



Universidad
Zaragoza

Proyecto Final de Carrera

Integración y visualización de mapas históricos

Autor

David Gayán Asensio

Directora

M^a José Pérez Pérez

Ponente

Rubén Béjar Hernández

Escuela de Ingeniería y Arquitectura

Universidad de Zaragoza

2016/2017

Agradecimientos:

Quiero agradecer a mi familia y amigos todo el apoyo brindado durante el desarrollo de este proyecto. En especial a Virginia, sin ella no habría sido posible que todo esto sea realidad.

También quiero expresar mi agradecimiento al personal del GeospatiumLab, en especial a María José, que me han guiado y apoyado a lo largo de todo este proyecto.

Integración y visualización de mapas históricos

Resumen

GeoSpatiumLab S.L, a partir de ahora GeoSLab¹, es una empresa de base tecnológica, especializada en el tratamiento digital de la información geoespacial y georreferenciada y sus ámbitos de aplicación. GeoSLab nace en enero de 2007 con el objetivo último de poner en valor fuera del ámbito universitario la tecnología desarrollada por el Grupo de Sistemas de Información Avanzados (IAAA²) de la Universidad de Zaragoza a lo largo de una trayectoria de más de 10 años. Su principal misión es crear valor para sus clientes prestando servicios de gestión avanzada de la información espacial. Una de las principales líneas de trabajo de GeoSLab es la consultoría y desarrollo de tecnología en el ámbito de las Infraestructuras de Datos Espaciales, especialmente las conformes con la Directiva Marco Europea INSPIRE.

Son numerosos los sistemas que existen en la actualidad que publican datos geográficos. De la mano de la Directiva Marco INSPIRE, han sido muchos los visores cartográficos que se han puesto en funcionamiento. Prácticamente todos ellos trabajan con datos geográficos ya creados para ser expuestos en sistemas informáticos.

En este Proyecto Fin de Carrera se aborda el reto de integrar en estas tecnologías de visualización datos que contienen información geográfica, en este caso mapas históricos antiguos, no concebidos con este fin. Concretamente, se va a partir de 20 planos de la ciudad de Zaragoza creados en los últimos 300 años y que han sido escaneados a muy alta resolución por el Archivo Municipal del Ayuntamiento de Zaragoza³.

Se desarrollarán los procesos necesarios para conseguir que los datos de partida, que no cumplen los estándares geográficos al uso, los cumplan para su publicación posterior. Además de esto, se intentará optimizar los datos, tanto en su estructura como su almacenamiento, para mejorar el acceso a los mismos y conseguir aumentar la eficiencia de los servicios donde se publiquen.

La publicación y visualización de esta información geográfica se realizará en dos ámbitos diferentes. Uno de ellos es el proyecto ENERGIC OD⁴ (*European NEtwork for Redistributing Geospatial Information to user Communities - Open Data*) donde se pretende probar la validez del concepto del mismo. Esto se llevará a cabo mediante el uso en un sistema de visualización de información geográfica de un componente desarrollado dentro del marco de este proyecto denominado *Virtual Hub*. El otro es IDEZar⁵, la infraestructura de datos espaciales del Ayuntamiento de Zaragoza, cuyo objetivo es poner a disposición de la ciudadanía y de profesionales de manera sencilla esta valiosa información, que hasta ahora no era ni accesible ni podía ser explotada correctamente al no tener asociada información geográfica que la enriquece enormemente y aumenta su valor.

¹ <http://www.geoslab.com/>

² <http://iaaa.unizar.es>

³ <http://www.zaragoza.es>

⁴ <http://www.energic-od.eu/>

⁵ <http://idezar.zaragoza.es/>

Tabla de contenidos

TABLA DE CONTENIDOS	IV
1. INTRODUCCIÓN	1
1.1. CONTEXTO PROFESIONAL.....	1
1.2. CONTEXTO TECNOLÓGICO	2
1.2.1. Sistemas de Información Geográfica (SIG)	2
1.2.2. IDEZar	5
1.2.3. Energic OD	6
1.3. MOTIVACIÓN	6
1.4. OBJETIVOS	7
1.5. METODOLOGÍA Y ACTIVIDADES	8
1.6. ESTRUCTURA DEL DOCUMENTO	11
2. TRABAJO REALIZADO	12
2.1. ARQUITECTURA DEL SISTEMA	12
2.2. TRATAMIENTO Y PROCESADO DE LOS MAPAS HISTÓRICOS.....	15
2.2.1. Estudio de herramientas	15
2.2.2. Elección de la herramienta	16
2.2.3. Tratamiento y procesamiento de los mapas históricos.....	17
2.3. SERVICIOS DE PUBLICACIÓN DE LOS MAPAS HISTÓRICOS GEORREFERENCIADOS.....	21
2.3.1. Estudio de servicios de publicación	21
2.3.2. Elección del servicio de publicación	21
2.3.3. Configuración de los servicios de publicación	22
2.4. APLICACIÓN WEB DE VISUALIZACIÓN	26
2.4.1. Estudio de la tecnología existente	26
2.4.2. Diseño de la Aplicación	28
2.4.3. Desarrollo	32
2.5. PLAN DE PRUEBAS E INTEGRACIÓN	34
3. CONCLUSIONES.....	35
3.1. RESULTADO DEL PROYECTO.....	35
3.2. LÍNEAS FUTURAS.....	36
3.3. INCIDENCIAS Y PROBLEMAS ENCONTRADOS	37
3.4. VALORACIÓN PERSONAL.....	38
ÍNDICE DE FIGURAS.....	39

ANEXO I	ESTADO DEL ARTE	41
ANEXO I.I	ESTUDIO DE LA SITUACIÓN ACTUAL.....	41
ANEXO II	ANÁLISIS DEL SISTEMA.....	45
ANEXO II.I	REQUISITOS DEL SISTEMA	45
ANEXO III	TRATAMIENTO Y PROCESAMIENTO DE LOS MAPAS HISTÓRICOS.....	51
ANEXO III.I	INSTALACIÓN QGIS	51
ANEXO III.II	INSTALACIÓN GDAL.....	52
ANEXO III.III	CARGA DE DATOS DE APOYO PARA LA GEORREFERENCIACIÓN	53
ANEXO III.IV	PROCESO DE TRATAMIENTO DE LOS MAPAS HISTÓRICOS.....	57
ANEXO IV	SERVICIOS DE PUBLICACIÓN DE INFORMACIÓN GEOGRÁFICA	65
ANEXO IV.I	INSTALACIÓN DE GEOSERVER.....	65
ANEXO IV.II	INSTALACIÓN DE VIRTUAL HUB	68
ANEXO IV.III	PUBLICACIÓN DE LOS MAPAS GEORREFERENCIADOS EN GEOSERVER.....	70
ANEXO IV.IV	PUBLICACIÓN DE LOS MAPAS GEORREFERENCIADOS EN VIRTUAL HUB.....	77
ANEXO V	APLICACIÓN DE VISUALIZACIÓN DE MAPAS HISTÓRICOS.....	85
ANEXO IV.V	DESARROLLO DE LA APLICACIÓN WEB	85
ANEXO IV.VI	MANUAL DE USUARIO DE LA APLICACIÓN WEB	88
ANEXO VI	PLAN DE PRUEBAS	93
ANEXO VII	HERRAMIENTAS Y PROGRAMAS UTILIZADOS EN EL PROYECTO.....	95

1. Introducción

El presente documento tiene como objetivo recoger toda la información relacionada con la realización de este Proyecto Final de Carrera, titulado “Integración y visualización de mapas históricos”. En esta introducción se abordará el contexto, tanto profesional como tecnológico, así como las distintas motivaciones que han dado origen a este proyecto, junto con los objetivos que se han generado y que han sido abordados.

1.1. Contexto profesional

Este Proyecto Final de Carrera se ha llevado a cabo en GeoSpatiumLab S.L, en adelante GeoSLab, una empresa especializada en el desarrollo de Sistemas de Información Geográfica (SIGs) y servicios basados en la localización. La empresa surge como Spin-Off de la Universidad de Zaragoza, y uno de sus objetivos principales es facilitar la transferencia de la tecnología generada por el Grupo de Sistemas de Información Avanzada (IAAA), perteneciente al Departamento de Informática e Ingeniería de Sistemas.

En cuanto a GeoSLab, es una empresa que nace en el 2.007 con el objetivo de dar uso a la tecnología desarrollada por parte del IAAA. Se trata de una empresa especializada en el tratamiento digital de la información geoespacial y georreferenciada y sus ámbitos de aplicación. Cuenta con una amplia experiencia en el ámbito de las Infraestructuras de Datos Espaciales (IDE), siendo éstas, sistemas de información integrados por un conjunto de recursos que permite el acceso y la gestión de datos y servicios geográficos disponibles en Internet, los cuales cumplen una serie normas, estándares y especificaciones que regulan y garantizan la interoperabilidad de la información geográfica. Posee como una de sus líneas de negocio el desarrollo de soluciones que facilitan el tratamiento y la visualización de datos de base geográfica, tanto en entorno Web como móvil. Son los denominados sistemas de información geográfica (SIG), así como servicios basados en localización.

