms/vmap0',
{layers:'basic',format:'image/jpeg',transparent:true,attribution:'OsGEO'});
cupul1.addTo(map);
</script>
</body>
</html>
```

Cartociudad



```

1 <html>
2   <head>
3     <title>Pruebas cliente ligero de un servicio IZM</title>
4     <meta name="description" content="PRUEBAS DEMONSTRACION IZM"/>
5     <meta name="keyword" content="IZM, GEOGRAFICA, CLIENTE LIGERO"/>
6     <meta name="author" content="GRUPO DE TRABAJO IZM-HISTORIASDEISRAEL"/>
7     <a href="http://www.izm.es/">http://www.izm.es/</a>
8     <script src="http://www.cartovisor.es/js/VisualizadorCartografico/Cartovisor.js" type="text/javascript"></script>
9   </head>
10
11   <body>
12     </body>
13   </html>

```

Índice

1. Visores de la I.E.

2. Clientes Ligeros

3. Ejemplos de clientes ligeros:

- Open Layers
- Leaflet

4. Clientes pesados

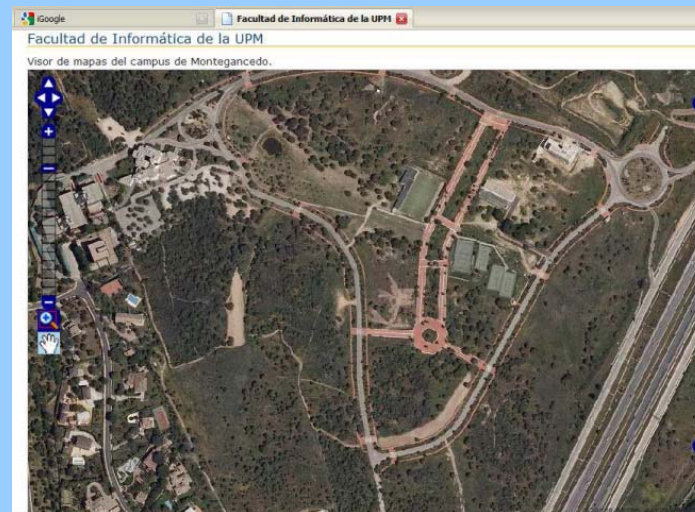
5. Otros clientes

3.1. OpenLayers

Arquitectura

Se compone de tres partes:

