

# SIG. Tema 6: Presentación (Servidores de Mapas)

José Samos Jiménez

2020 jsamos (lsi-ugr)  
Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Granada

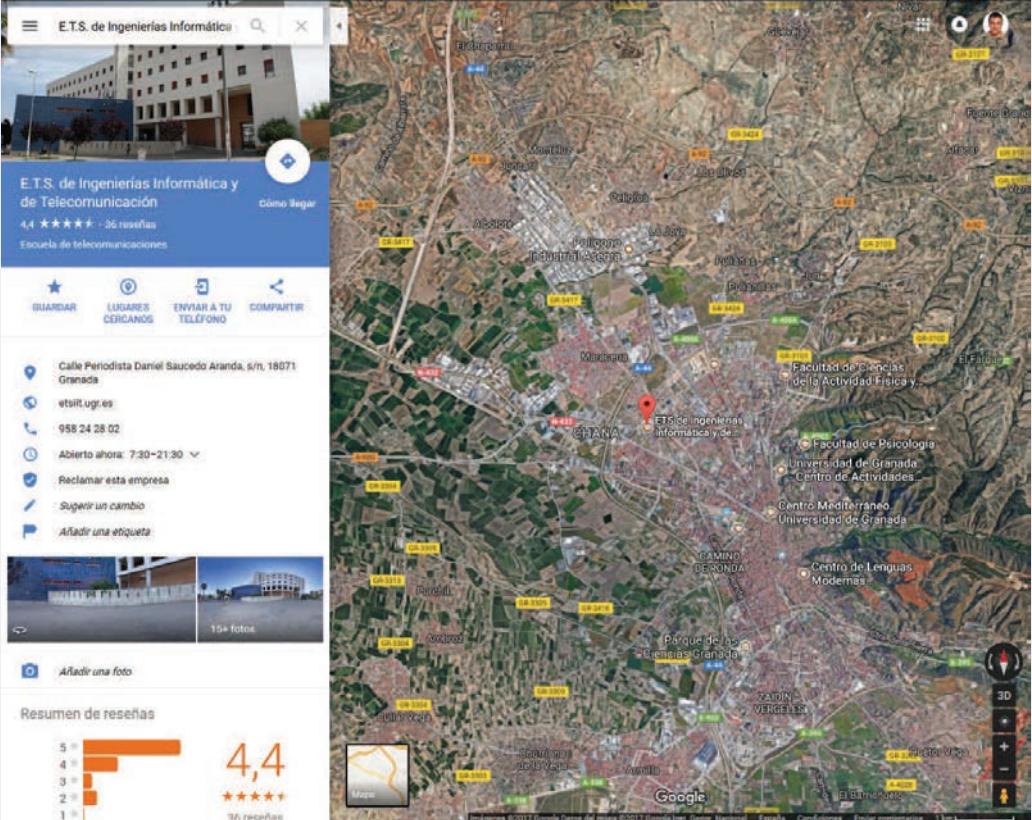
Curso 2020-21



## Resultado tradicional



# Resultado actual



The screenshot shows a Google Maps search result for 'E.T.S. de Ingenierías Informática y de Telecomunicación'. The map is centered on the university campus in Granada, Spain. A red marker indicates the exact location of the institution. The sidebar on the left provides detailed information about the place, including its name, address (Calle Periodista Daniel Saucedo Aranda, s/n, 18071 Granada), phone number (958 24 28 02), and opening hours (7:30-21:30). It also displays a 4.4 rating from 36 reviews and a photo gallery with 15+ photos. The map shows various landmarks and streets around the campus.

## Contenido

- 1 Historia de los GIS
- 2 Tecnología web convencional
- 3 Servidores de cartografía
- 4 Clientes de cartografía
- 5 Geospatial mashups
- 6 Bibliografía

# Historia de los GIS

1 Historia de los GIS

2 Tecnología web convencional

3 Servidores de cartografía

4 Clientes de cartografía

5 Geospatial mashups

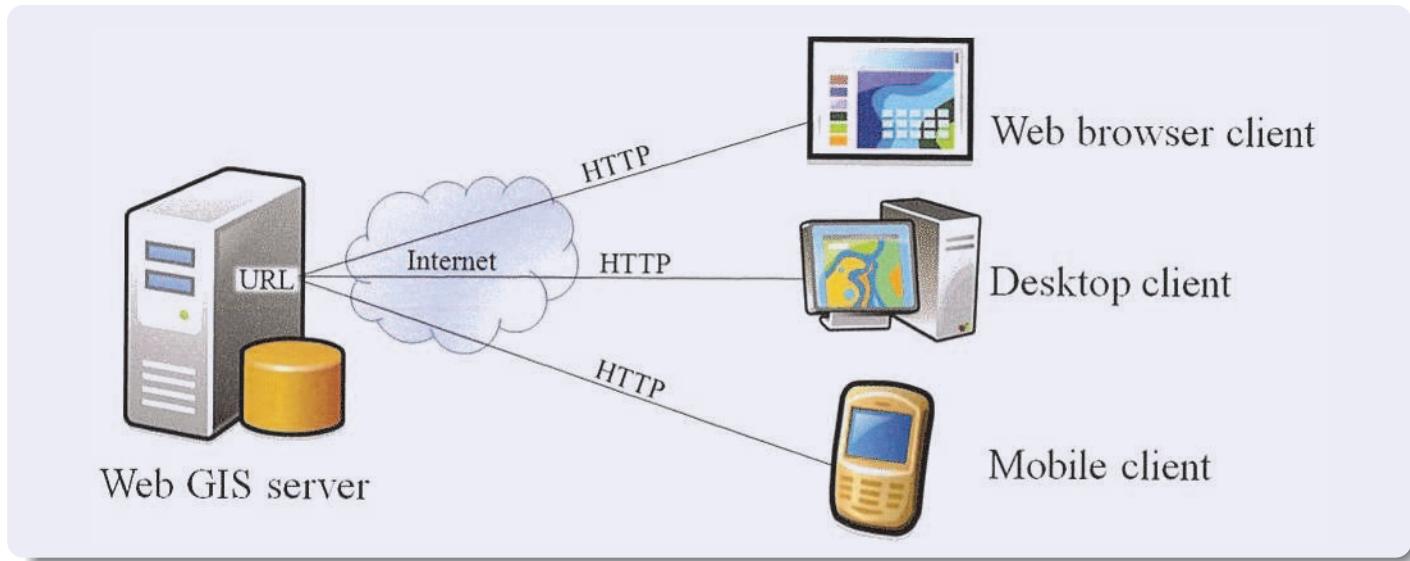
6 Bibliografía

## Hitos en la historia



- Año 1993: Primer servidor de mapas, Xerox PARC Viewer.
- Año 1994: Primer atlas digital en línea: Atlas Nacional de Canadá.
- Años 2000: Web Mapping.

# Web Mapping

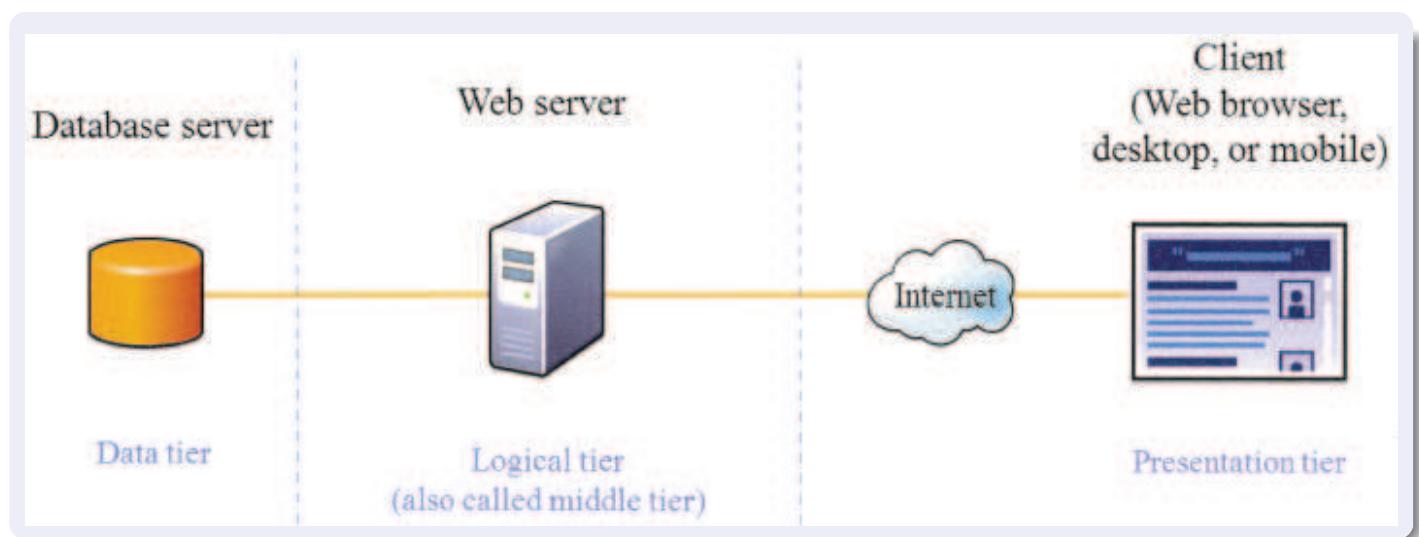


# Tecnología web convencional

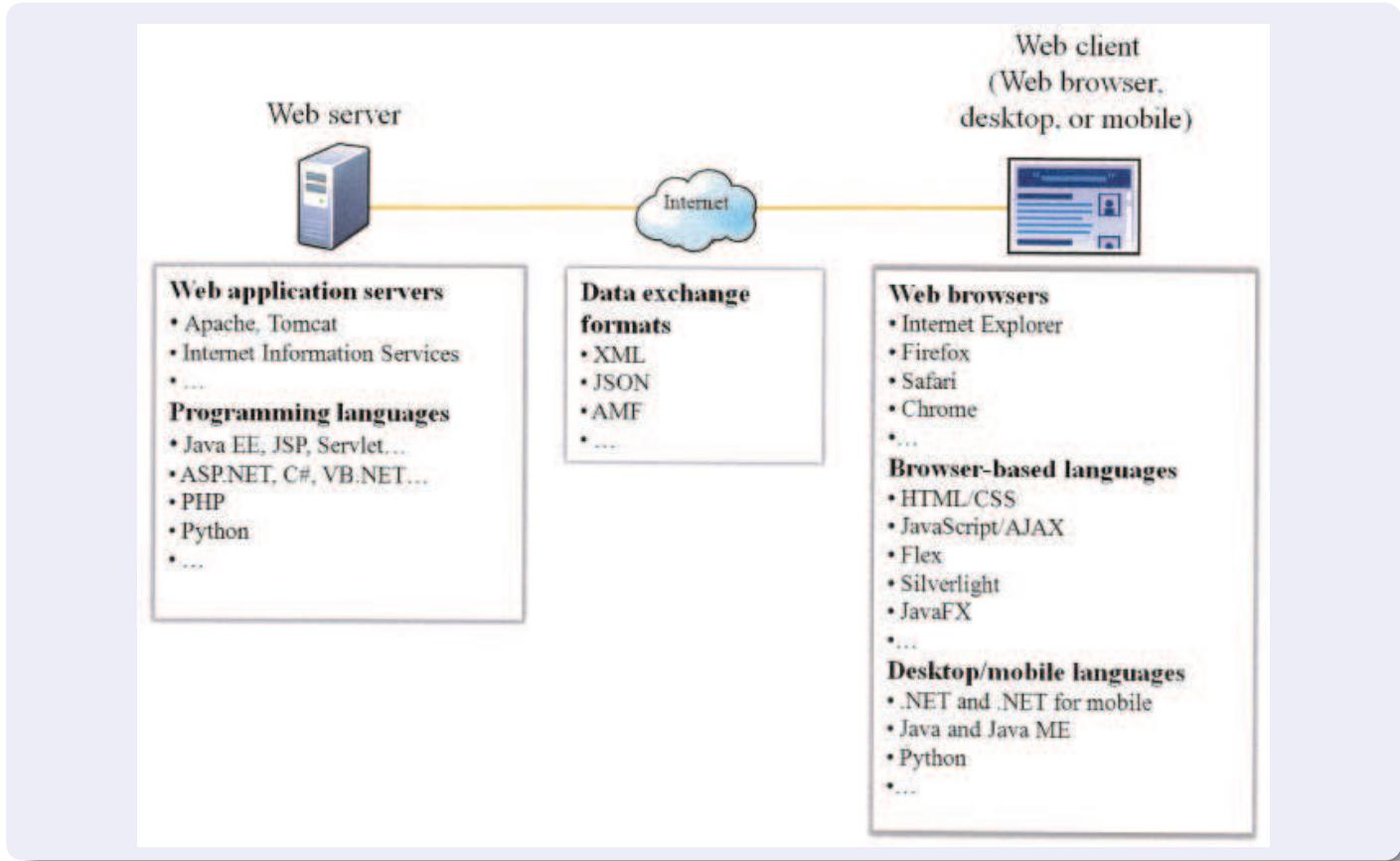
- 1 Historia de los GIS
  - 2 Tecnología web convencional
    - Arquitectura y tecnología
    - Limitaciones
  - 3 Servidores de cartografía
  - 4 Clientes de cartografía
  - 5 Geospatial mashups
  - 6 Bibliografía