GeoSLab posee una amplia experiencia nacional y europea en el ámbito de las Infraestructuras de datos espaciales, IDEs. La empresa tiene en su cartera de clientes al Instituto Geográfico Nacional⁶, la Confederación Hidrográfica del Ebro⁷ y al Ayuntamiento de Zaragoza⁸, con el que lleva colaborando desde 2004 en la implantación y evolución del su infraestructura de datos espaciales (IDEZar⁹), entre otros.

⁶ <http://www.ign.es/ign/main/index.do>

⁷ <http://www.chebro.es/>

⁸ <http://www.zaragoza.es>

⁹ <http://www.zaragoza.es/ciudad/idezar/>

1.2. Contexto tecnológico

1.2.1. Sistemas de Información Geográfica (SIG)

Los Sistemas de Información Geográfica (*Geographic Information Systems, GIS*) son un conjunto de herramientas que permiten capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. Algunos ejemplos de información geográficamente referenciada son, imágenes aéreas o de satélite de zonas de tierra, siempre y cuando se pueda determinar una localización geográfica sobre los mismos. Ya sea un plano de una ciudad, un mapa de carreteras digitalizado, datos sobre población, tipos de suelo o precipitaciones, medidas en forma de mapa que lleva asociados atributos numéricos y textuales.

Los SIG separan la información en diferentes capas temáticas y las almacenan independientemente. Permitiendo trabajar con ellas de manera rápida y sencilla, facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar nueva información que, de otra forma, no se podría obtener.

A las capas de un SIG se les conoce como coberturas geográficas y se muestran de forma superpuesta, *Figura 1*. Cada una de estas capas puede llevar asociados datos alfanuméricos sobre sus elementos. Estos datos pueden estar almacenados en una base de datos, en ficheros, etc. pero con la condición de que contengan información que permita asociarlos con el elemento geográfico de las capas sobre el que proporcionan información.

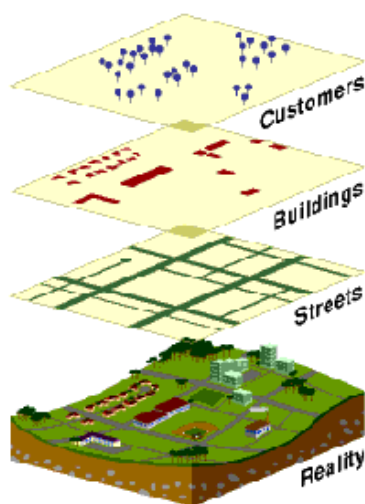


Figura 1. Estructura de capas de un SIG

Hay dos tipos fundamentales de datos dentro de los SIG que forman las capas: datos ráster y datos vectoriales. Los datos ráster son una matriz de celdas que representan un cierto espacio continuo, cuyo contenido son valores numéricos que representan el valor de alguna propiedad relacionada. En esta matriz cada elemento representa una zona geográfica (generalmente un cuadrado de algunos metros de lado). El tamaño de este cuadrado es la resolución de la cobertura ráster; a mayor resolución (cuadrado más pequeño), más nivel de detalle. Para que se pueda considerar una cobertura geográfica y usarse en un SIG, debe existir una forma de asociar a cada elemento de la cobertura ráster una posición geográfica sobre la Tierra (generalmente como un par de

coordenadas), para lo que deben pasar por un proceso de georreferenciación. Hay dos formatos que normalmente contienen coberturas ráster,

- Un fichero de imagen en formato estándar (JPEG, GIF, BMP...) con un fichero que aporte la georreferenciación, *Word file* o fichero del mundo.
- Un fichero en formato especial (GeoTIFF¹⁰, *ASCII Grid*...) que contiene tanto la malla como la georreferenciación de la misma.

En las coberturas vectoriales la información geográfica se almacena en forma de elementos geométricos (básicamente puntos, líneas y polígonos, aunque existen tipos mucho más elaborados) definidos mediante sus coordenadas numéricas en un sistema de coordenadas concreto. Las coberturas vectoriales son el formato más utilizado para asociar información alfanumérica, aunque también es posible asignarla a datos ráster. A los elementos geográficos se les asocia un identificador gracias al cual se le pueden asociar datos alfanuméricos en una o varias filas. Hay varios formatos para almacenar coberturas vectoriales, uno de los más utilizados es el formato *Shapefile*¹¹. Se trata de un formato creado por ESRI¹² que consta de:

- Un fichero con extensión SHP que contiene los elementos geográficos. Cada cobertura contiene exclusivamente un tipo de elementos geométricos, (puntos, líneas o polígonos). En este fichero no se indica el sistema de coordenadas en el que están definidos los elementos geográficos.
- Un fichero de índice con extensión SHX para agilizar el acceso no secuencial a los elementos del fichero SHP.
- Un fichero en formato DBase con extensión DBF, que posee una tabla donde se encuentran los atributos alfanuméricos de cada elemento geográfico. Cada fila de la tabla de este fichero debe corresponder con un elemento geográfico del fichero SHP (y sólo con uno).
- Puede haber otros ficheros con índices espaciales para agilizar el trabajo con el *Shapefile*, pero no son obligatorios.

Ambos tipos de datos necesitan estar georreferenciados, por lo que es necesario aplicar un proceso para asignar una localización geográfica sobre la superficie de la Tierra a cada uno de los elementos que los forman. La georreferenciación de los datos tipo ráster se hace a través de un fichero externo, en el que se indican las coordenadas de la esquina superior izquierda, el número de celdas que posee el ráster, y el tamaño de cada una de estas o si el formato de almacenamiento lo permite insertando toda esta información dentro del propio fichero. En el caso de datos vectoriales, cada uno de los elementos de la cobertura viene descrito por unas coordenadas respecto a un sistema de referencia conocido por el SIG.

En los últimos años, una de las tendencias en los SIG ha sido la de ser cada vez más accesibles a través de Internet. Una de las respuestas a esta tendencia es la especificación de servicios de mapas por Internet propuesta por el consorcio *Open Geospatial Consortium*¹³ (OGC). Éste fue creado en 1994 con el fin de definir estándares abiertos e interoperables dentro de los Sistemas de Información

¹⁰ <https://en.wikipedia.org/wiki/GeoTIFF>

¹¹ <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/shapefiles/what-is-a-shapefile.htm>

¹² <http://www.esri.com/>

¹³ <http://www.opengeospatial.org/>

Geográfica, mediante la persecución de acuerdos entre las empresas del sector que posibiliten la interoperación de sus sistemas de geoprocesamiento y faciliten el intercambio de información geográfica.

El estándar *Web Map Service*¹⁴ (WMS) definido por OGC permite generar mapas dinámicamente a partir de datos espaciales georreferenciados. Según este estándar, un ‘mapa’ es una representación de la información geográfica en forma de un archivo de imagen digital para ser presentada en la pantalla de un ordenador. Los WMS ofrecen no sólo un conjunto de capas para formar los mapas, sino también un conjunto de estilos de visualización asociados a las capas, que facilitan la tarea de superponer los mapas generados con otros. El estándar establece que siempre debe existir un estilo por defecto asociado a las capas ofrecidas, pero permite que se definan otros estilos para la misma capa (por ejemplo, un estilo semitransparente o transparente para capas con polígonos, o diferentes estilos para capas que representan distintos niveles de carreteras, etc.). Estas capas y estilos se definen en el fichero de *Capabilities* del servicio.

El estándar *Web Map Tile Service*¹⁵ (WMTS) de OGC proporciona un enfoque complementario al WMS. A diferencia del WMS que fue concebido para poder compartir mapas personalizados y se adoptó como una solución ideal para mostrar datos dinámicos, el WMTS renuncia a la personalización de estos mapas para obtener una mayor escalabilidad. Esto se logra sirviéndose de datos prerrenderizados donde la extensión geográfica a mostrar y las escalas, han sido restringidas a un conjunto discreto de teselas que siguen una geometría de malla regular.

Un servicio WMTS permite almacenar los datos previamente obtenidos agilizando la carga de los mismos en caso de que estos vuelvan a ser solicitados (caché). Este servicio usa un modelo de teselas parametrizado de tal manera que un cliente puede hacer peticiones de un conjunto discreto de éstas y recibir rápidamente del servidor fragmentos de imágenes prerrenderizadas (*Tiles* o teselas), que generalmente ya no requieren de ninguna manipulación posterior para ser mostrados en pantalla.

Cada una de las capas de un servidor WMTS sigue una o diversas estructuras piramidales de escalas (*Tile Matrix Sets* o conjunto de Matrices de Teselas), en la que cada escala o nivel de la pirámide (*Tile Matrix* o Matriz de Teselas), es una rasterización y fragmentación regular de los datos geográficos a una escala o tamaño de píxel concreto, *Figura 2*. Por ello, una capa puede estar disponible en varios sistemas de coordenadas, y tener diferente ámbito en función de éstos.