1. La propia librería **OpenLayers** que tiene una serie de primitivas para permitir la visualización de las capas de información.
2. La información geográfica que será publicada en el cliente
3. Código JavaScript incrustado en un archivo **HTML**



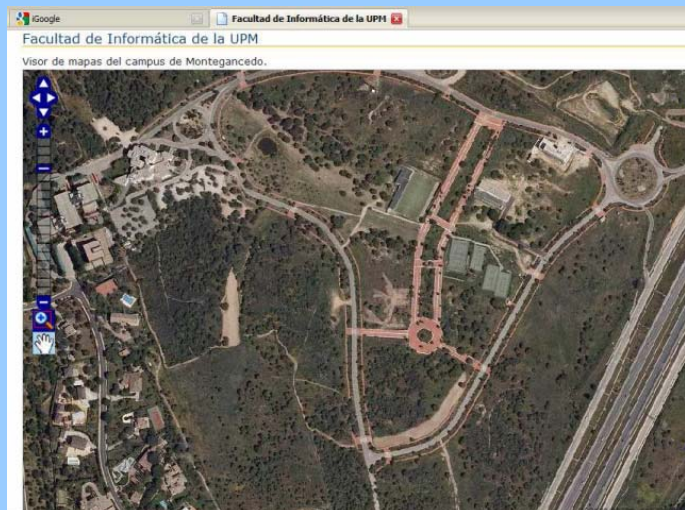
3.1. Open Layers

Pantalla del cliente Web

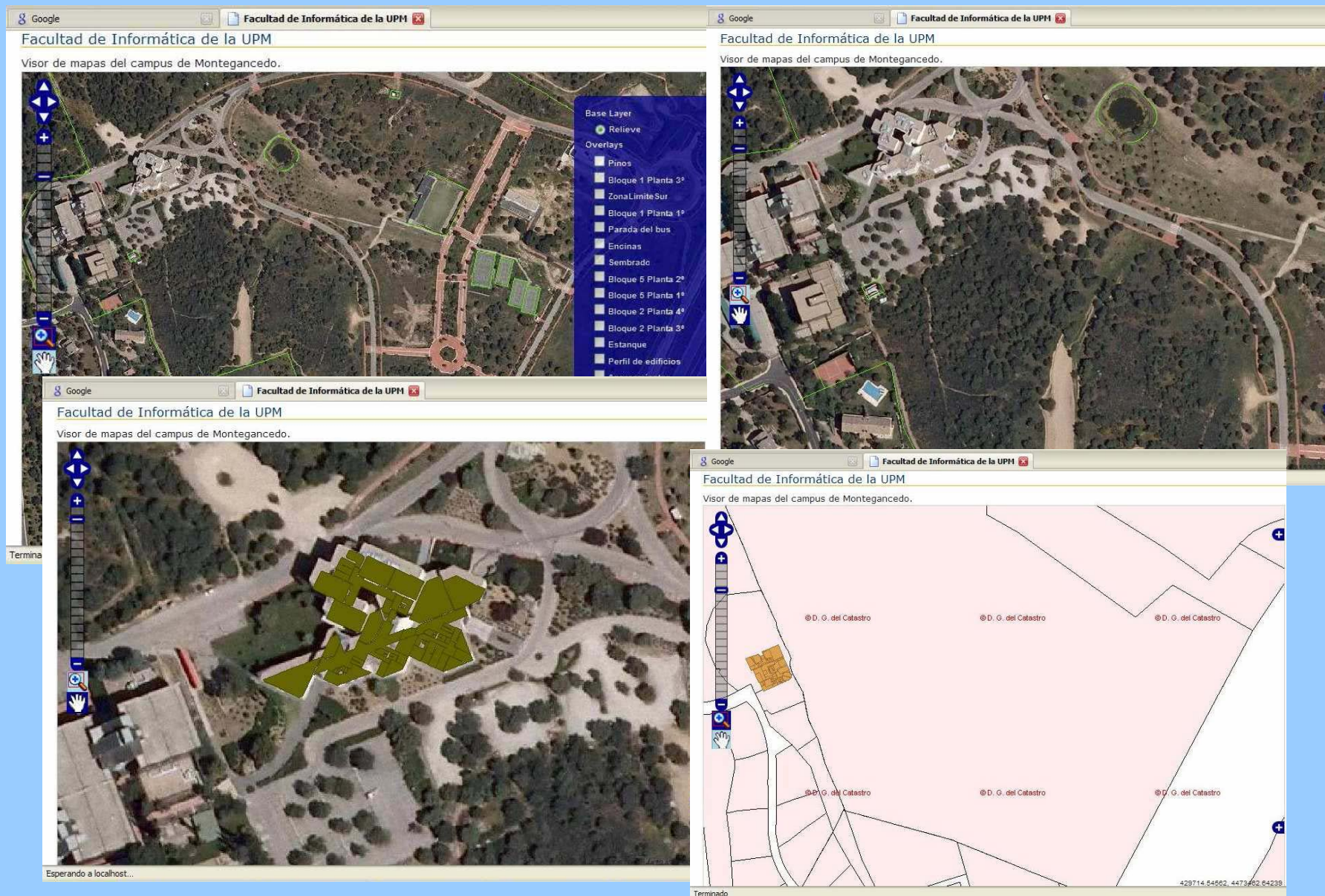
- **Un navegador de cuatro botones.** Es posible navegar por el mapa seleccionando el icono de la mano y “arrastrando” el mapa
- **Una barra de zoom.** OpenLayers permite como máximo 18 niveles de zoom, y al iniciar el cliente, éste está fijado al nivel 14
- **Las coordenadas del sistema de referencia en la parte inferior derecha.** Donde está situado el cursor en el mapa
- **Dos pestañas en el lateral derecho.** Al desplegarse visualizan el menú de selección de capas y el mapa pequeño de referencia.

Ejemplo:

Una sola capa base que es la foto de satélite del PNOA
Diferentes capas overlay, que se pueden superponer



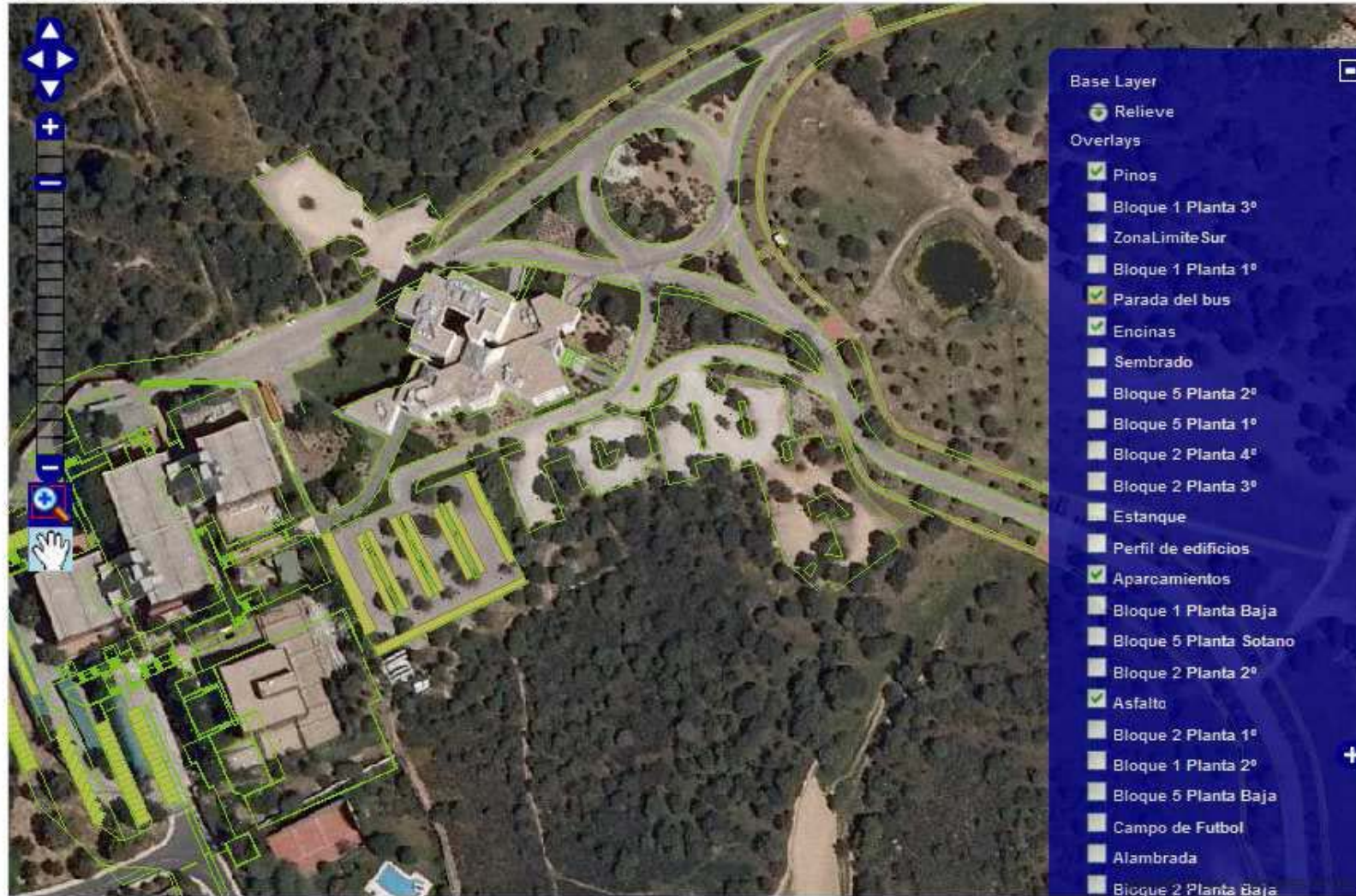