# Arquitectura y tecnología

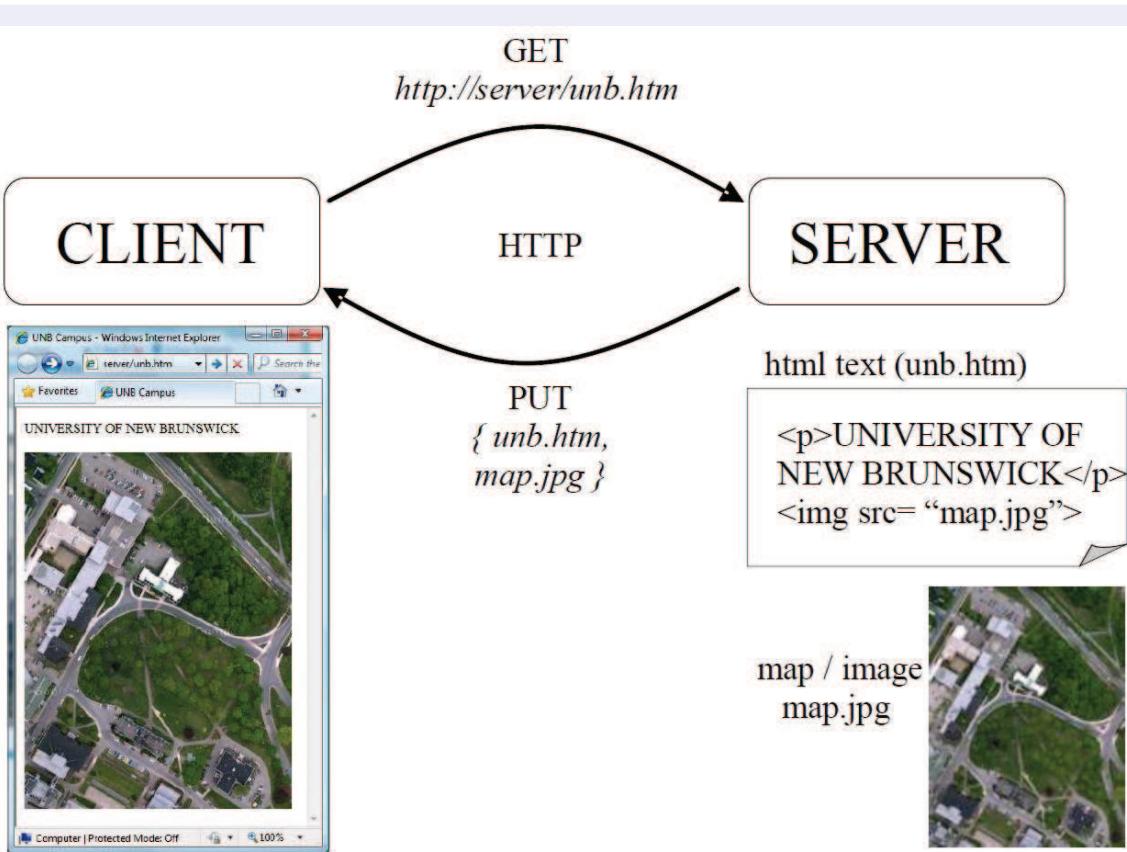
## Arquitectura Web



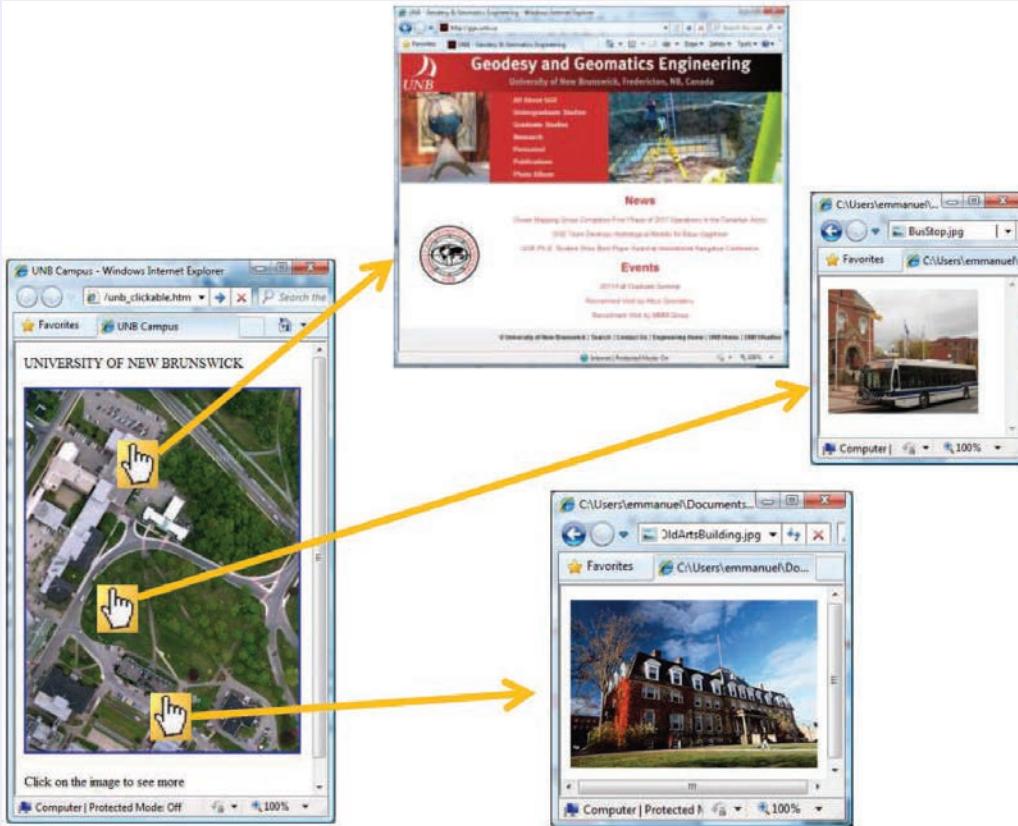
# Tecnología Web



## Mapas estáticos



# Mapas estáticos con zonas clickables



## Limitaciones

# Limitaciones de la tecnología web convencional

## Funcionalidad

- Satisface la demanda de texto e imágenes estáticas.

## Requisitos a nivel servidor

- Mapa a varios niveles de zoom no predeterminados.
- Selección de zonas de una imagen.

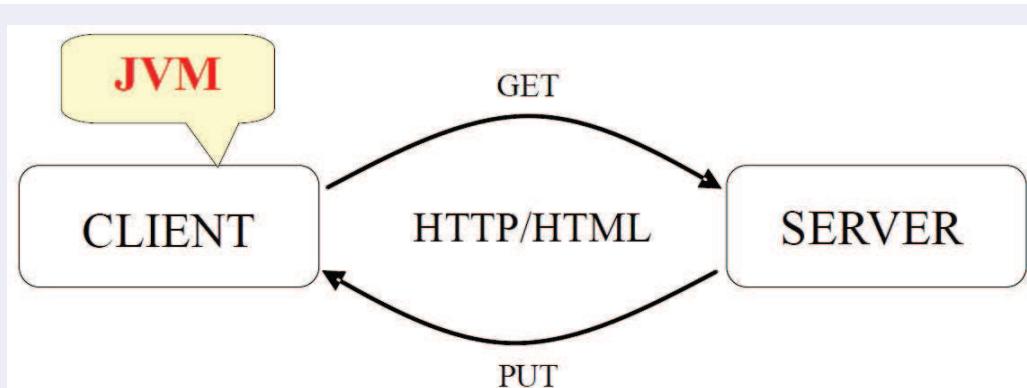
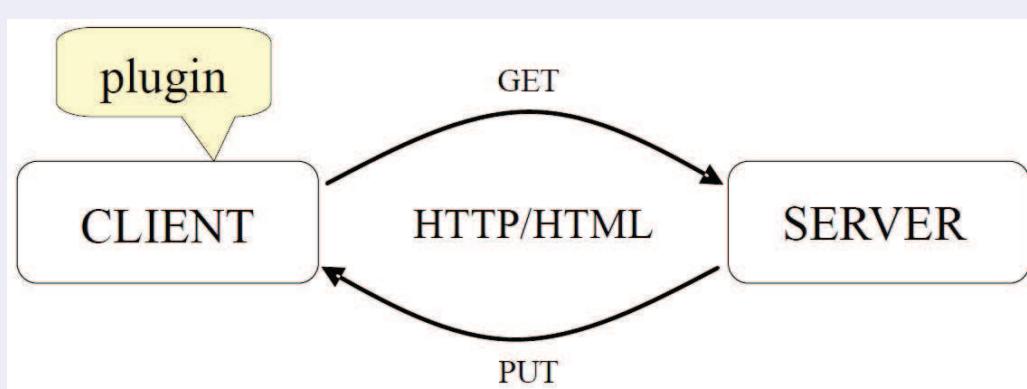
Requerirían programación específica en el servidor.

## Requisitos a nivel cliente

- Representación de mapas.
- Edición de mapas.

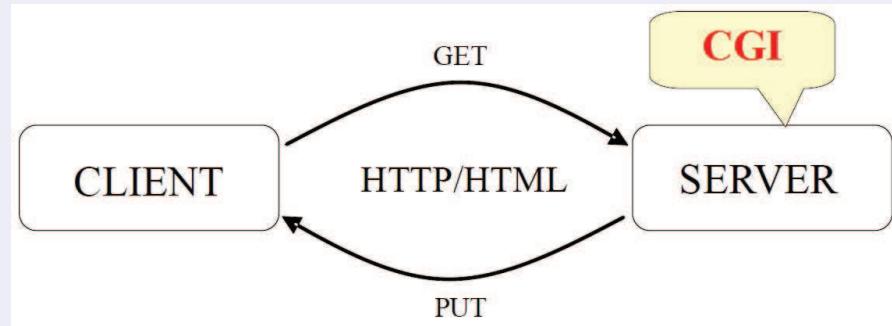
Requerirían programación específica en el cliente.