¹⁴ <http://www.opengeospatial.org/standards/wms>

¹⁵ <http://www.opengeospatial.org/standards/wmts>

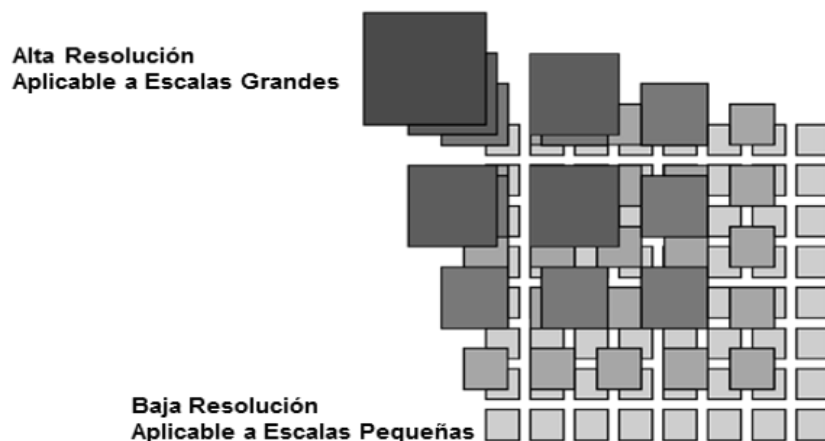


Figura 2. Detalle la estructura de datos de una TileCache

1.2.2. IDEZar

IDEZar¹⁶ es la Infraestructura de Datos Espaciales de Zaragoza, que tiene entre sus objetivos principales facilitar a la ciudadanía el acceso a la información de la ciudad localizándola sobre el mapa. Permitiendo así, ofrecer servicios y aplicaciones que proporcionan un valor añadido a la gran cantidad de contenidos que el Ayuntamiento de Zaragoza ofrece a la ciudadanía.

La iniciativa IDEZar nace en el año 2004 con la finalidad de llevar a cabo la implantación de una Infraestructura de Datos Espaciales a nivel local. El trabajo realizado durante estos años ha permitido que hoy en día sean múltiples y variadas las funcionalidades y los datos que el Ayuntamiento ofrece a la ciudadanía mediante esta infraestructura. Éstas van desde el nuevo servicio "Zaragoza al instante" de la página principal de la Web de la ciudad, donde se ofrece una visión dinámica y en tiempo real de la ciudad sobre el mapa, pasando por el servicio de Callejero¹⁷ (uno de los primeros ofrecidos por IDEZar). Forman parte también de estas prestaciones, servicios vinculados a la movilidad que permiten la planificación de rutas multimodales en transporte público.

Gracias a la cercanía de la infraestructura a la ciudadanía, IDEZar fue premiada en la categoría de usabilidad en el concurso "EUROGI/eSDI-Net Awards 2011" promovido por la organización EUROGI (*Umbrella Organization for Geographic Information*), cuyo objetivo es reconocer y poner en valor las buenas prácticas en Infraestructuras de Datos Espaciales.

¹⁶ <http://idezar.zaragoza.es/>

¹⁷ <http://idezar.zaragoza.es/callejero/>

1.2.3. Energic OD

El proyecto ENERGIC OD¹⁸ (*European NETwork for Redistributing Geospatial Information to user Communities - Open Data*) afronta la búsqueda de soluciones al problema de la falta de acuerdos que hagan compatible y accesible la información geográfica que está en Internet, lo que representa una barrera para la explotación potencial de estos tipos de información, que son creados y ofrecidos por un conjunto muy heterogéneo de sectores de la sociedad.

El mundo de la Información Geográfica (IG) es, en la actualidad, extremadamente heterogéneo. Los requisitos del usuario y del sistema, suelen ser demasiado variados para ser satisfechos por un solo sistema o tecnología. Esto ha conducido a una total falta de acuerdo sobre los estándares de interoperabilidad, creando una serie de problemas que impiden explotar al máximo las características de la IG. En este enfoque, cada componente específico tiene que realizar todas las acciones de interoperabilidad necesarias para la interconexión con cada sistema.

En sus líneas de trabajo incluyen la creación de distribuidores virtuales de datos (*Virtual Hubs*), diseñados y puestos en marcha en otros proyectos recientes de la Unión Europea. Éstos actuarán como “*brokers*” capaces de realizar todas las labores de interoperabilidad necesarias para conectar sistemas de información heterogéneos. Estas aplicaciones afrontarán las necesidades de empresas, ciudadanos e instituciones públicas, haciendo uso de datos abiertos geográficos tanto públicos como privados, para mejorar la explotación de los mismos y para desarrollar nuevos mercados potenciales.

El proyecto está parcialmente financiado por el Programa ICT PSP como parte del Programa Marco para la Competitividad y la Innovación de la Comunidad Europea, con una aportación de 2,63 millones de euros, de un presupuesto total de 5,36 millones.

Este enfoque innovador facilitará en gran medida el desarrollo de aplicaciones nuevas y multidisciplinarias basadas en la explotación plena de las IGs (abiertas), estimulando así la innovación y las actividades empresariales. Para construir *Virtual Hubs* compatibles con INSPIRE¹⁹, se utilizarán e integrarán varias tecnologías de software existentes actualmente, como la plataforma *Atl@nte*, *SpatiumCube*, *CatMDEdit*, *GeoServer*, *Deegree*, *PostgreSQL*, *EasySDI*, *Sync'Serv* y *GeoExt*.

1.3. Motivación

A día de hoy son muchos los datos de carácter geográfico cuyo acceso no es posible. Gran cantidad de mapas y planos históricos no pueden ser accedidos de manera sencilla por no haber sido digitalizados, o si se ha hecho, por haberlo hecho pensando en preservar ese documento y no en facilitar su distribución y uso.

Además, actualmente el uso de la información georreferenciada en Internet, tanto mapas como servicios basados en la ubicación, se ha extendido ampliamente. Esto se ha debido gracias al trabajo de entidades como el consorcio *Open Geospatial Consortium (OGC)*. Los servicios web OGC son un conjunto de tecnologías que facilitan la disponibilidad y el acceso a la información geoespacial

¹⁸ <http://www.energic-od.eu/>

¹⁹ <http://inspire.ec.europa.eu/>

usando un conjunto de estándares y especificaciones definidas por OGC. Esto permite que las aplicaciones operen bajo condiciones conocidas garantizándose la interoperabilidad entre sistemas. La interoperabilidad es la capacidad para que diversos sistemas puedan comunicarse, ejecutar procesos o transferir datos entre ellos, sin necesidad que el usuario tenga ningún conocimiento específico, sino más bien genérico, del sistema.

Las principales motivaciones para la realización de este proyecto son:

- Definir e implementar un proceso para que datos de carácter geográfico que no cumplen los estándares geográficos para su publicación y explotación en Internet, los cumplan para su publicación posterior.
- Experimentar con el *Virtual Hub* en el contexto del proyecto ENERGIC OD para obtener conclusiones sobre su validez/idoneidad para su uso en Sistemas de Información Geográfica.
- Ofrecer una nueva herramienta que forme parte de la actual infraestructura de datos espaciales del Ayuntamiento de Zaragoza y que permita a la ciudadanía descubrir la evolución de la ciudad a lo largo de su historia a través de su cartografía, poniendo en valor el catálogo de mapas históricos del Archivo municipal de la ciudad.

1.4. Objetivos

El objetivo principal de este Proyecto Final de Carrera es integrar datos geográficos no concebidos con el fin de ser publicados en los actuales sistemas de información geográfica. Concretamente planos históricos de la ciudad de Zaragoza creados en los últimos 300 años tras escanearlos a muy alta resolución por parte del Ayuntamiento de Zaragoza, en servicios de publicación de datos geográficos estandarizados por el *Open Geospatial Consortium* (OGC²⁰). Posteriormente serán accesibles a través de visores geográficos que cuenten con las funcionalidades habituales en este tipo de aplicaciones Web, a la vez que incorporan algunas nuevas destinadas a dar relevancia a las condiciones de documentos históricos.

Las tareas a realizar serán:

- Tratamiento y procesado de datos. Los datos de partida no cumplen los estándares geográficos al uso. Su orientación y escala no coincide con ningún sistema compatible. Por ello, el primer trabajo será georreferenciar y corregir cada uno de los planos al objeto de que puedan ser superpuestos sobre el plano actual de la ciudad, de tal forma que exista una completa coincidencia entre puntos de referencia de ambos. Este proceso de transformación supone la incorporación de información adicional a cada plano que puede incrementar notablemente su tamaño de almacenamiento, que en algún caso supera 1GB. Es un proceso que debe ser realizado manualmente y con herramientas muy especializadas.
- Publicación de los datos. Instalación, configuración y puesta en marcha de un servidor especializado en la publicación de datos geográficos. En este proceso hay que efectuar un diseño de los estilos que se seguirán para la presentación de la información. En este paso hay

²⁰ <http://www.opengeospatial.org/>

que tomar en consideración que unos planos se van a poder superponer a los otros (necesidad de definir transparencias) lo que requiere mantener una compatibilidad de estilos para que no se mezclen. A su vez, todos los nuevos planos se podrán superponer sobre los ya publicados por la Infraestructura de Datos Espaciales del Ayuntamiento de Zaragoza, lo que requiere mantener a su vez la compatibilidad con ellos. Finalmente, es necesario optimizar su publicación ya que el montante total de información que se va a publicar ronda los 60GB.