Visualización de capas Open Layers.



Superposición de capas OpenLayers

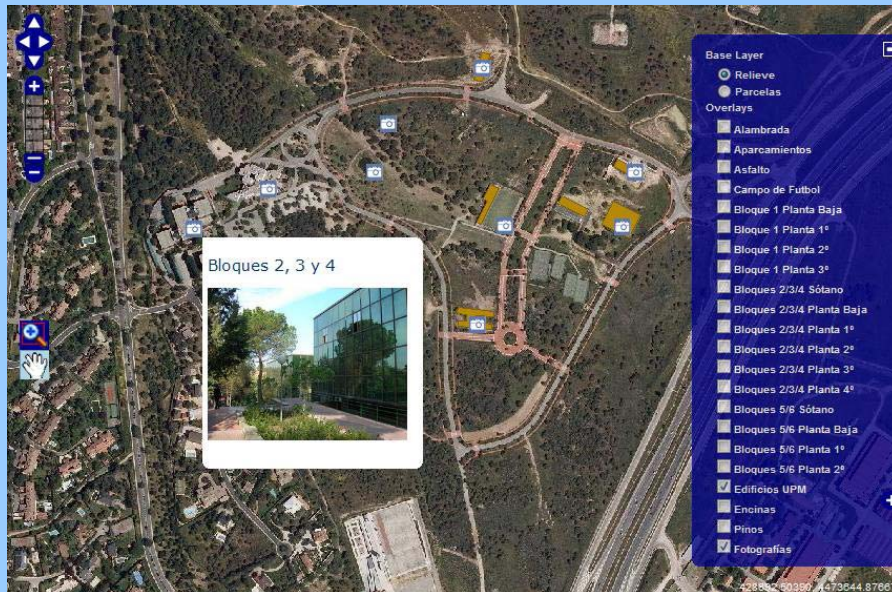
Facultad de Informática de la UPM

Visor de mapas del campus de Montegancedo.




Terminado

Visualización



Más información

<http://www.openlayers.org/>

 OpenLayers

HomeSupport & Development

SponsorshipDocumentationDownloadFAQEmail ListsBlog


OpenLayers: Free Maps for the Web

Get OpenLayers Now!

- 2.13.1 (Stable): [.tar.gz](#) | [zip](#)
- [2.13.1 Release Notes](#)
- [API Documentation, User documentation](#)
- See examples of OpenLayers Usage: [Release Examples \(2.13.1\)](#), [Development Examples](#)
- Fork us on [GitHub](#)

About OpenLayers

OpenLayers makes it easy to put a dynamic map in any web page. It can display map tiles and markers loaded from any source. OpenLayers has been developed to further the use of geographic information of all kinds. OpenLayers is completely free, Open Source JavaScript, released under the [2-clause BSD License](#) (also known as the [FreeBSD](#)).