# Ampliación de funcionalidades del cliente

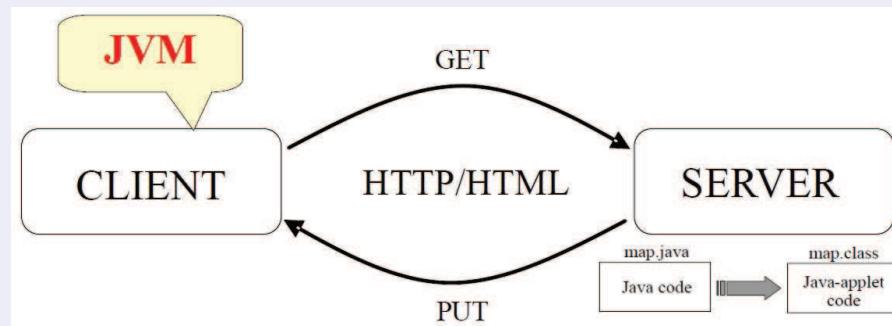


# Ampliación de funcionalidades del servidor

## CGI (Common Gateway Interface)



## JVM (Java Virtual Machine)

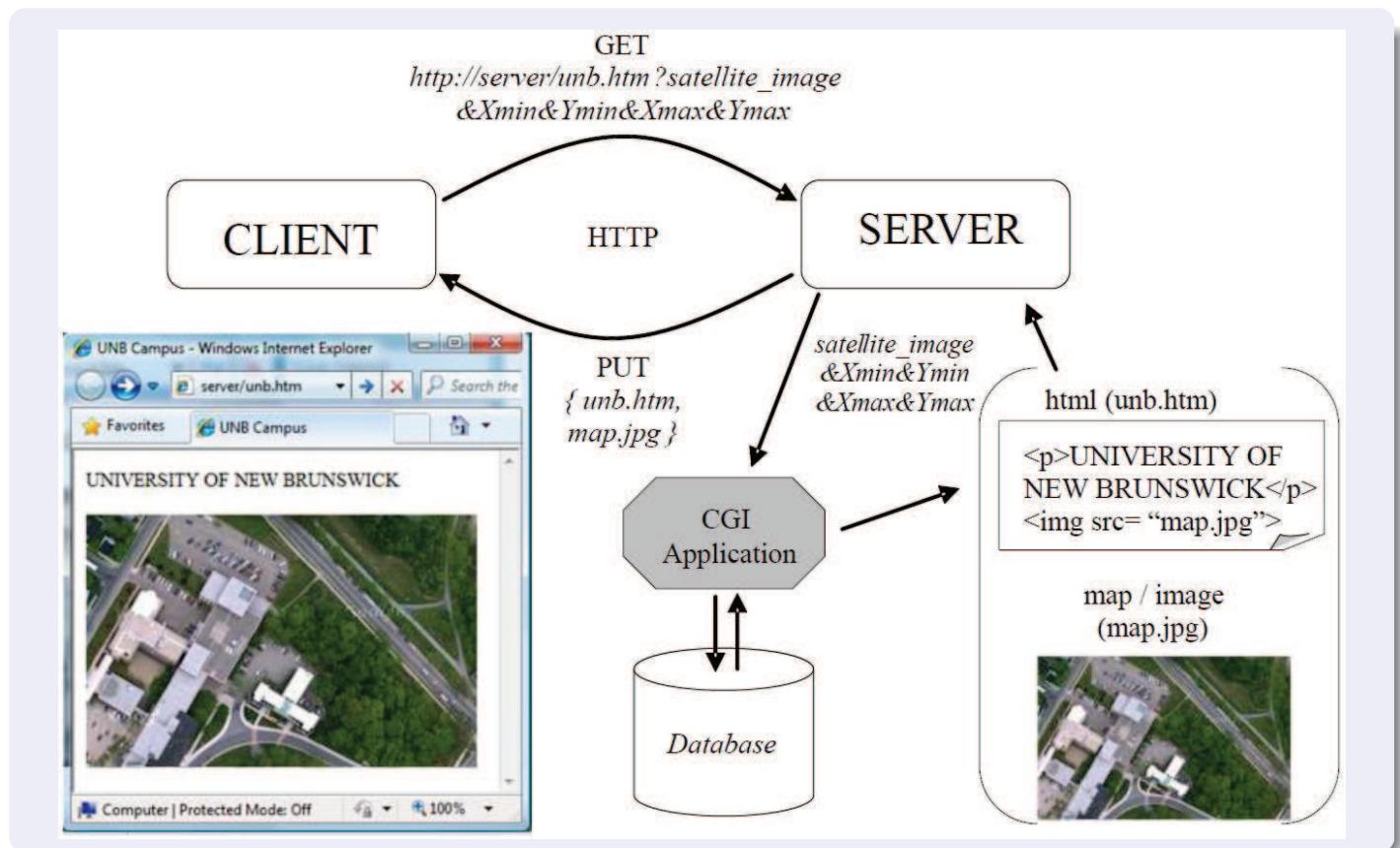


## Servidores de cartografía

- 1 Historia de los GIS
- 2 Tecnología web convencional
- 3 Servidores de cartografía
  - Funcionalidad del servidor
  - Comparativa de servidores
  - Optimización
  - Distribución de la funcionalidad
- 4 Clientes de cartografía
- 5 Geospatial mashups
- 6 Bibliografía

# Funcionalidad del servidor

## Ejemplo: servidor de cartografía con CGI



# Servidor de cartografía

## Objetivo

- Generar imágenes estáticas a medida para entregar a los clientes.
- Almacenar imágenes previamente generadas para satisfacer futuras peticiones.

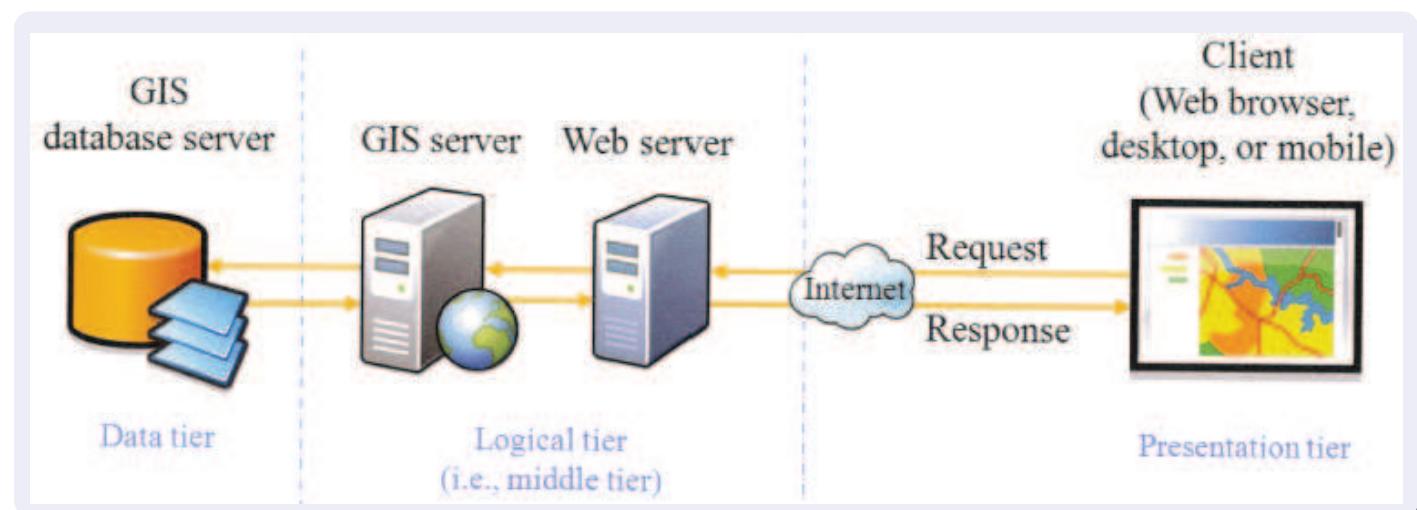
## Ejemplos open source

- **GeoServer** <http://geoserver.org/>
- **MapServer** <http://mapserver.org/>
  - ▶ <http://mapserver.org/introduction.html#installation-and-requirements>
- **QGIS Sever** [https://docs.qgis.org/2.14/es/docs/user\\_manual/working\\_with\\_ogr/ogr\\_server\\_support.html](https://docs.qgis.org/2.14/es/docs/user_manual/working_with_ogr/ogr_server_support.html)
  - ▶ <http://demo.qgis.org/>

## Ejemplo propietario

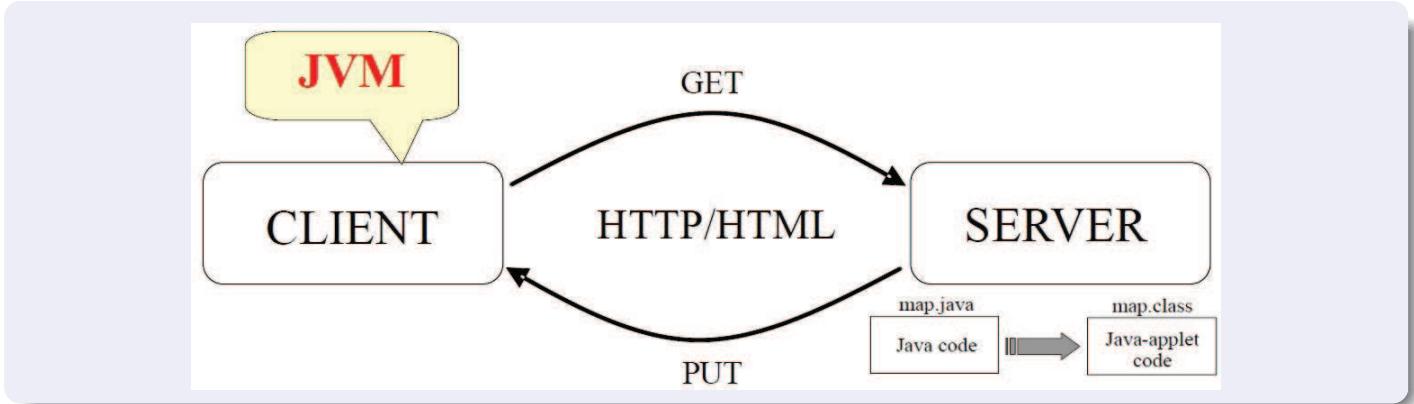
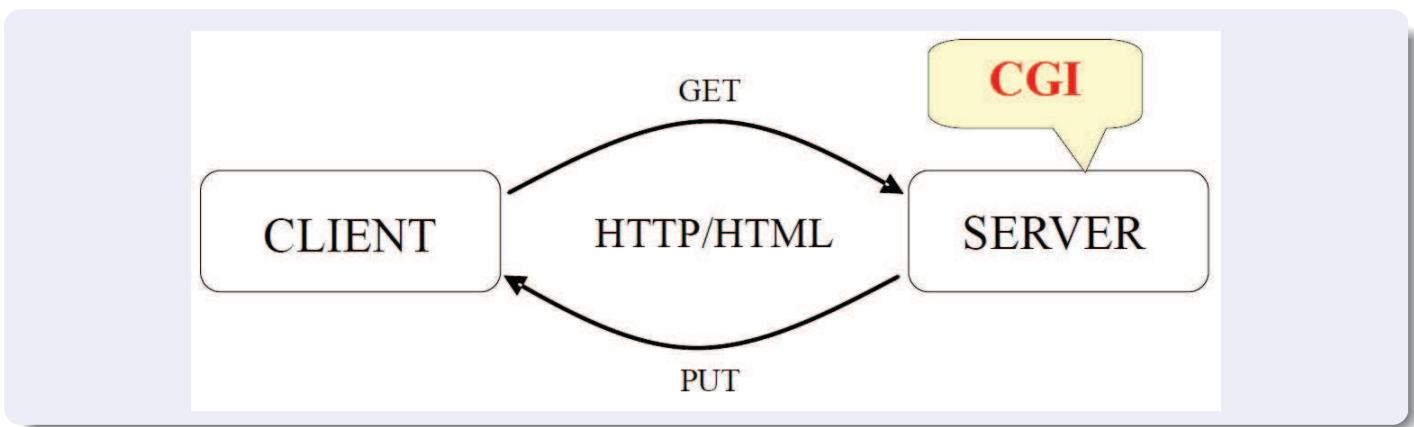
- ArcGIS.

# Servidor Web y servidor de cartografía



# Comparativa de servidores

## Ampliación de funcionalidades del servidor



# Requisitos de los servidores

Service	MapServer	GeoServer	QGIS Server
Java runtime (JRE)	No	Yes	No
Python	No*	No	No*
PHP	No*	No	No
.NET	No*	No	No
CGI/Fast-CGI	Yes	No	Yes

- No\*: Funciona sin el servicio pero, si dispone de él, ofrece más posibilidades.