- Desarrollo de la aplicación Web de visualización. Partiendo de tecnología de visualización con la que ya se cuenta en la empresa, se va a desarrollar una aplicación de visualización orientada a satisfacer los requerimientos de dos escenarios. En el primero, se integrará la aplicación en la infraestructura de datos espaciales del Ayuntamiento de Zaragoza. En el segundo, se integrará con uno de los *Virtual Hubs* del proyecto europeo ENERGIC OD.
- Pruebas del sistema. Las dos aplicaciones Web van a ser usadas por el público en general y es necesario efectuar un gran número de pruebas que aseguren que todos los nuevos planos han sido correctamente integrados y se visualizan perfectamente a todos los niveles de zoom.
- Integración y despliegue del sistema. En ambos escenarios se llevarán a cabo todas las tareas necesarias para integrar y poner en marcha todos los componentes usados y desarrollados asegurando su correcto funcionamiento dentro de las infraestructuras ya existentes.

1.5. Metodología y actividades

El proyecto se ha desarrollado en la empresa GeoSLab, empresa que tiene una larga experiencia con proyectos de gran envergadura. Se han seguido algunas de sus metodologías para la gestión del proyecto. GeoSLab posee su propio Sistema Integrado de Gestión, que se encuentra certificado en base a las normas UNE-EN-ISO 9001 “Sistemas de Gestión de la Calidad”²¹, UNE-EN-ISO 14001:2004²² “Sistemas de gestión medioambiental” y SPICE (ISO/IEC 15504-5)²³. SPICE es una norma abierta e internacional para evaluar y mejorar la capacidad y madurez de los procesos, que se aplica a la evaluación y mejora de la calidad del proceso de desarrollo y mantenimiento de software.

Para la realización de este proyecto se ha utilizado el ciclo de vida de desarrollo iterativo incremental. Esta metodología se basa en la construcción de procesos, componentes y/o desarrollo de secciones de software reducidas que son probadas y van aumentando su tamaño con el tiempo. De esta manera se facilita la detección de errores, además de posibilitar la modificación de algunas especificaciones por parte del cliente, permitiendo un desarrollo flexible al cambio.

Durante todo el proyecto se ha llevado un control de versiones y copias de seguridad de los elementos generados.

²¹ http://es.wikipedia.org/wiki/ISO_9001

²² <http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0032367#.UUBpaDfNS3U>

²³ http://es.wikipedia.org/wiki/ISO/IEC_15504

Antes de comenzar con el desarrollo del sistema, se realizó un periodo de formación en tecnologías y estándares sobre publicación de información geográfica y herramientas para el manejo de información ráster, debido al desconocimiento previo. Además de un proceso en el que se estudiaron las diferentes alternativas existentes para la georreferenciación y el tratamiento de la información ráster y su posterior publicación. Debido al desconocimiento previo sobre todos estos campos, hay que hacer especial mención a la gran cantidad de tiempo invertido en la familiarización con los distintos formatos en los que se almacena la información, herramientas para su tratamiento y servicios para su publicación.

Después de obtener los conocimientos necesarios para poder comprender el funcionamiento de los distintos elementos del sistema, se comenzó a trabajar en la especificación de requisitos que debe poseer él mismo y en concreto cada una de las herramientas y/o componentes que van a intervenir en cada uno de los procesos. Una vez establecidos se propusieron los primeros diseños de cada uno de los elementos del sistema hasta que se consiguieron los diseños que se creyó cumplían con todos los requisitos.

Por último, se establecieron las fases que deberían ir resolviéndose consecutivamente para el correcto desarrollo del proyecto, ya que cada uno dependía de la correcta consecución del anterior:

- Análisis y diseño
 - Estudio del estado del arte.
 - Definición de requisitos del sistema.
 - Diseño de la arquitectura del sistema.
 - Formación en las distintas tecnologías a usar en el proyecto.
 - Diseño de la aplicación web de visualización.
- Desarrollo
 - Tratamiento y procesado de los datos de partida para conseguir su publicación en servicios de información geográfica.
 - Configuración, carga de los datos y puesta en marcha de ambos servicios de publicación de información geográfica.
 - Desarrollo de la aplicación web de visualización de la información ofrecida por los servicios anteriores.
- Diseño y ejecución del plan de pruebas.
- Integración y despliegue.

Se puede ver la planificación del proyecto plasmada mediante un diagrama de Gantt, *Figura 3*.

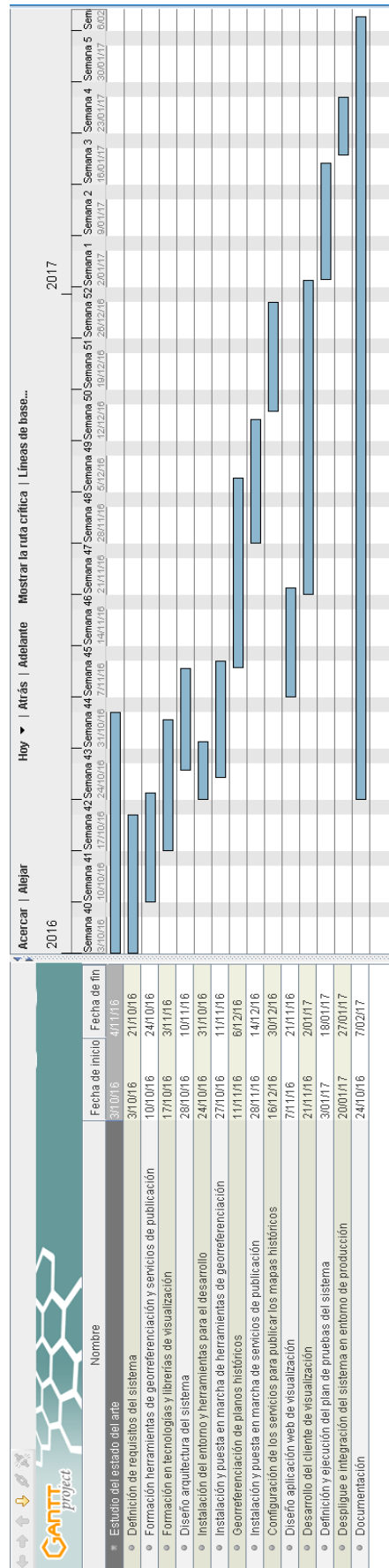


Figura 3. Cronograma de planificación del proyecto

1.6. Estructura del documento

La estructura del documento se organiza en tres capítulos principales complementados por siete anexos en los que se complementa la información dada en los capítulos principales. En estos capítulos se describen de forma clara y ordenada las actividades llevadas para alcanzar los objetivos propuesto en este proyecto.

- El capítulo **Introducción**, describe el contexto, tanto tecnológico como profesional, las motivaciones para la realización de este PFC, y los objetivos que en él se abordan.
- El capítulo **Trabajo realizado**, se explican de forma concreta y concisa las tareas que se han llevado a cabo durante el proyecto. En primer lugar, se presentará la arquitectura del sistema. Posteriormente, el trabajo realizado en el tratamiento, publicación y visualización de los mapas históricos. Por último, las pruebas realizadas y las tareas de despliegue del sistema.
- El capítulo **Conclusiones**, muestra las conclusiones obtenidas del desarrollo de las actividades realizadas en la ejecución del trabajo. Además, se describen de forma breve los trabajos futuros que pueden llevarse a cabo para complementar este proyecto. También se explican los problemas y dificultades que han surgido durante el desarrollo del trabajo, finalizando con una valoración personal del mismo.
- En la última parte del documento se encuentran los **Anexos**, que describen de una forma más detallada el trabajo realizado en la realización de este PFC.

2. Trabajo realizado

En este capítulo se va explicar de forma concreta y concisa las tareas que se han llevado a cabo durante el proyecto. Primero, se introducirá la arquitectura del sistema. Después, se mostrará el trabajo realizado dividido en cuatro bloques. Un primer bloque de tratamiento y procesamiento de datos, un segundo bloque de publicación de información geográfica, un tercer bloque de desarrollo de la herramienta web de visualización y un cuarto bloque en el que se muestra la ejecución del plan de prueba e integración y despliegue en las infraestructuras existentes.

2.1. Arquitectura del sistema

En este apartado se explica la arquitectura de alto nivel del sistema. Para su definición, se tuvieron en cuenta las distintas necesidades y requisitos que planteaba cada uno de los dos casos de uso del sistema a desarrollar.