Put an open map widget in any web page!

Toward OpenLayers 3!

We've begun the development effort to make the next major version of OpenLayers a reality. [OpenLayers 3](#) is a comprehensive rewrite of the library, targeting the latest in HTML5 and CSS3 features. The library will continue to have broad support for projections, standard protocols, and editing functionality from OpenLayers 2.x. The new version of the library will focus on performance improvements, lighter builds, prettier visual components, an improved API, and more. Some of the major highlights are:

- **WebGL**: promises to bring 3D capabilities and increased performance for all mapping needs to the latest browsers. OpenLayers 3.0 will offer WebGL, while degrading nicely in less capable browsers.
- **Cesium**: The OpenLayers community will also integrate the new Cesium library to enable full 3D spinning globe capabilities directly into the 3.0 release.
- **Closure Compiler**: By utilizing the Closure Compiler, applications developers will be able to create smaller and faster libraries, easing the use of the extensive OpenLayers 3.0 toolkit.
- **A new codebase**: This offers an opportunity to clean up some of the "clunky" ways of doing things in OpenLayers. The team will also create with new API designs, which will be more accessible to all.
- **High-quality documentation**: The new release will also feature documentation with fresh examples and default designs in OpenLayers 3.0. Making a toolkit standout is about more than the actual code.

If you've used OpenLayers in the past, if you're using it in production deployments, or if you may use it in the future, please consider contributing to our collaborative funding effort. We are aiming to raise \$350,000, to fully fund OpenLayers 3.0 in one shot. Contributing right now is the most effective time to help OpenLayers. Kickstarter-style campaign is running at <http://www.indiegogo.com/o13>, read our [blog post](#) to know more about it. Any amount will help but if you are able to give \$5,000 or more we're offering a number of different 'perks' to help make it easier to contribute:


- \$5,000: Your organization's logo will appear on the OpenLayers 3.0 Thank You page.
- \$25,000: A 1 year OpenLayers support package with two named users.
- \$40,000: Logo and link will appear on the OpenLayers homepage as a "Sustaining Sponsor", as well as at the top of the OpenLayers 3.0 Thank You page.
- \$75,000: A 1 year OpenLayers enterprise support package, with up to 8 named users, and "Sustaining Sponsor" logos and credits.

These packages will be delivered by companies employing OpenLayers committers. For more information please email [o13-funding \[at\] googlegroups.com](mailto:o13-funding@googlegroups.com).


For Developers!

OpenLayers is a pure JavaScript library for displaying map data in most modern web browsers, with no server-side dependencies. OpenLayers implements a JavaScript API for building rich web-based geographic applications, similar to the Google Maps APIs, with one important difference -- OpenLayers is Free Software, developed for and


Books about OpenLayers




OpenLayers Cookbook
Antonio Santiago Perez
Available in Print and eBook formats from Packt Publishing.



OpenLayers 2.10 Beginner's Guide
Erik Hazzard
Available in Print and eBook formats from Packt Publishing.



OpenLayers (in German)
Till Adams & Marc Jansen
Available from Open Source Press.



OSGeo
Your Open Source Compass

OpenLayers is a project of the Open Source Geospatial Foundation.

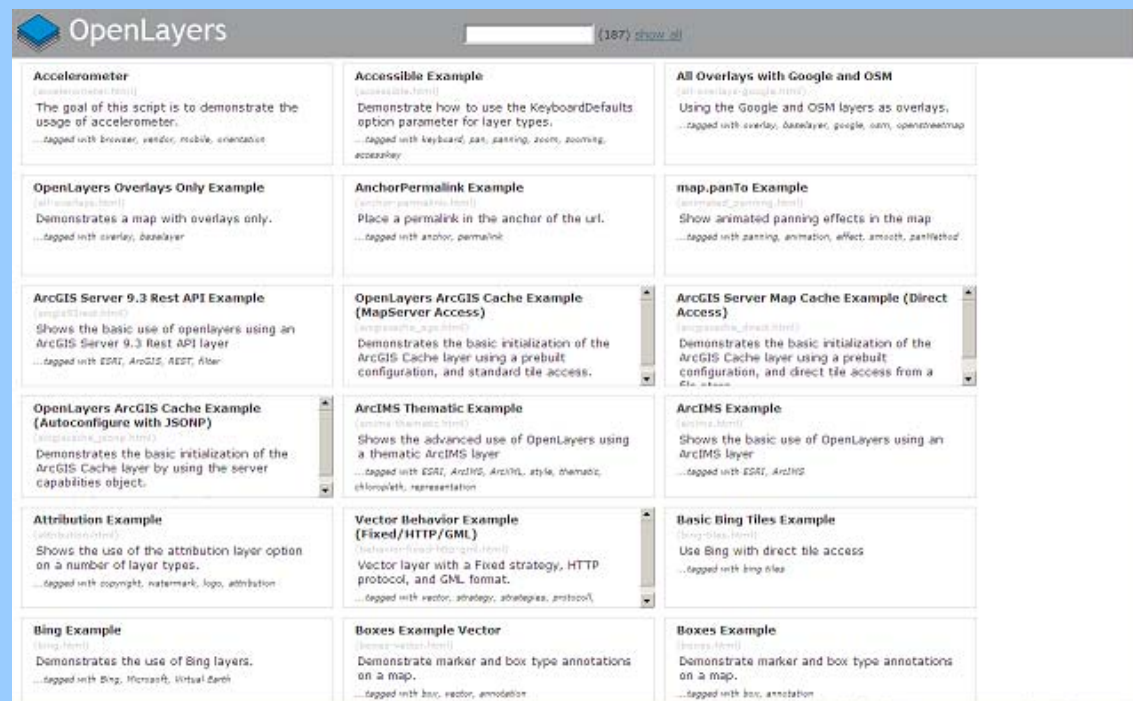
- Más información

- Documentación:

- <http://docs.openlayers.org/>

- Ejemplos:

- <http://openlayers.org/dev/examples/>



3.2. Leaflet

Practica M.3.2.