## OWS (OGC Web Services)

### WMS (Web Map Service)

- Obtención a partir de ráster y vectores de imágenes georreferenciadas en formato JPEG, PNG, TIFF (y otros).

### WMTS (Web Map Tile Service)

- Similar a WMS pero con un mecanismo adicional de almacenamiento caché para reutilizar imágenes.

### WFS (Web Feature Service)

- Obtención de vectores en formato XML (GML o KML) o GeoJSON. Se usa en conjunción con WMS.

### WFS-T (Web Feature Service Transactional)

- Permite editar vectores sin que los usuarios accedan directamente a la BD.

### WCS (Web Coverage Service)

- Peticiones de cobertura geográfica (WMS y WFS en un servicio).

# Soporte OWS (OGC Web Services)

Service	MapServer	GeoServer	QGIS Server
WMS	Yes	Yes	Yes
WMTS	Yes <sup>a</sup>	Yes	No
WFS	Yes	Yes	Yes
WFS-T	Yes <sup>b</sup> (for PostGIS)	Yes	Yes
Custom <sup>b</sup>	Yes	Yes	Yes

a. Support is available via an extra downloadable plug-in or library.

b. The product has its own custom protocols that provide functionality beyond what is defined in the OGC standards.

## Formatos de SGBD espaciales soportados

Service	MapServer	GeoServer	QGIS Server
Oracle Spatial/Locator	Yes*	Yes*	Yes*
SQL Server 2008/2012	Yes*	Yes*	Yes
PostGIS geography	Yes	Yes	Yes
PostGIS raster	Yes	Yes*	Yes
Basic raster	Yes	Yes	Yes
MrSID	Yes*	Yes*	Yes*
SpatiaLite	Yes*	Yes	Yes
MySQL	Yes*	Yes*	Yes

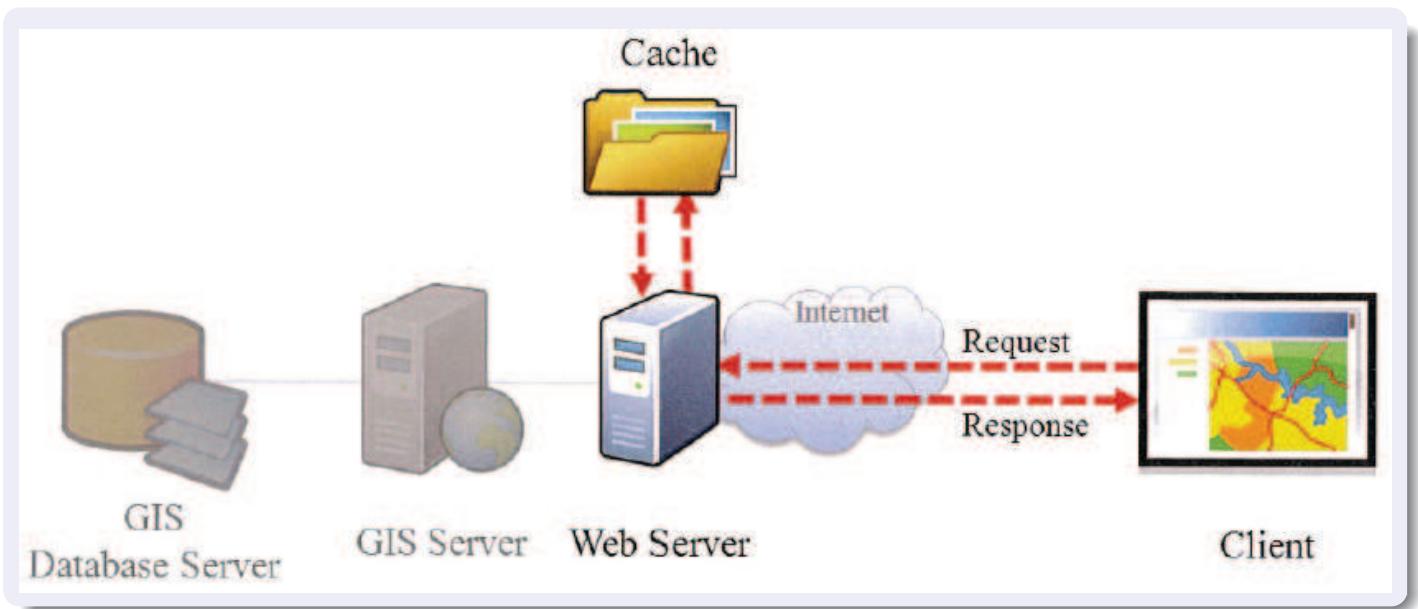
\* Support is available via an extra downloadable plug-in or library.

# Ejemplo: QGIS Server

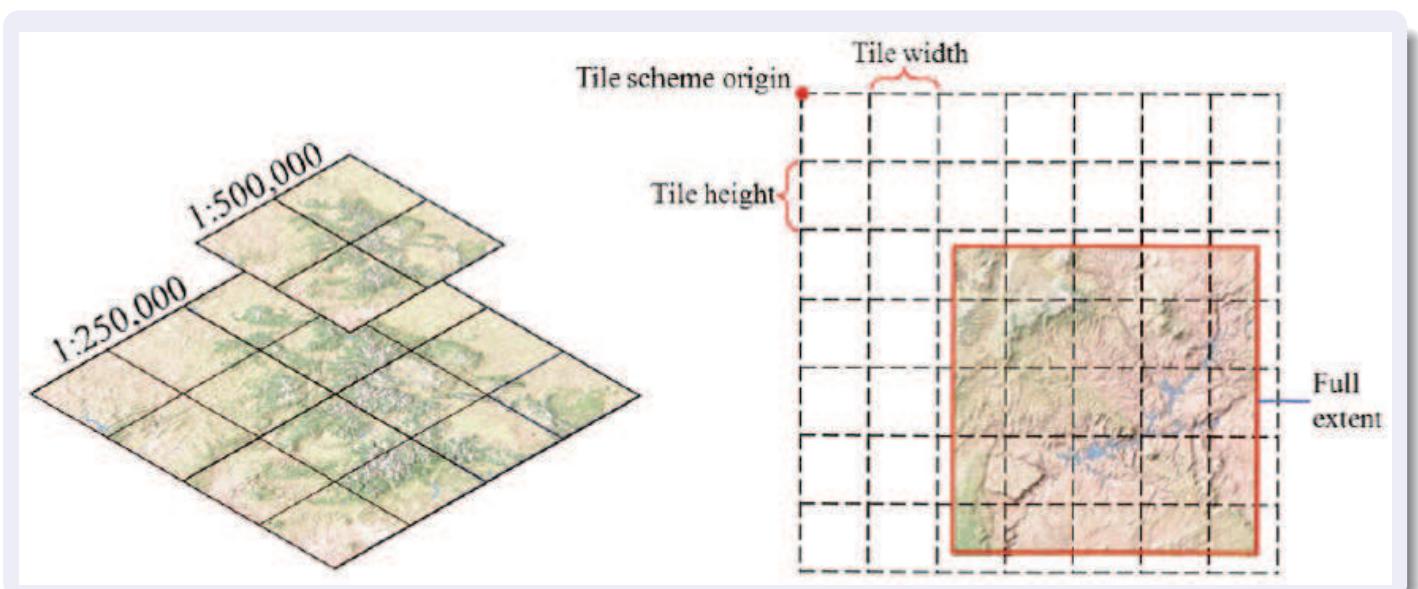
The screenshot shows a web browser displaying the QGIS Documentation website. The header includes the QGIS logo, version 2.2, the text 'DOCUMENTATION QGIS 2.2', and a language selection button set to 'English'. The main title is 'QGIS as OGC Data Server'. Below the title, there is a detailed description of QGIS Server, mentioning its implementation of WMS 1.3, WFS 1.0.0, and WCS 1.1.1, and its advanced cartographic features. It also notes its integration with FastCGI/CGI and various web servers like Apache and Lighttpd, and its funding by EU projects Orchestra, Sany, and the city of Uster in Switzerland. The text continues to explain QGIS Server's use of QGIS as a back end, the Qt library for graphics, and C++ programming. It highlights the 'Publish to Web' plugin, which exports layers and symbology for QGIS Server. The text concludes by stating that both desktop and server versions use the same visualization libraries, resulting in consistent maps on the web.