Por un lado, el escenario existente en IDEZar, necesitaba un sistema robusto y probado que no fuese proclive a fallos y eficiente a la hora de responder a una gran carga de peticiones por parte de los usuarios. Además, en IDEZar ya existía una infraestructura previa en la que debía integrarse la solución desarrollada, y este fue un importante factor a tener en cuenta a la hora de plantear la arquitectura del sistema. Por otro lado, hubo que tener en cuenta el requisito principal del proyecto ENERGIC OD, consistente en la experimentación de la integración en un sistema de un *Virtual Hub*, núcleo del proyecto, como elemento de publicación de los mapas históricos

Estos dos escenarios tan distintos entre sí con objetivos tan opuestos, ENERGIC OD la experimentación de un componente en desarrollo con una propuesta innovadora e IDEZar crear un sistema que deberá integrarse en una infraestructura usada por miles de usuarios, planteo un problema. *Virtual Hub* era un componente no finalista e inestable, estando en desarrollo continuo, incluyendo mejoras y subsanando errores. Esto provocaba que algunas de las características básicas y necesarias para IDEZar, aún no estuviesen disponibles en el momento de usarlas y por tanto no fuese un sistema estable. Además, su propia concepción de diseño, ser un elemento homogeneizador de muy variadas fuentes a distintas interfaces estándares, era indicativa de que posiblemente no fuese adecuado para atender las necesidades de IDEZar. Por último, la incertidumbre de la evolución y mantenimiento de la tecnología *Virtual Hub* según las conclusiones que se pudieran obtener en el proyecto. Todo esto nos llevó a tomar la decisión que *Virtual Hub* no era adecuado para IDEZar y necesitábamos buscar una solución robusta y eficiente.

Por todo esto se decidió configurar dos servicios de publicación de información geográfica distintos en paralelo. Para el caso del proyecto de ENERGIC OD se utilizó el *Virtual Hub* impuesto por el proyecto, mientras que para el caso de IDEZar se decidió utilizar *Geoserver*.

En la *Figura 4*, se puede observar la infraestructura de IDEZar en la que debía ser integrada la solución propuesta. Esta infraestructura mantenida por el Ayuntamiento de Zaragoza está compuesta por 4 servidores y configurada con una política de balanceo de carga que permite redirigir las peticiones según la carga del sistema para optimizar su rendimiento. Un servidor hace de punto de entrada común y es el encargado de llevar a cabo la política de balanceo y los otros tres serán los

encargados de responder a las peticiones y realizar la integración de nuestra solución. Las características de estos servidores son las siguientes:

- Intel Xeon 3 GHz
- 4-6 GB RAM
- Disco duro de 250 GB
- Windows Server 2003

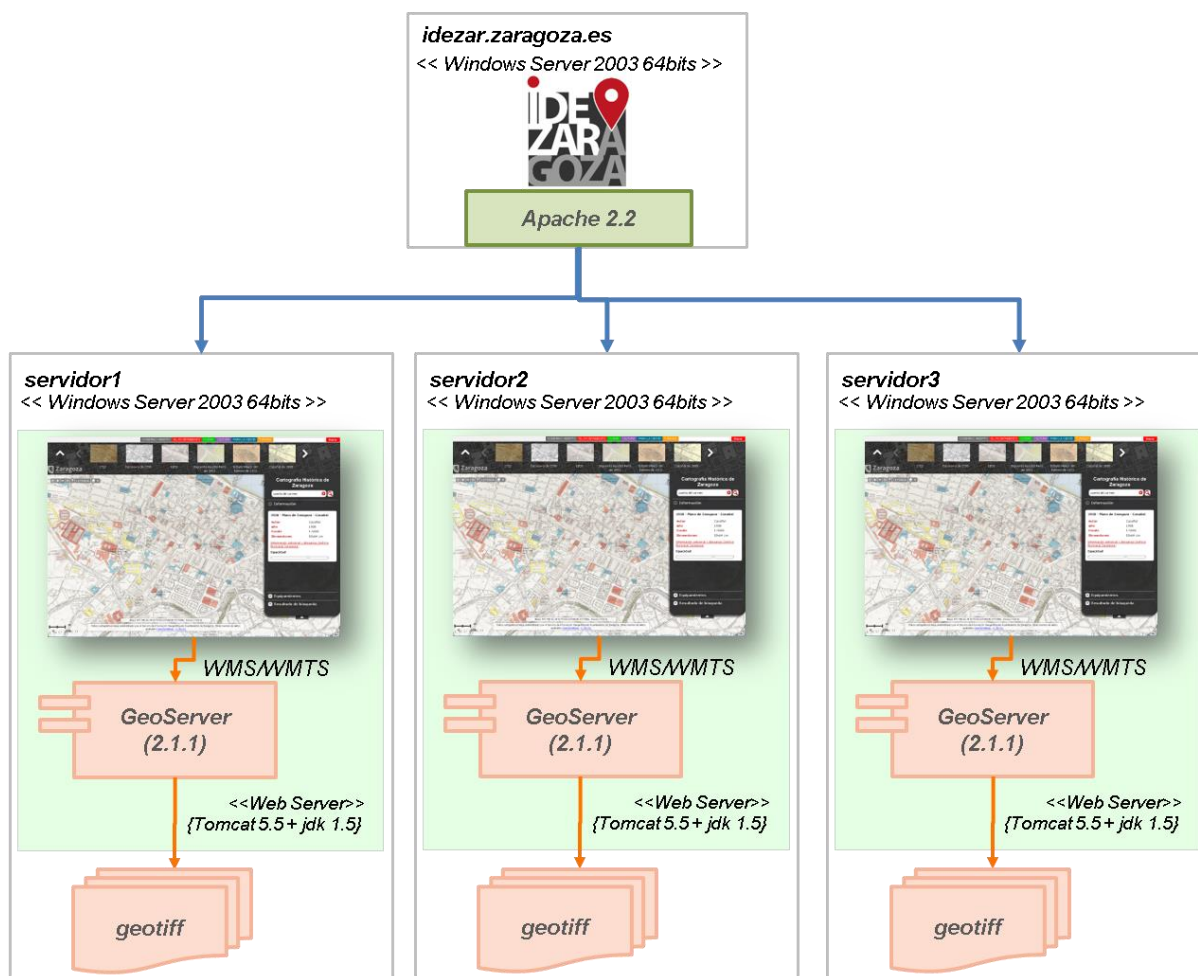


Figura 4. Arquitectura del sistema propuesta para integrar con IDEZar.

En la *Figura 5* se puede observar el detalle de la arquitectura del sistema en cada uno de los tres servidores encargados de atender las peticiones recibidas. Se observa como se ha integrado Geoserver al mismo nivel que otros servicios ya existentes dentro del variado ecosistema de la infraestructura de IDEZar, en la que ya se contaba con un servicio WMTS propio para servir la capa base del visualizador, un catálogo SRW para la resolución de las búsquedas por nombre de calle y un servicio API REST para poder mostrar los distintos tipos de equipamiento disponibles para la ciudadanía.