Librería JavaScript opensource para crear mapas interactivos y para la visualización de los datos de un mapa en la mayoría de los navegadores web

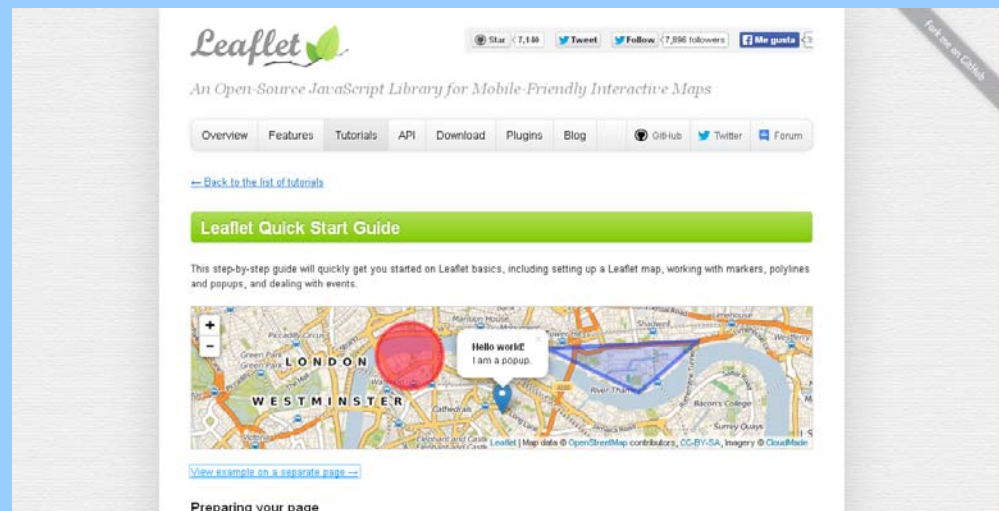
Leaflet es utilizada por: Wikimedia, OSM, ...

- **Implementa un API para el desarrollo de aplicaciones geográficas en la red similar a la de Google Maps**
- **Permite integrar herramientas sencillas de visualización**
- **Ofrece una gran cantidad de posibles configuraciones**
- **Mediante Leaflet es posible visualizar en una página web cualquier combinación de mapas ofrecida por cualquier servidor WMS**

3.2. Leaflet

Ventajas:

- Sencillo y rápido de aprender
- Facilidad de uso
- Características básicas pero que funcionan a la perfección
- Soporte móvil
- HTML 5 y CSS3
- Ampliable con plugins
- API bien documentada



3.2. Leaflet

Arquitectura

Una aplicación Leaflet se compone de tres partes:

1. La propia librería JavaScript `leaflet.js` que tiene una serie de primitivas para permitir la visualización de las capas de información.
2. La información geográfica que será publicada en el cliente
3. Código JavaScript incrustado en un archivo HTML



Conceptos básicos

- **Mapa [L.map]**
 - Nucleo de Leaflet
 - Se muestra en el componente de HTML cuyo id se le indica al crear el mapa
 - Es necesario fijar la posición inicial [setView()]
- **Capa [L.TileLayer()]**
 - Representa el origen remoto de datos
 - Formado por teselas
 - Al crearlo se indica la URL de origen de datos
 - Hay que añadirla [addTo(map)]

3.2 Leaflet

Ejemplo: Practica 3.2.1

```
<html>
<head>
  <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.6.4/leaflet.css"/>
  <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.6.4/leaflet.ie.css" />
  <script src="http://cdn.leafletjs.com/leaflet-0.6.4/leaflet.js"></script>
</head>
<body>
  <div style="width:100%; height:100%" id="map"></div>
  <script>
    var map = L.map('map',{crs: L.CRS.EPSG4326}).setView([0,0],3);
    var capa1 = L.tileLayer.wms("http://vmap0.tiles.osgeo.org/wms/vmap0",
      {layers: 'basic', format: 'image/jpeg', transparent: true, attribution: "OsGEO"});
    capa1.addTo(map);
  </script>
</body>
</html>
```



3.2 Leaflet

Funcionalidades

1. Añadir capas
2. Añadir funcionalidades de visualización

3.2. Leaflet

1. Capas (Practica M.3.2. Parte 1)

Dos tipos de capas de información:

- **BaseLayers** (capas base) son excluyentes entre sí
- **Overlays** (capas superpuestas) se pueden superponer entre ellas para ser visualizadas conjuntamente

Soporta diferentes fuentes de datos: WMS, Yahoo!Maps, WorldWind (NASA World Wind).. ...