## Optimización

## Optimización mediante cache (i)

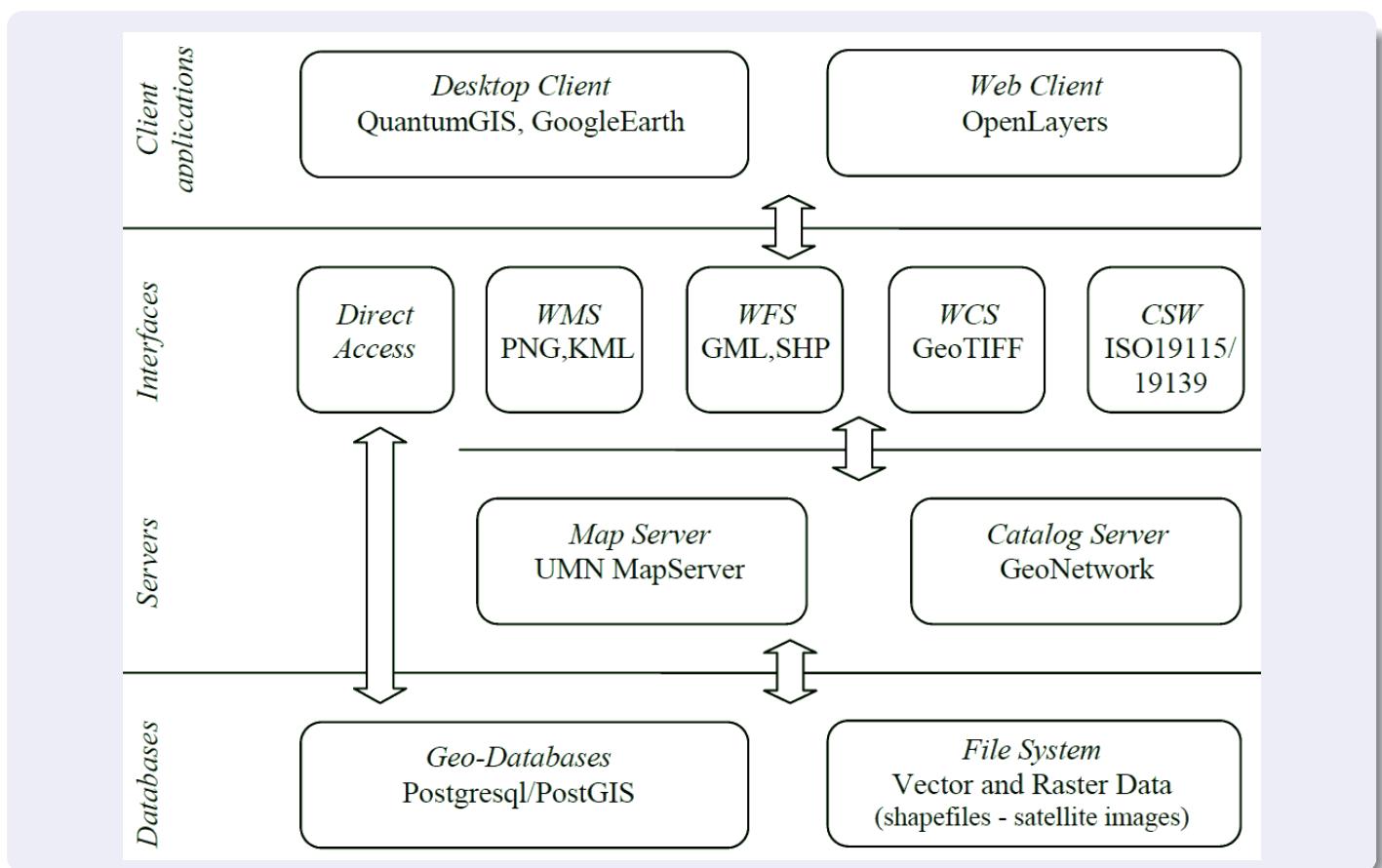


## Optimización mediante cache (y ii)

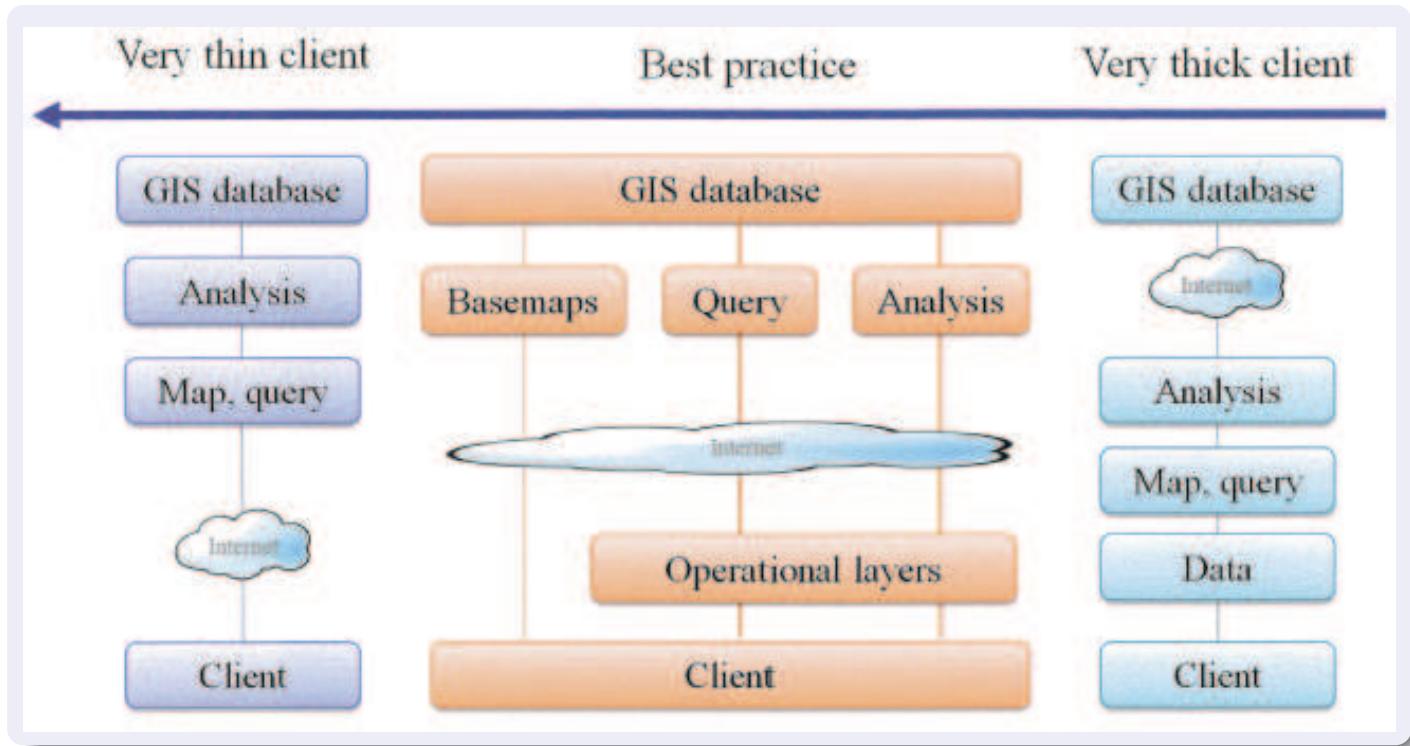


# Distribución de la funcionalidad

## Arquitectura



# Distribución de la funcionalidad



## Clients de cartografía

- 1 Historia de los GIS
- 2 Tecnología web convencional
- 3 Servidores de cartografía
- 4 Cientes de cartografía
  - Tipos de clientes
  - Librerías
  - Plugins
  - Servicios propietarios
  - Ejemplos
- 5 Geospatial mashups
- 6 Bibliografía

# Tipos de clientes

## Tipos de clientes de cartografía

### Clientes desktop

- Open source: QGIS, uDig, gvSIG, OpenJUMP.
- Propietarios: ArcGIS, Manifold, MapInfo, Cadcorp SIS.

### Clientes web

- Librerías
  - ▶ OpenLayers: se usa con GeoExplorer de OpenGeo.
  - ▶ Leaflet: usada por OpenStreetMap y CartoDB.
- Plugins
  - ▶ SVG Viewer
  - ▶ Google Earth

### Servicios propietarios

- Google Maps, Bing, MapQuest.

# Librerías

## OpenLayers

The screenshot shows the official OpenLayers website. The header features the OpenLayers logo and navigation links for Docs, Examples, API, and Code. Below the header is a large map of the world. The tagline reads: "A high-performance, feature-packed library for all your mapping needs." A message box informs users about the latest version (v4.2.0) and provides links to documentation and examples. The "OVERVIEW" section explains that OpenLayers is a free, open-source JavaScript library for displaying maps. The "FEATURES" section highlights "Tiled Layers" (supporting OSM, Bing, MapBox, Stamen, and XYZ sources), "Vector Layers" (rendering data from GeoJSON, TopoJSON, KML, GML, and Mapbox vector tiles), "Cutting Edge, Fast & Mobile Ready" (leveraging Canvas 2D, WebGL, and HTML5), and "Easy to Customize and Extend" (allowing customization via CSS and API hooks). Each feature is accompanied by a small icon.

# Leaflet (i)

The screenshot shows the official Leaflet website. At the top is the Leaflet logo with a green leaf icon. Below it is the tagline "an open-source JavaScript library for mobile-friendly interactive maps". A navigation bar includes links for Overview, Tutorials, Docs, Download, Plugins, and Blog. A news banner at the top states "Aug 8, 2017 — Leaflet 1.2.0 has been released." The main content area features a paragraph about Leaflet's design principles and its status as the leading open-source library. It includes a map of Hyde Park in London with a blue marker and a callout box that says "A pretty CSS3 popup. Easily customizable." At the bottom of the page is a footer with social media icons and a copyright notice.

# Leaflet (y ii)

This screenshot shows the "Features" page of the Leaflet documentation. It is organized into several sections:

- Layers Out of the Box**: Includes Tile layers, WMS, Markers, Popups, Vector layers (polylines, polygons, circles, rectangles), Image overlays, and GeoJSON.
- Customization Features**: Includes Pure CSS3 popups and controls for easy restyling, Image- and HTML-based markers, A simple interface for custom map layers and controls, Custom map projections (with EPSG:3857/4326/3395 out of the box), and Powerful OOP facilities for extending existing classes.
- Map Controls**: Includes Zoom buttons, Attribution, Layer switcher, and Scale.
- Interaction Features**: Includes Drag panning with inertia, Scroll wheel zoom, Pinch-zoom on mobile, Double click zoom, Zoom to area (shift-drag), Keyboard navigation, Events: click, mouseover, etc., and Marker dragging.
- Performance Features**: Includes Hardware acceleration on mobile makes it feel as smooth as native apps, Utilizing CSS3 features to make panning and zooming really smooth, Smart polyline/polygon rendering with dynamic clipping and simplification makes it very fast, Modular build system for leaving out features you don't need, and Tap delay elimination on mobile.
- Browser Support**: Lists Desktop support for Chrome, Firefox, Safari 5+, Opera 12+, and IE 7-11.
- Mobile**: Lists support for Safari for iOS 7+, Android browser 2.2+, 3.1+, 4+, Chrome for mobile, Firefox for mobile, and IE10+ for Win8 devices.
- Misc**: Includes Extremely lightweight and No external dependencies.
- Visual Features**: Includes Zoom and pan animation, Tile and popup fade animation, Very nice default design for markers, popups and map controls, and Retina resolution support.

# Plugins

## SVG



**Graphics Markup for the Web**

SVG is a markup language for describing two-dimensional graphics applications and images, and a set of related graphics script interfaces. [SVG 1.1 2nd edition](#) is a W3C Recommendation and is the most recent version of the full specification.

SVG is supported by all modern browsers for desktops and mobiles. Some features, such as SMIL animation and SVG Fonts are not as widely supported. There are many SVG authoring tools, and export to SVG is supported by all major vector graphics authoring tools.

```
<!DOCTYPE html>
<html>
<body>

<svg height="210" width="500">
  <polygon points="100,10 40,198 190,78 10,78
160,198" style="fill:lime;stroke:purple;
stroke-width:5;fill-rule:nonzero;">
  Sorry, your browser does not support inline
SVG.
</svg>
</body>
</html>
```