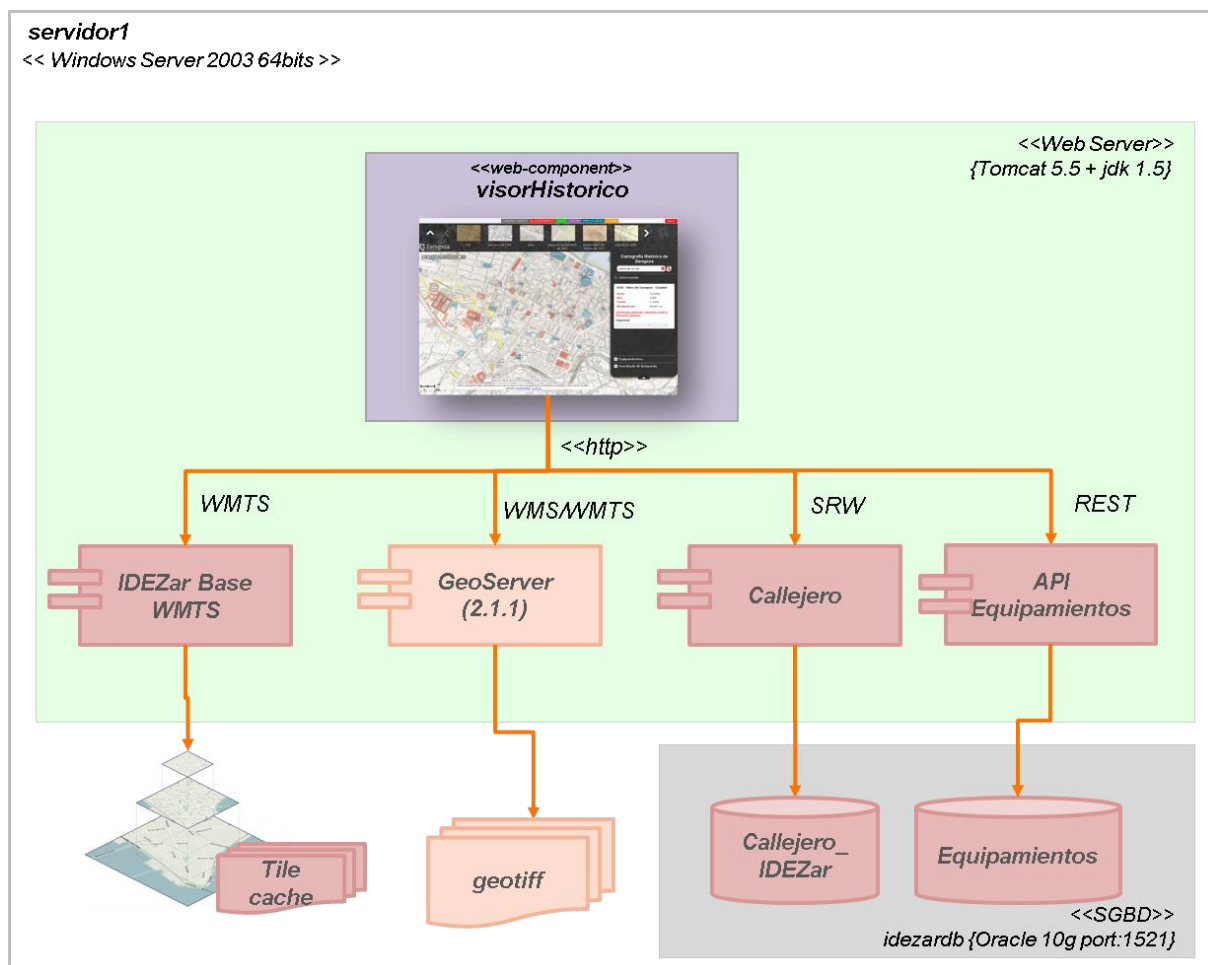


Figura 5. Arquitectura en detalle de la solución propuesta para IDEZar

Por último, en la Figura 6 se puede observar el detalle de la solución propuesta para el caso del proyecto ENERGIC OD, que es exactamente igual que la vista anteriormente pero sustituyendo Geoserver por el Virtual Hub impuesto como requisito por el proyecto ENERGIC OD.

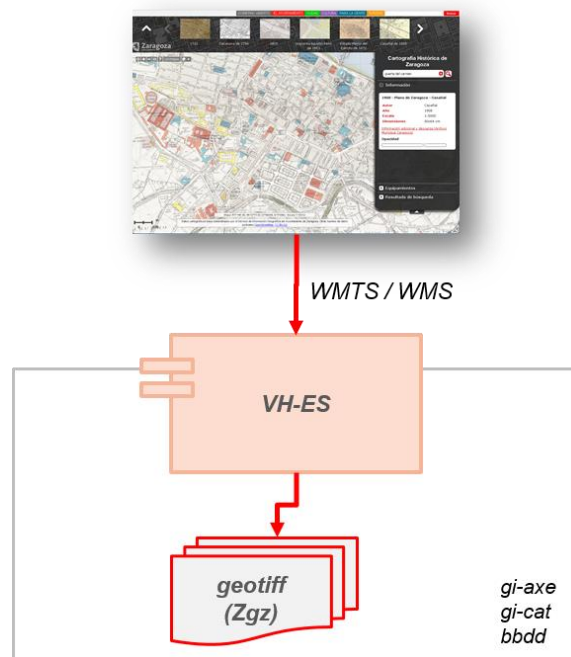


Figura 6. Detalle de la arquitectura para el caso ENERGIC OD

Durante el transcurso del proyecto se comprobó que esta decisión de diseño fue totalmente acertada, aunque en principio pareciese no ser la más óptima, ya que aparecieron múltiples inconvenientes de configuración y rendimiento del *Virtual Hub* que hicieron imposible su uso en IDEZar.

2.2. Tratamiento y procesamiento de los mapas históricos

2.2.1. Estudio de herramientas

Antes de comenzar con el proyecto, se realizó un estudio del estado del arte de las herramientas y librerías existentes hoy en día relacionados con el tratamiento y procesamiento de información ráster. Este estudio permitió aportar nuevos puntos de vista al proyecto, así como nuevas ideas no planteadas en un inicio. En el **Anexo I** se puede ver el estudio realizado en detalle.

Actualmente existe software SIG en el mercado que permiten realizar las distintas operaciones necesarias para conseguir enriquecer con información geográfica los mapas históricos digitalizados que ahora no dejan de ser una mera imagen. Algunos de ellos han sido creados por la comunidad de usuarios y distribuidos bajo una licencia GNU. Otros han sido desarrollados con un carácter comercial, y distribuidos con licencia de pago.

En un principio se consideró realizar un desarrollo propio apoyado en alguna de las librerías de tratamiento de información geográfica existentes, pero se desistió debido a la gran cantidad de tiempo que conllevaría a pesar de poder tener un mayor control sobre el proceso final. Por ello se optó por buscar una herramienta de escritorio SIG entre todas las que hay y comprobar si alguna cumplía con nuestras necesidades.

Tras realizar el estudio del arte se comprobó que había múltiples herramientas en el mercado (*ArcGIS*²⁴, *QGIS*²⁵, *gvSIG*²⁶, *uDIG*²⁷, *OpenJUMP*²⁸, *GDAL/OGR*²⁹, *GRASS*³⁰) que en principio podrían satisfacer nuestros requerimientos, por lo que las necesidades respecto a este apartado estaban cubiertas, restando sólo seleccionar la mejor de entre todas las existentes.

2.2.2. Elección de la herramienta

A partir de los requisitos del sistema enumerados en el **Anexo II.I** se han extraído características que debería tener la herramienta que seleccionásemos:

1. Capacidad para interactuar con distintos formatos de información ráster, en especial el formato GeoTIFF.
2. Manejo de ficheros de datos ráster de gran tamaño.
3. Permitir la corrección y georreferenciación de datos ráster sin información geográfica.
4. Capacidad de exportación a formatos de datos ráster abiertos y compatibles con los servicios de publicación OGC.
5. Software no propietario.
6. Facilidad de uso
7. Posibilidad de funcionar en distintos entornos

El paso siguiente fue realizar unas pruebas previas de uso en cada herramienta que poseía las características enumeradas anteriormente con un mapa histórico sencillo para familiarizarse con ella. Tras estas pruebas de contacto se seleccionaron cuatro herramientas.

Como herramienta de escritorio quedaron como candidatas *gvSIG* y *QGIS*. Ambas dos son Sistemas de Información Geográfica de código libre multiplataforma, *gvSIG* basada en Java y *QGIS* desarrollada en C++, son gratuitas, con licencia GNU/GPL y apoyo de la comunidad.

Tras probar ambas herramientas se decidió usar *QGIS* por las siguientes razones:

- Interfaz amigable y funcional. Esta es una razón fundamental, ya que facilita el aprendizaje de uso de la herramienta y optimiza el tiempo de trabajo. *QGIS* posee una interfaz mucho mejor y más intuitivo que *gvSIG*.

²⁴ <https://www.arcgis.com/>

²⁵ <http://www.qgis.org/es/site/>

²⁶ <http://www.gvsig.com/es>

²⁷ <http://udig.refractory.net/>

²⁸ <http://openjump.org/>

²⁹ <http://www.gdal.org/>

³⁰ <https://grass.osgeo.org/>

- Estabilidad. En las pruebas realizadas *QGIS* se mostró más estable que *gvSIG*, que sufrió problemas que interfirieron en su uso.
- Apoyo de la comunidad. Actualmente *QGIS* sigue sacando versiones regularmente y posee una comunidad de usuarios y desarrolladores activa. *gvSIG* no se encuentra en la misma situación actualmente.
- Componentes integrados. *QGIS* integra actualmente, librerías como GRASS y GDAL para ofrecer características avanzadas en el tratamiento de información geográfica, y en especial de datos ráster. *gvSIG* se apoya en *Geotools*, una librería también muy importante en el mundo SIG, pero más orientada a la información geográfica vectorial.

Por todos estos motivos *QGIS* fue elegida como herramienta de escritorio para el tratamiento y procesamiento de datos. Además, abre sus puertas a todos aquellos que quieran colaborar en su desarrollo, por lo que si fuese necesario se podrían llevar a cabo modificaciones oportunas para adaptarla a nuestras necesidades.

Durante esta fase también se encontraron herramientas para ser usadas desde la línea de comandos. Estas herramientas no disponen de una interfaz gráfica que facilite su uso, pero como contrapartida se tiene acceso a toda la potencia y opciones que ofrecen cada uno de sus comandos, una vez se paga el precio del proceso de aprendizaje.

Las dos herramientas seleccionadas fueron GRASS y GDAL/OGR. Tras probar ambas herramientas como en el caso anterior, se llegó a la conclusión de que ambas son muy similares, tenían amplia documentación de cada uno de los comandos que ofrecen, y cumplían con creces las características básicas que se necesitaban. Se decidió quedarse con GDAL porque su instalación y comienzo de uso resultaban menos dificultosos que GRASS y permitía que sus comandos interactuaran directamente con los ficheros que contienen los datos, mientras que en GRASS se necesita realizar una carga previa para que cada formato distinto sea transformado al modelo común que utiliza.