3.2. Leaflet

1. Añadir capas (Practica M.3.2. Parte 1)

Se puede añadir una capa procedente de un WMS mediante la primitiva:
“L.tileLayer.wms”

1. Si es una capa Overlay :

```
var nombre_capa = L.tileLayer.wms(“nombre que tomará la capa en el  
cliente”, “dirección de la capa”, {layers: ‘nombre de la capa en el servidor’,  
format: “formato de la imagen “, transparent: true, attribution:”nombre”});
```

2. Si es una nueva capa base:

```
var nombre_capa = L.tileLayer.wms (“nombre que tomará la capa en el  
cliente”, “dirección de la capa”, {layers: ‘nombre de la capa en el servidor’});
```

Se añaden las capas a la pagina:

```
L.control.layers(nombre_capa_1,nombre_capa_2,...,nombre_capa_n).addTo(  
map);
```

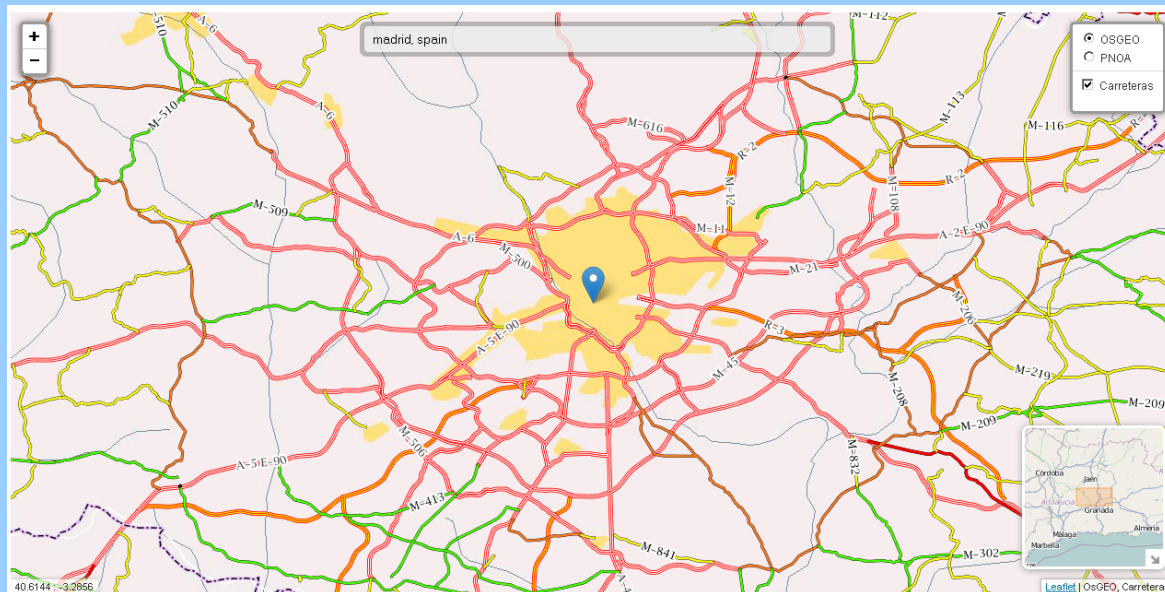
2. Añadir Funcionalidades de visualización (Practica M.3.2. Parte 2)

Caja de localización/Buscador/Control Raton

Se necesita indicar en la página la **localización del script**, que puede ser consultado:

- **De forma local.** No requiere ningún tipo de instalación, tan solo hay que copiar el script con todas sus carpetas y librerías en el servidor.
- **De forma remota.** En el cliente web se especifica la dirección de las diferentes capas de información

```
//Minimap
var osm2 = new L.TileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {minZoom: 0,
maxZoom: 13});
var miniMap = new L.Control.MiniMap(osm2,{ toggleDisplay: true }).addTo(map);
//MousePosition
var mousePosition = new L.Control.MousePosition().addTo(map);
//Búsqueda
new L.Control.GeoSearch({provider: new L.GeoSearch.Provider.Google({region: ''})}).addTo(map);
```



Otros clientes ligeros

EzWeb

- Plataforma Web open source estándar y abierta que permite al usuario/a **construir su propio entorno de trabajo** mediante la selección, configuración, combinación e interconexión de componentes simples denominados gadgets.