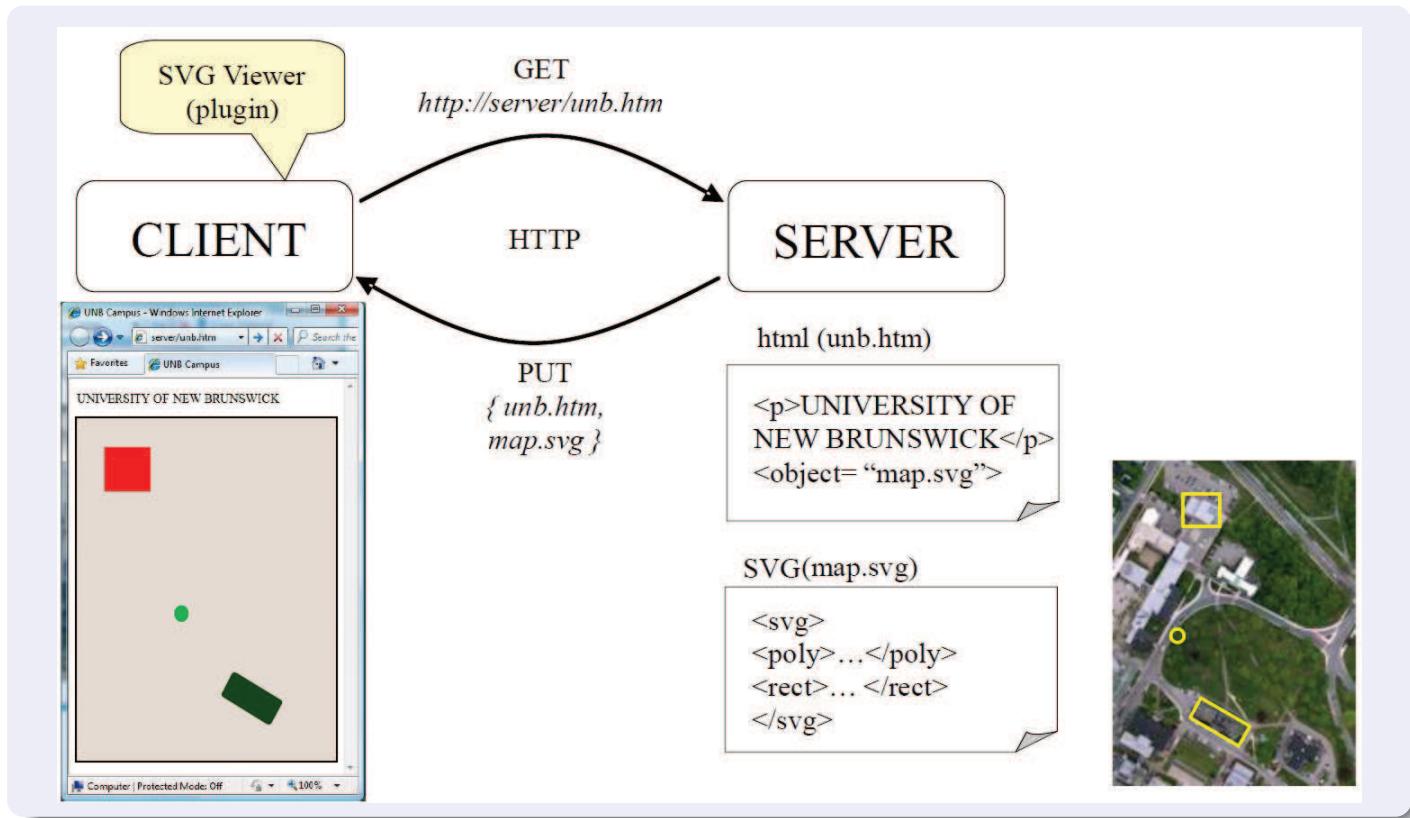


```
<!DOCTYPE html>
<html>
<body>

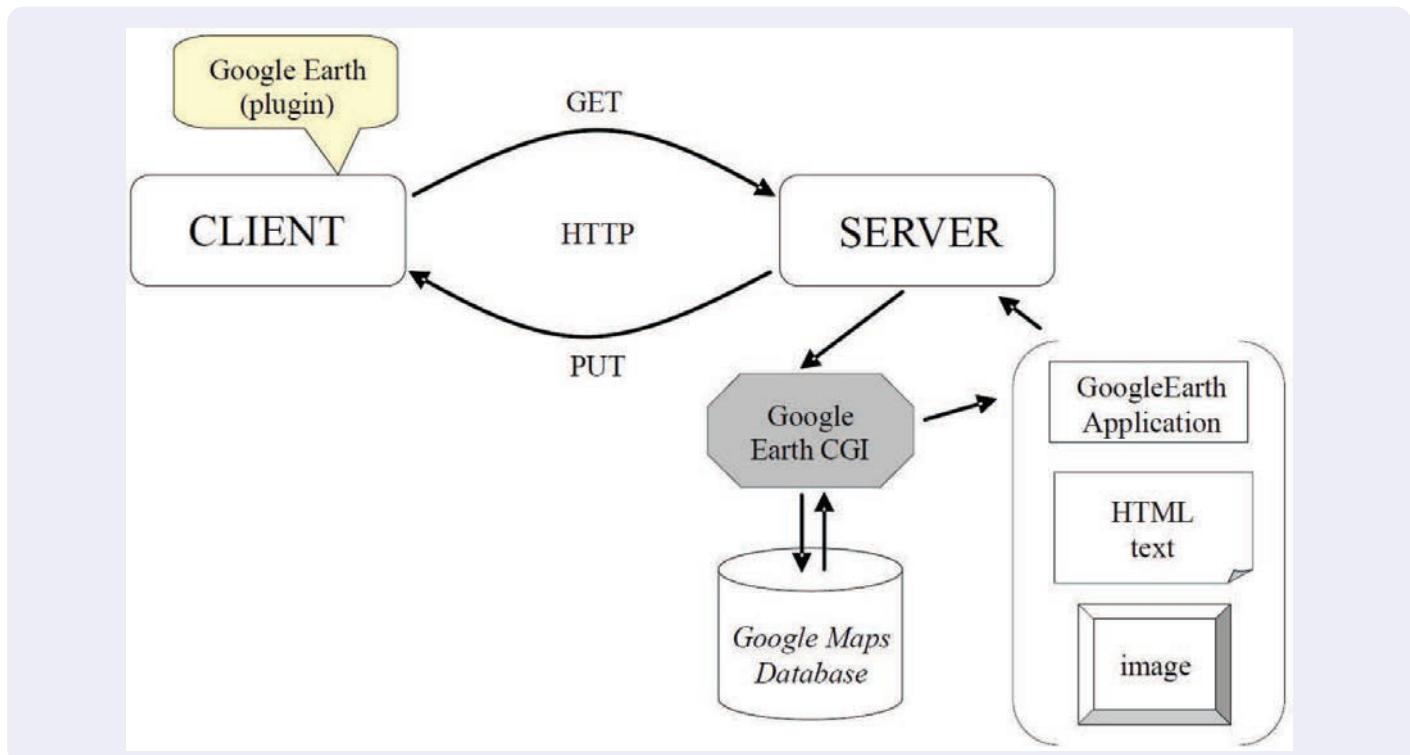
<svg height="210" width="500">
  <polygon points="100,10 40,198 190,78 10,78
160,198" style="fill:lime;stroke:purple;
stroke-width:5;fill-rule:evenodd;">
  Sorry, your browser does not support inline
SVG.
</svg>
</body>
</html>
```



# SVG Viewer plugin



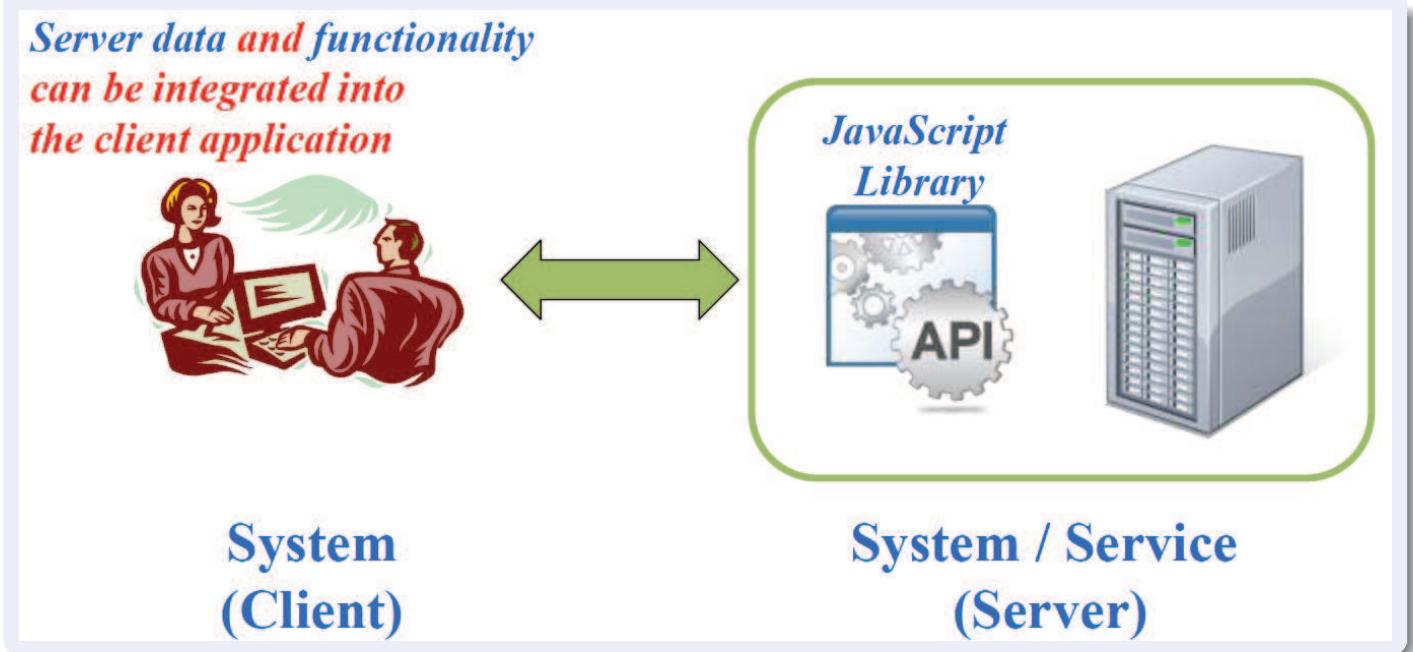
# Google Earth plugin



- Permite ver los mapas de Google Earth integrados en las páginas web.