2.2.3. Tratamiento y procesamiento de los mapas históricos

Actualmente el Ayuntamiento de Zaragoza está ofreciendo como datos abiertos muchos documentos históricos relacionados con la ciudad que han sido digitalizados, entre los que hay varios mapas históricos. Aunque los mapas históricos digitalizados están disponibles como datos abiertos, el formato en el que se ofrecen estos mapas limita su reutilización y explotación, ya que son archivos ráster en formato *DjVu*. Se trata de un formato de archivo abierto diseñado principalmente para almacenar documentos escaneados, muy adecuado para archivar y preservar a largo plazo documentos digitalizados, pero no está muy extendido y no permite que su contenido (mapas) tenga asociada información de georreferenciación. De entre todos estos mapas se hizo una selección de 20 que fueron redigitalizados en un formato más común (TIFF³¹) y con una mayor resolución, lo que conlleva un aumento del tamaño de cada uno de ellos. Estos mapas no se pueden utilizar directamente en la aplicación, ya que deben ser georreferenciados primero. La georreferenciación es

³¹ <https://en.wikipedia.org/wiki/TIFF>

el proceso de asignar coordenadas del mundo real a cada píxel del archivo ráster. Este proceso se ha realizado con software de código abierto *QGIS*.

Para la georreferenciación, se deben seleccionar varios puntos de la imagen y a cada uno de ellos se le debe asignar sus coordenadas en el mundo real, que dará lugar a los llamados punto de control. Para encontrar las coordenadas en el mundo real de estos puntos seleccionados en el mapa histórico, habrá que servirse de algún tipo de dato con información espacial asociada, donde se puedan identificar sin lugar a dudas y con gran precisión los puntos seleccionados en el mapa histórico. Se trata pues de un proceso manual y la calidad de la georreferenciación final depende en gran medida de la precisión con la que se realiza la creación de los puntos de control. Esta acción debe realizarse como al menos tres veces, ya que este es el número mínimo de puntos de control que necesitan los algoritmos que posteriormente vamos a aplicar. Un mayor número de puntos de control repartidos homogéneamente por todo el mapa a tratar darán como resultado una georreferenciación de mayor calidad.

En la mayoría de los mapas, los puntos seleccionados son puntos que no han cambiado a lo largo de los años, como, por ejemplo, en la ciudad de Zaragoza las esquinas de la Basílica del Pilar o el Palacio de la Aljafería, *Figura 7*.



Figura 7. Ejemplo de toma de puntos de control

Tras la toma de los puntos de control será necesario seleccionar distintas opciones para llevar a cabo el proceso de georreferenciación y que influyen directamente en la calidad del resultado y el tiempo de procesamiento.

- **Tipo de transformación:** Es el algoritmo que se usará para realizar las transformaciones necesarias que impliquen la georreferenciación del mapa histórico.
 - *Linear algorithm.*
 - *Helmert transformation*
 - *Polynomial algorithms 1-3*
 - *Thin plate spline (TPS)*
 - *Projective transformation*
- **Método de muestreo:** es el método aplicado para conseguir rellenar los huecos que pueden surgir durante el proceso en la imagen inicial mediante datos “inventados”.
 - Interpolación vecino más cercano.
 - Interpolación lineal
 - Interpolación cúbica
 - Interpolación *S-Spline*
 - Interpolación Lanczos

Tras documentarse sobre las características de cada método de transformación y muestreo se seleccionaron la transformación ‘*Thin plate spline*’ y el método de muestreo bilineal por ser las mejores en relación tiempo de computación/resultado.

Para almacenar el resultado de la georreferenciación del mapa histórico se consideró utilizar formatos de almacenamiento eficientes en acceso y de gran compresión como ECW³² o JPG2000³³. Pero la necesidad de superponer y comparar los planos históricos con una base cartográfica actual impidió su uso, teniendo que optar por el formato GeoTIFF, el cual permite transparencia, pero no tiene las ventajas de los anteriores.

Una vez definidas las acciones necesarias para georreferenciar un mapa con éxito y elegido el formato para almacenarlo, se comenzó con la georreferenciación de los demás. Durante el transcurso de este proceso surgieron dos inconvenientes que se pudieron solventar con éxito:

- En primer lugar, se trataron los mapas de menor tamaño, los cuales fueron tratados desde el propio QGIS sin ningún tipo de problemática. Al comenzar a tratar los mapas de mayor tamaño surgió un problema con la escritura en disco del resultado de la operación en formato GeoTIFF. Tras investigar sobre el problema se descubrió que radicaba en que QGIS internamente utiliza las GDAL para realizar esta operación y no incluía en la ejecución de un comando el parámetro específico necesario para poder escribir ficheros GeoTIFF de gran tamaño.

³² [https://en.wikipedia.org/wiki/ECW_\(file_format\)](https://en.wikipedia.org/wiki/ECW_(file_format))

³³ <https://jpeg.org/jpeg2000/>

Debido a este problema nos vimos obligados a dividir el proceso en dos pasos. El primero de ellos es el expuesto anteriormente, pero en lugar de lanzar la ejecución de la transformación desde el mismo QGIS lo que se hace es exportar los comandos GDAL que QGIS compone a través de la información recogida en su interfaz y posteriormente ejecuta. En segundo lugar, se ejecutan estos comandos desde la línea de comandos sobre una instancia de GDAL instalada previamente, añadiendo el parámetro que permite la escritura de ficheros GeoTIFF de gran tamaño de manera manual.

- Aplicando el proceso definido para el tratamiento en dos o tres mapas éste fallaba sin dar ningún tipo de información de la causa del error. La única opción fue utilizar una parametrización del proceso distinta, eligiendo un tipo de transformación menos compleja. En estos casos se optó por utilizar los métodos de transformación polinomial de grado 3 y 2, finalizando la georreferenciación con éxito y un resultado de calidad.

Todo este proceso, la carga de un dato fuente de referencia para encontrar los puntos de control y la instalación de los distintos componentes, se documenta ampliamente en el **Anexo III**.

2.3. Servicios de publicación de los mapas históricos georreferenciados.

2.3.1. Estudio de servicios de publicación

A pesar de que uno de los requisitos de ENEGIC OD nos imponía un servicio de publicación a usar, se tomó la decisión de utilizar otro servicio de publicación de información de probada eficiencia y robustez. Esto se hizo así ya que el servicio impuesto por el proyecto ENEGIC OD todavía estaba en desarrollo y no se podía garantizar cuales serían sus características finales.

Respecto a los servicios de publicación de información geográfica, se siguió el mismo procedimiento que en el caso de la herramienta SIG. Se realizó un estudio del estado del arte actual que nos permitiese conocer la situación respecto a los componentes existentes hoy en día relacionados con los servicios de publicación de información geográfica. En el **Anexo I** se puede ver el estudio realizado con más detalle.

Se comprobó que había herramientas en el mercado (*GeoServer*, *MapServer*, *Deegree*) pero su número era bastante inferior al de herramientas de tratamiento de datos disponibles. Aun así, las existentes parecían ser capaces de cubrir nuestras necesidades respecto a este apartado. Sería necesario seleccionar la mejor de entre todas ellas para nuestro propósito.

2.3.2. Elección del servicio de publicación

A partir de los requisitos del sistema enumerados en el **Anexo II.I** se han extraído características que debería tener el servicio de publicación que seleccionásemos :

1. Acceso a los formatos ráster generados en el proceso anterior.
2. Cumplir con los estándares establecidos por OGC para los servicios de publicación de datos geográficos.
3. Software no propietario.
4. Eficiencia en la respuesta a las peticiones recibidas.
5. Facilidad de uso e interfaz amigable.
6. Posibilidad de funcionar en distintos entornos.

Al contrario que en el caso de las herramientas para el tratamiento de datos, el número de servicios de publicación encontrados era mucho menor. Las diferencias entre ellos en cambio eran bastantes mayores. Tras estudiar las características de cada uno de ellos para ver que cumplían las enumeradas anteriormente, comprobar el soporte de sus desarrolladores y el apoyo de la comunidad que recibían, *GeoServer* sobresalía por encima de *MapServer* y *Deegree*. Posee una interfaz muy cuidada e intuitiva para la publicación y configuración del servicio, carga una gran variedad de formatos (tanto vectoriales como ráster) de información geográfica, ofrece múltiples interfaces OGC (*Web Map*

Service, *Web Map Tile Service*, *Web Coverage Service*³⁴, *Web Feature Service*³⁵), está bastante bien documentado, tiene actualizaciones constantes, apoyo de la comunidad y posibilidad de ampliarlo mediante *plugins*.