MyMobileWeb

- Plataforma open source que permite realizar el desarrollo de sitios Web accesibles desde dispositivos móviles.
- Facilita la presentación de contenidos de manera independiente al dispositivo separando lógica de presentación.
- Capaz de identificar el dispositivo con el que se accede, asociarle la presentación que le corresponde y gestionar la transición entre las presentaciones.

<http://mymobileweb.morfeo-project.org>



Índice

1. Visores de la I.E.

2. Clientes Ligeros

3. Ejemplos de clientes ligeros:

- Open Layers
- Leaflet

4. Clientes pesados

Principales características de un cliente pesado (*thick client*)

- El cliente implementa la mayor parte de la lógica de aplicación
- El cliente realiza procesamiento sobre los datos de usuario antes de comunicar con servidor y realizar demandas
- Puede ser una aplicación (un SIG) o un navegador que pueda recibir código desde el servidor (*applets, plug-ins*) para incrementar sus capacidades de proceso de información espacial
- El servidor puede ser relativamente sencillo, las responsabilidades son mínimas (p.e. gestión datos)
- Más control por parte de los usuarios
- Permite que ciertas operaciones se realicen en tiempo real

Cientes pesados

- Es necesario instalarlos en el ordenador
- Es necesario disponer de Internet
- Ofrecen diferentes funcionalidades más complejas que las que ofrece un cliente ligero.
- Realiza la mayor parte del procesamiento de datos.
- Los datos son almacenados en el servidor.

Ejemplos:

- Aplicación Java o C++, que se instala y ejecuta,
- GoogleEarth
- gvSIG
- ArcGIS

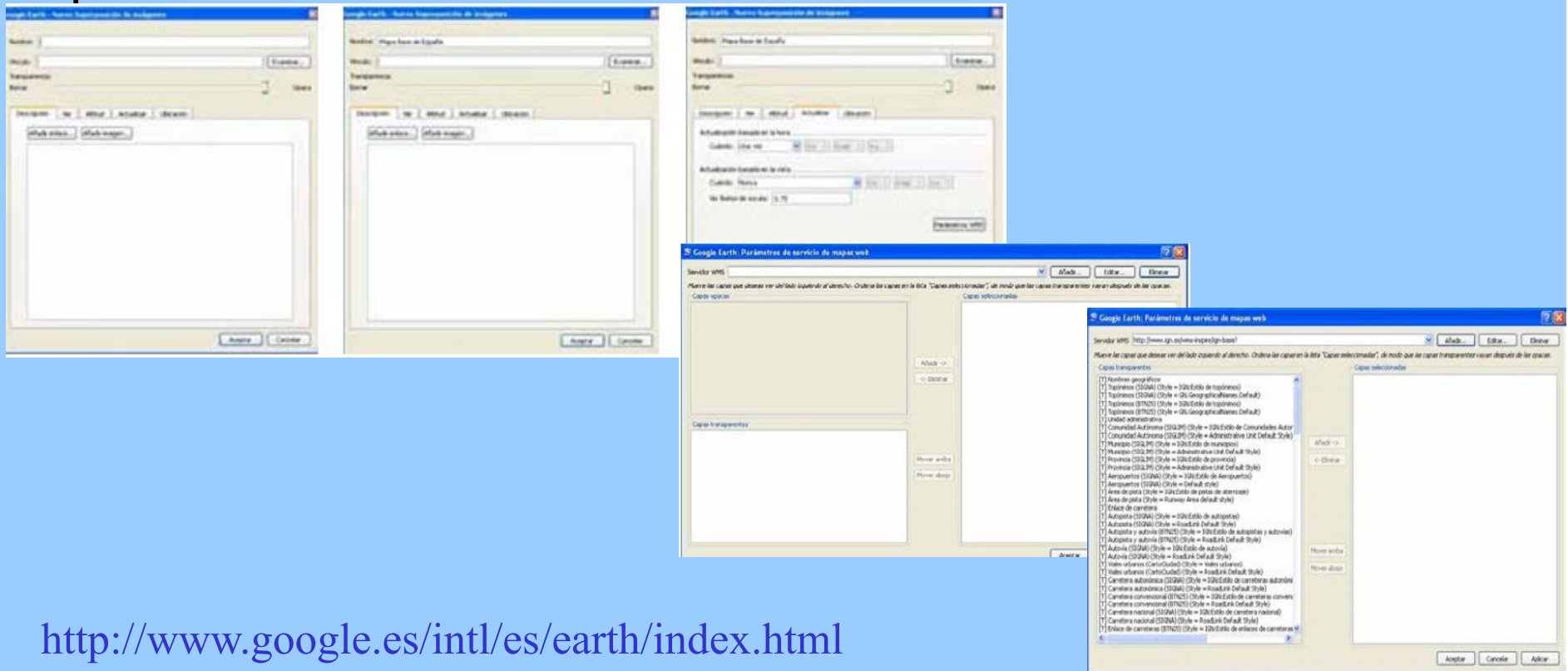
<http://www.idee.es/web/guest/herramientas-gratuitas>

Google Earth

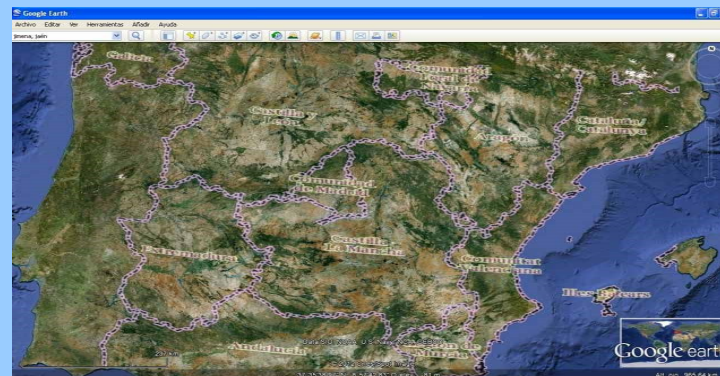
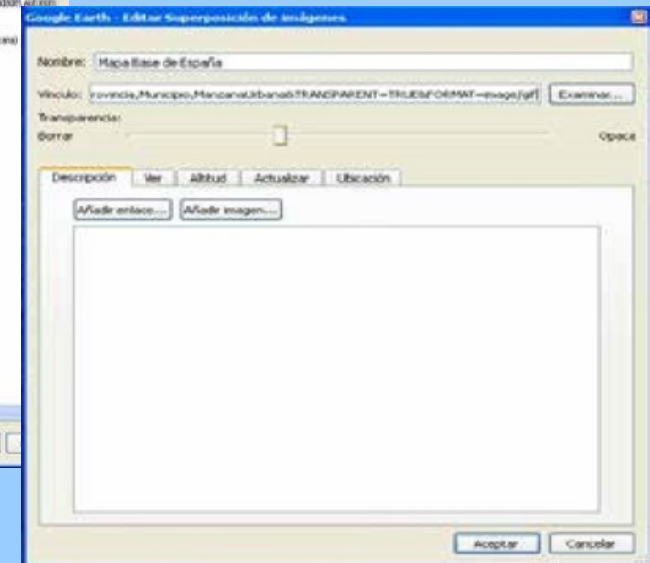
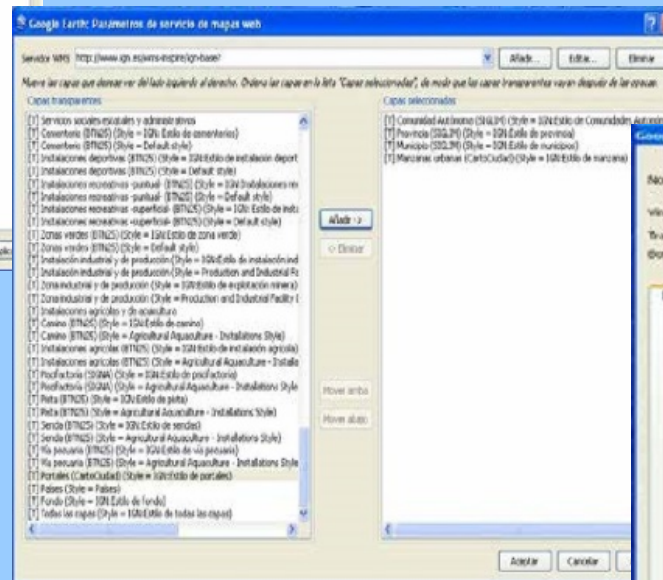
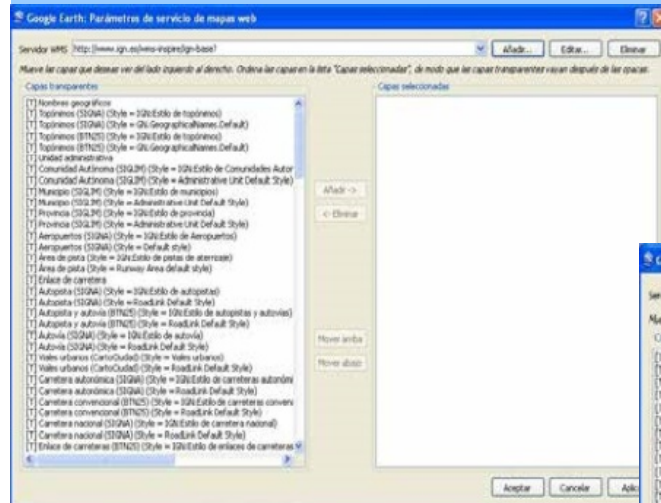
Permite visualizar múltiple cartografía, basada en información de satélite

Uno de los programas más populares

Permite cargar servicios WMS y guardar como .kml un wms cargado previamente



Google Earth



Capacidad de cliente pesado IDE, puede conectarse de manera sencilla y efectiva:

- A servidores WMS, WFS y WCS,
- A servidores no estándar como ArcIMS.



Página oficial

www.gvsig.gva.es (<http://www.gvsig.gva.es/>).

gvSIG