# Servicios propietarios

## API JavaScript



# Servicios propietarios

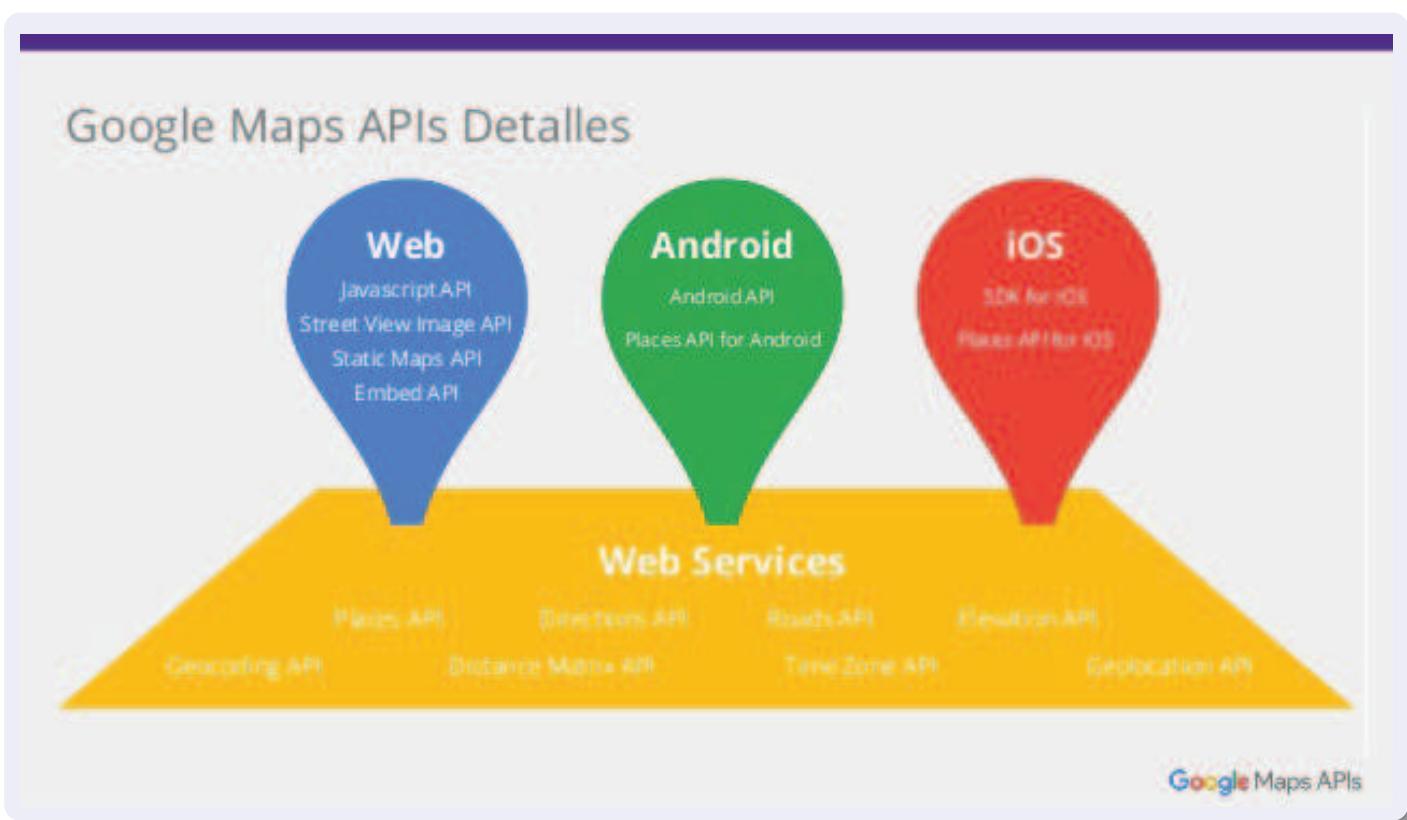
## Servicios propietarios

- Google Maps
- Bing
- MapQuest

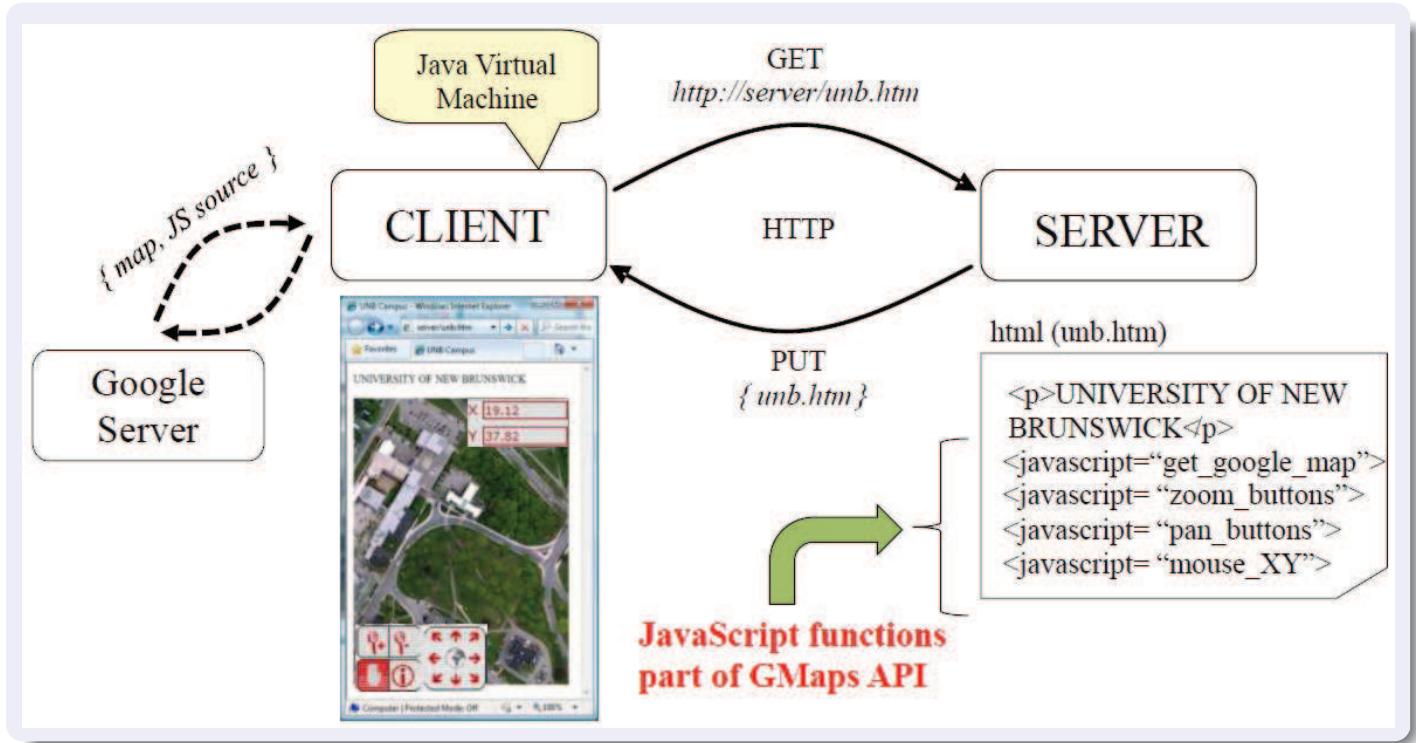
## Características

- Empaquetan cliente, servidor y datos.
- Accesible mediante API JavaScript particular.
- Son poco flexibles.
- Gratis si se usan sin fines comerciales, no para uso comercial.

# Google Maps API



# Uso de Google Maps API



## Ejemplos

# GeoServer con GoogleMaps, OpenLayers y Leaflet

## GoogleMaps

```
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map { height: 100%; width: 100% }
    </style>
    <script type="text/javascript" src="https://maps.google.com/maps/api/js?sensor=false&language=en"></script>
    <script type="text/javascript" src="../base.js"></script>
    <script type="text/javascript" src="../WMS.js"></script>
    <script type="text/javascript" src="map.js"></script>
  </head>
  <body onload="mapinitialize();">
    <div id="map"></div>
  </body>
</html>
```

## OpenLayers

```
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map_canvas { height: 100% }
    </style>
    <script type="text/javascript" src="http://openlayers.org/api/2.13/OpenLayers.js"></script>
    <script type="text/javascript" src="../base.js"></script>
    <script type="text/javascript" src="map.js"></script>
  </head>
  <body onload="mapinitialize();">
    <div id="map" style="width:100%; height:100%"></div>
  </body>
</html>
```

## Leaflet

```
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.4/leaflet.css" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map { height: 100%; width: 100%; background-color:#badbf1; }
    </style>
    <script src="http://cdn.leafletjs.com/leaflet-0.4/leaflet.js"></script>
    <script type="text/javascript" src="base.js"></script>
    <script src="map.js"></script>
  </head>
  <body onload="mapinitialize();">
    <div id="map"></div>
  </body>
</html>
```

# GeoServer con OpenLayers y Leaflet

```
var GEOSERVERBASE = "http://localhost:8082";
var CountyLayer = "sig:municipios";
```

## OpenLayers

```
function mapinitialize() {
  var bounds = new OpenLayers.Bounds(
    100511.0,3987329.0,621383.8967,4288889.0
  );

  var options = {
    maxExtent: bounds,
    projection: 'EPSG:23030',
    units: 'meters'
  };
  map = new OpenLayers.Map('map', options);

  var countries = new OpenLayers.Layer.WMS(
    CountyLayer, GEOSERVERBASE + '/geoserver/sig/wfs',
    {
      layers: CountyLayer
    }
  );

  map.addLayer(countries);

  map.zoomTo(1);
  map.panTo(new OpenLayers.LonLat(-4.7086738,37.8550964));
}
```

## Leaflet

```
function mapinitialize() {

  counties = new L.TileLayer.WMS(GEOSERVERBASE + "/geoserver/sig/wms",
  {
    layers: CountyLayer,
    format: 'image/png',
    transparent: true,
    attribution: ""
  });

  map = new L.Map('map',
  {
    center: new L.LatLng(37.8550964,-4.7086738),
    zoom: 7,
    layers: [counties],
    zoomControl: true
  });
}
```

# GeoServer con GoogleMaps

```
//Create a google map
map = new google.maps.Map(document.getElementById("map"),mapOptions);

//add all the overlays we want to this array
var overlayMaps =[

    {
        getTileUrl: function(coord, zoom)
        {
            var IULP = new google.maps.Point(coord.x*256,(coord.y+1)*256);
            var ILRP = new google.maps.Point((coord.x+1)*256,coord.y*256);

            var projectionMap = new MercatorProjection();

            var IULg = projectionMap.fromDivPixelToSphericalMercator(IULP, zoom);
            var ILRg = projectionMap.fromDivPixelToSphericalMercator(ILRP, zoom);

            var IUL_Latitude = IULg.y;
            var IUL_Longitude = IULg.x;
            var ILR_Latitude = ILRg.y;
            var ILR_Longitude = ILRg.x;

            if (ILR_Longitude < IUL_Longitude){
                ILR_Longitude = Math.abs(ILR_Longitude);
            }
            return GEOSERVERBASE + "/geoserver/wms?" + wmsparams.join("&") + "&layers=" +
                CountyLayer + "&bbox=" + IUL_Longitude + "," + IUL_Latitude + "," + 
                ILR_Longitude + "," + ILR_Latitude;
        },
        tileSize: new google.maps.Size(256, 256),
        isPng: true,
        maxZoom: 15,
        minZoom: 4,
        alt: 'Counties'
    }
];

//add all overlays to the map
for (i=0; i<overlayMaps.length; i++){
    var overlayMap = new google.maps.ImageMapType(overlayMaps[i]);
    map.overlayMapTypes.push(overlayMap);
    map.overlayMapTypes.setAt(overlayMaps[i],overlayMap);
}
```

## Geospatial mashups

- 1 Historia de los GIS
- 2 Tecnología web convencional
- 3 Servidores de cartografía
- 4 Clientes de cartografía
- 5 Geospatial mashups
- 6 Bibliografía

## Web 2.0

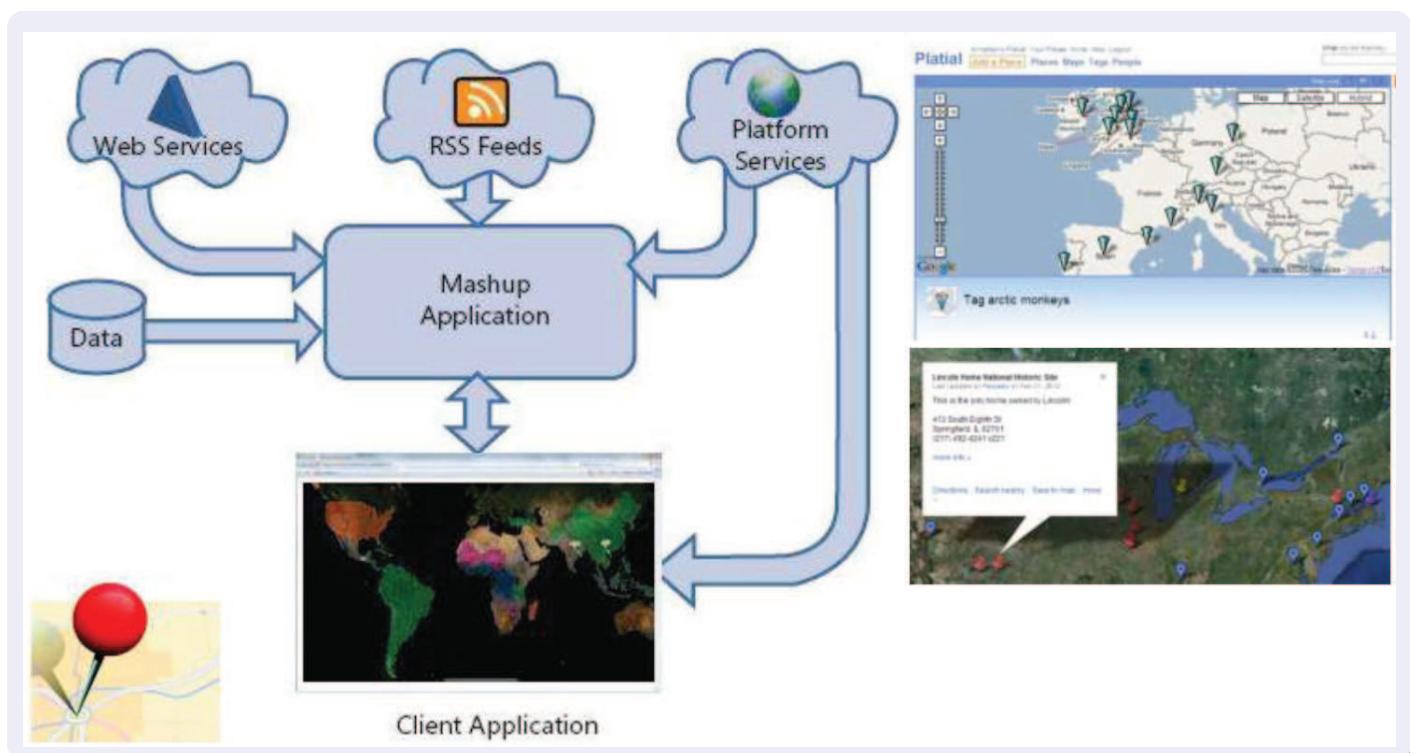


José Samos Jiménez ( 2020 jsamos (lsi-ugr) ISIG. Tema 6: Presentación (Servidores de Ma

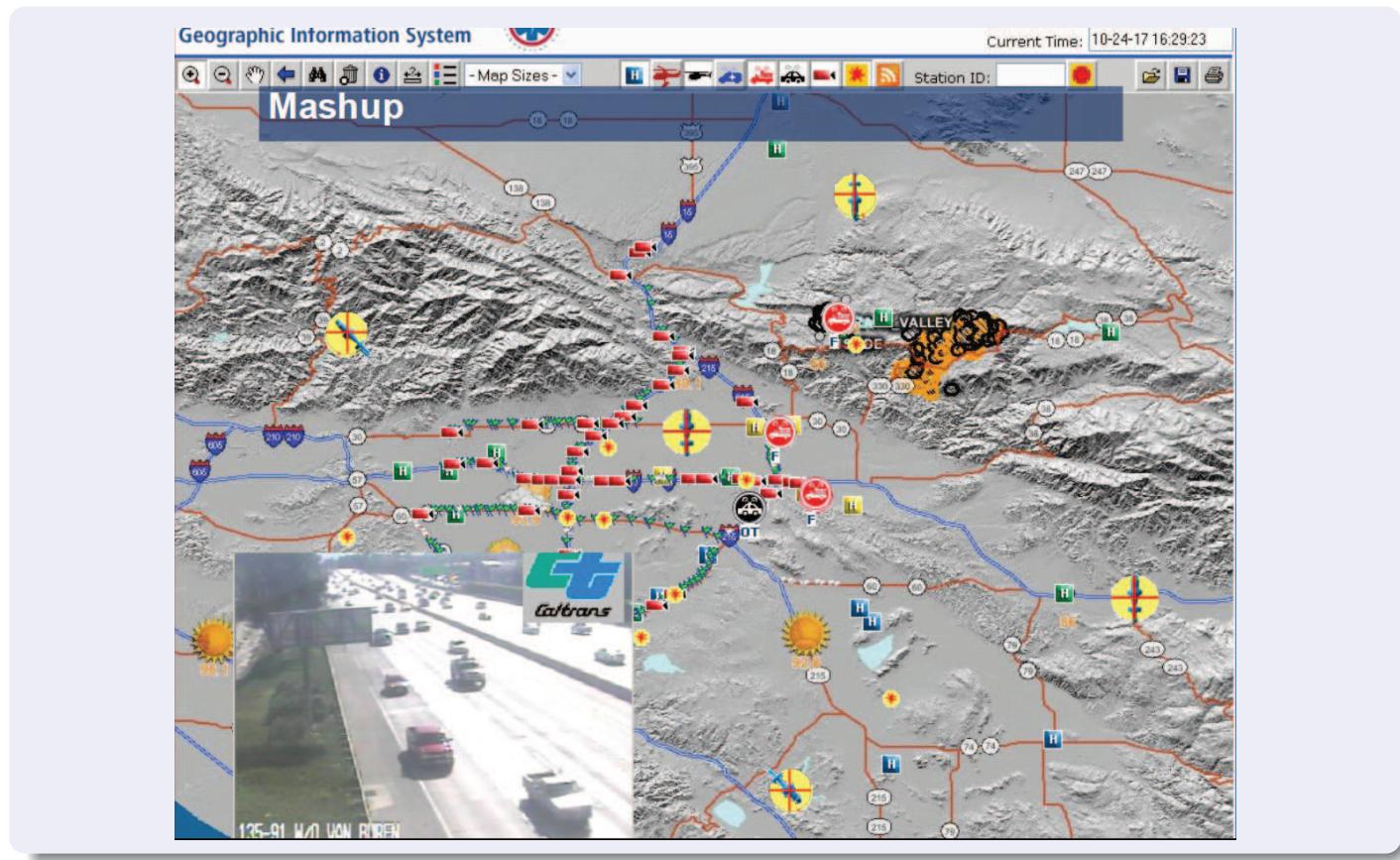
Curso 2020-21

57 / 64

# Maps o geospatial mashups



# Geospatial mashups

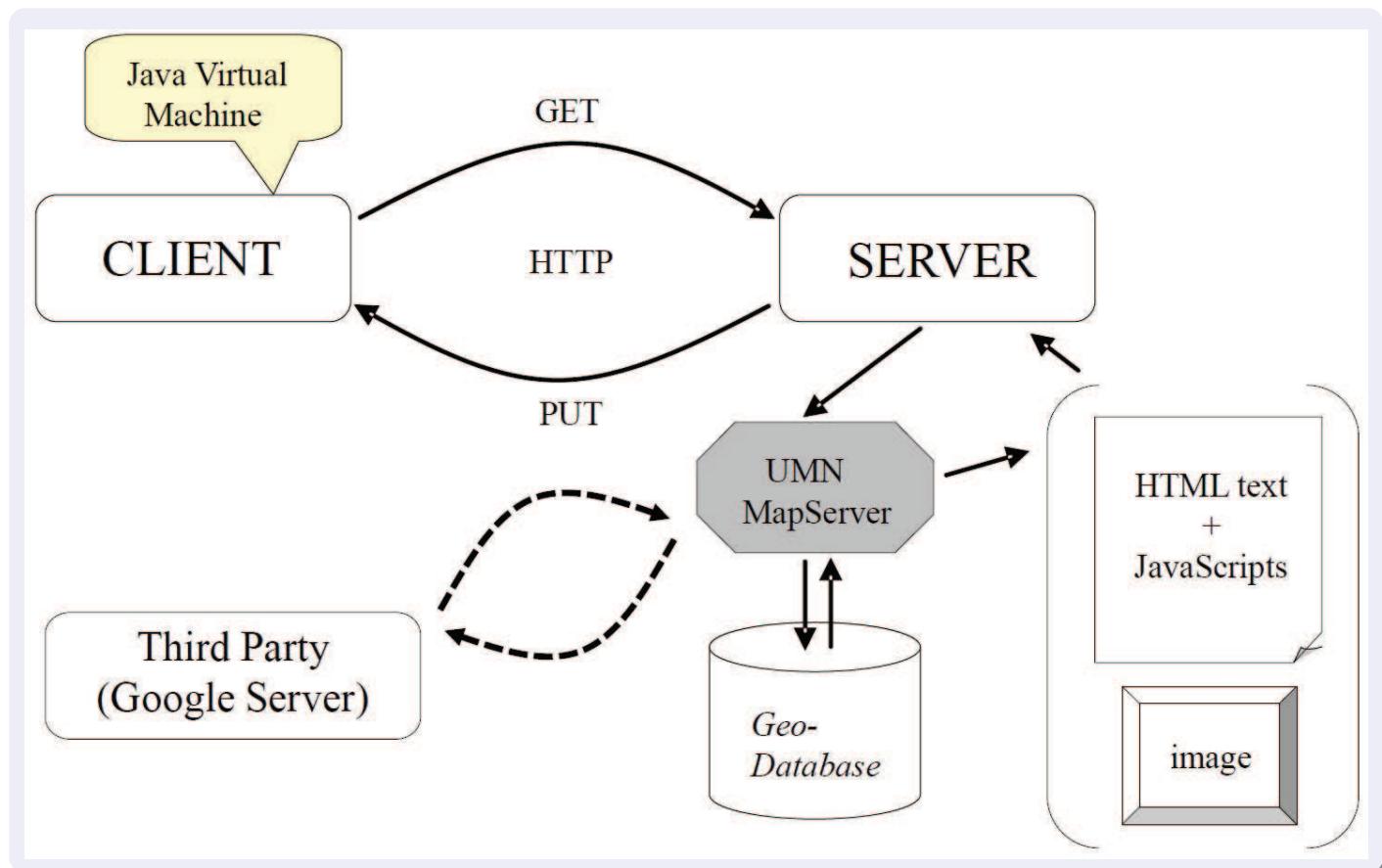


José Samos Jiménez ( 2020 jsamos (lsi-ugr) |SIG. Tema 6: Presentación (Servidores de Ma

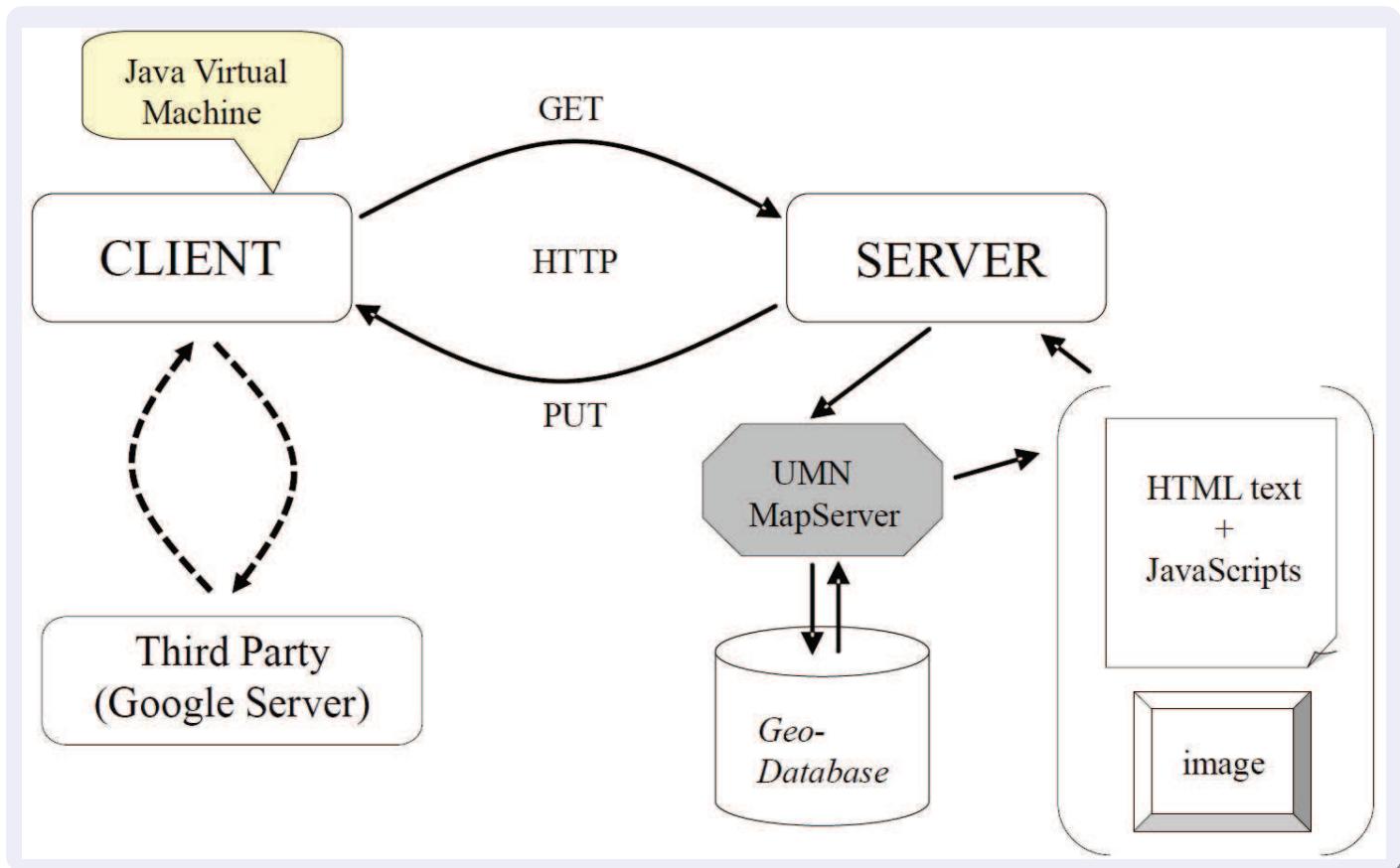
Curso 2020-21

59 / 64

# Mashups en el servidor



# Mashups en el cliente



[www.ushahidi.com](http://www.ushahidi.com)

The screenshot shows the Ushahidi website. At the top left is the Ushahidi logo. The top right features a navigation bar with links for "Features", "Plans", "Enterprise", "MORE", "CONTACT", and "LOG IN". The main header reads "READ THE CROWD" with the tagline "Don't just get the data. Get the whole story." Below the header is a call-to-action box: "Create your first deployment for free" with fields for "Your email address" and a "GET STARTED" button, along with links for "or contact sales" and "or download open source code from GitHub now". In the center-left, there's a smartphone displaying a map with location pins and a message: "Overcrowding between 5th St. & Salem Ave." and "Jess Shortland - 5 mins. via SMS". A large map of a city is visible in the background. At the bottom left, a mobile device screen shows a survey interface with various questions and options. To the right of the device, the text "Collect meaningful data" is displayed with the subtext "Gather data from any device with custom surveys and crowdsourcing tools". Arrows point from the device screen to icons for email, phone, Twitter, and computer, representing different data collection methods.

# Bibliografía

- 1 Historia de los GIS
- 2 Tecnología web convencional
- 3 Servidores de cartografía
- 4 Clientes de cartografía
- 5 Geospatial mashups
- 6 Bibliografía

## Bibliografía

- Dav07 Scott Davis. *GIS for Web Developers. Adding Where to Your Web Applications.* The Pragmatic Programmers, 2007.
- FS11 Pinde Fu and Jiulin Sun. *Web GIS Principles and Applications.* esri Press, 2011.
- OH15 Regina O. Obe and Leo S. Hsu. *PostGIS in Action (Second Edition).* Manning, 2015.
- Ste15 Emmanuel Stefanakis. *Web Mapping & Geospatial Web Services.* CreateSpace, 2015.