2.3.3. Configuración de los servicios de publicación

Una parte fundamental en el proyecto han sido los servicios de publicación de los mapas históricos una vez ya georreferenciados, ya que son los componentes encargados de permitir el acceso a estos datos de a través de una interfaz estándar sin que el solicitante tenga que tener ningún conocimiento de formatos de almacenamiento de información geográfica ráster. En ambos escenarios se planteó ofrecer las interfaces WMS (*Web Map Service*) y WMTS (*Web Map Tile Service*) de OGC para acceder a los mapas históricos.

La interfaz WMS se pensó ofrecer porque es una interfaz básica de acceso estándar a mapas que se puede utilizar en cualquier ámbito, ya que permite obtener un mapa de tamaño y extensión geográfica sin limitaciones, dentro de lo razonable. Es decir, te permite obtener un mapa a cualquier escala, al contrario que WMTS cuyas escalas están fijadas. Además, casi todos los servicios que ofrecen el estándar WMTS utilizan un WMS como fuente de datos para crear la cache de teselas de niveles de escalas prefijados, así que es un paso previo necesario para ofrecer el servicio con interfaz WMTS. Por otro lado, la interfaz WMTS se ofreció ya que proporciona un servicio con un tiempo de respuesta muy bajo que permite una buena experiencia de usuario.

En primer lugar, llevamos a cabo un periodo de estudio de la documentación de ambos componentes para conocer sus características y los pasos necesarios para realizar una configuración básica de ambos, *Geoserver* y *Virtual Hub*. Posteriormente activamos el acceso a los mismos mediante la interfaz WMS que ambos implementan y se configuraron un par de mapas para comprobar que era factible y como de eficientes eran. Esta prueba previa se llevó a cabo debido a que el fichero GeoTIFF resultante de la georreferenciación era mucho mayor que el del fichero de partida, lo que podría aumentar el tiempo de acceso a la información que contiene. Tras realizar una batería de peticiones de prueba se comprobó que ambos servicios no respondían con la suficiente presteza para cumplir su papel dentro de la estructura diseñada. Por ejemplo, una petición sobre uno de los mapas de mayor tamaño, tanto en extensión geográfica como en almacenamiento en disco (sobre unos 3,8 GB) en la que se ve toda su extensión en una imagen de 1000x900 píxeles tardaba unos 15 segundos en ser respondida por el servicio *Geoserver* y casi 20 segundos por el *Virtual Hub*, lo que no es aceptable para las necesidades a cubrir por nuestro sistema.

Además, se observó que el consumo de recursos de ambos servicios era bastante elevado. En principio este problema se vería paliado con el uso de la interfaz WMTS. Pero al tener que usar éste el servicio de WMS para crear la cache de teselas, provocaría que su inicialización fuese muy lenta.

Debido a estos problemas se tomó la decisión de investigar posibles maneras de optimizar la estructura y formato de almacenamiento de los mapas históricos para mejorar tanto el acceso a la

³⁴ <http://www.opengeospatial.org/standards/wcs>

³⁵ <http://www.opengeospatial.org/standards/wfs>

información que contienen como el consumo de recursos de los servicios que los sirven. Como primera opción se estudiaron las posibilidades ofrecidas por GDAL/OGR debido a que era una herramienta que ya estaba siendo utilizada en el proceso de georreferenciación y está orientada al trabajo con información ráster. Se estudió su documentación y tras encontrar los comandos que nos podrían ser útiles se realizaron múltiples pruebas con estos para intentar optimizar el tamaño y tiempo de acceso a los ficheros GeoTIFF.

Los mejores resultados se lograron mediante la ejecución de los comandos que se incluyeron en el siguiente *script*, *Figura 8*.

```
2 for f in "$(dirname "$0")"/*
3 do
4   if [ ${f: -4} == ".tif" ]
5   then
6     filebasename=$(basename "$f")
7     extension="${filebasename##*}"
8     filename="${filebasename%.*}"
9     echo "Procesando $filebasename..."
10    #gdalinfo $filebasename
11    echo "Ejecutando gdal_translate..."
12    gdal_translate -of GTiff -co TILED=YES -co COMPRESS=DEFLATE
13      $filebasename ./procesados/$filebasename
14    echo "Ejecutando gdaladdo..."
15    gdaladdo --config COMPRESS_OVERVIEW DEFLATE --config
16      INTERLEAVE_OVERVIEW PIXEL -r average ./procesados/$filebasename 2 4 8 16 32 64
17    echo "Procesado"
18  fi
19 done
```

Figura 8. Script de optimización de ficheros GeoTIFF

Un *script* es un archivo que contiene un conjunto de comandos que son ejecutados secuencialmente, desde el primero hasta el último. Este script, para Linux, escanea todos los ficheros con extensión .tif de un directorio y ejecuta sobre ellos los siguientes comandos:

- *gdal_translate*: nos permite realizar distintos tipos de operaciones sobre datos ráster, como conversión, recorte, remuestreo y reescalado de píxeles. En esta ocasión se utiliza para mosaicar la imagen, parámetro *TILED=YES*, lo que aumentara la eficiencia de acceso y recomprimirla, parámetro *COMPRESS=DEFLATE*.
- *gdaladdo*: se puede utilizar para construir o reconstruir *overviews* para la mayoría de los formatos de archivo. Un *overview* es una versión de la imagen inicial de menor resolución utilizada para mejorar el acceso. En este caso se han creado 6 niveles de *overview* para reducir el tiempo de acceso a los mapas, usando un método de muestro bilineal.

Se realizaron las mismas pruebas con los ficheros optimizados, obteniéndose unos tiempos de acceso de menos de 2 segundos en el caso de *Geoserver* y unos 3,5 segundos en el caso de *Virtual Hub*, lo cual era una mejora muy buena. Además, se consiguió reducir el tamaño de las imágenes en todos los casos, con porcentajes entre un 15% y un 40 % y reducir el consumo de recursos por parte del servicio. Esto se logra sobre todo gracias a la compresión de la imagen y a la creación de *overviews*, que agilizan en gran manera el acceso a los datos en escalas menores que la nativa del mapa.

Este resultado tan prometedor llevó a desestimar ofrecer el estándar WMTS en la infraestructura IDEZar, ya que este servicio sería un componente extra que necesita un mantenimiento y esfuerzo de configuración extra. De incluirlo, sería obligatorio configurar en este servicio cada mapa ofrecido a través del servicio WMS para crear un cache de teselas, lo que es costoso en tiempo y en espacio de almacenamiento. Y esta operación sería necesaria al incorporar nuevos mapas en el sistema, lo cual es bastante probable.

En el caso de ENERGIC OD se continuó con el mismo planteamiento de WMS y WMTS, ya que los estándares eran ofrecidos de manera nativa por el *Virtual Hub* en el mismo servicio. El estándar WMTS no dependía de que hubiese un servicio WMS como fuente de datos, ya que *Virtual Hub* en teoría a partir de la configuración de una fuente de datos, un mapa histórico, ofrecía los dos interfaces a la vez y sin costes extras de mantenimiento y configuración.

La configuración de ambas interfaces se realizó con éxito en *Virtual Hub*. Tras esto se observó que el soporte a la interfaz de WMTS no se había desarrollado de forma eficiente. Éste planteaba dos problemas de base que hacían que su uso fuese posible pero no recomendable en un sistema finalista:

- **Configuración:** los niveles de resolución que almacenará la cache de teselas están prefijados y siempre van desde el nivel 0 al elegido en la configuración del servicio, con un máximo de 30. Esto provoca que no se puedan generar los niveles en un intervalo dado, desde el nivel 10 al 18 por ejemplo. Además, al no poder definir los niveles de resolución ofrecidos por el servicio, que no son más que escalas de mapa que un WMS ofrece sin ningún problema, reduce la funcionalidad del servicio y las posibilidades de interacción con otros componentes.
- **Cacheo inexistente:** Las teselas devueltas por la interfaz WMTS no se cachean realmente. Se generan al vuelo accediendo a los ficheros que contienen los mapas históricos georreferenciados y se desechan. Esto desvirtúa la idea con la que fue definido la interfaz WMTS, ya que al no cachear ninguna tesela no se produce eficiencia de respuesta a cambio de prefijar las escalas de visualización.

Virtual Hub sí que cumple con su objetivo principal de servir como un único punto de entrada homogeneizador a distintas y heterogéneas fuentes de datos, pero en el momento de su integración en el proyecto su estado de desarrollo no era lo suficientemente maduro para cumplir con la estabilidad y eficiencia requerida por IDEZar.

Por último, como ya se ha comentado a lo largo del presente documento, el componente de publicación *Virtual Hub* estaba en desarrollo durante la elaboración de este PFC. Esto significa que no era un componente final, podían faltar funcionalidades que necesitásemos para un óptimo funcionamiento del sistema y aparecer fallos y comportamientos no deseados.

A continuación, se exponen los problemas encontrados durante la configuración del servicio, que fueron reportados debidamente a los socios del proyecto responsables para que fuesen solventados:

- Problemas con la carga de ficheros GeoTIFF de gran tamaño. Algunos de los ficheros GeoTIFF que contenían los planos de mayor tamaño no eran reconocidos y cargados por el servicio.
- Lectura incorrecta de ficheros GeoTIFF. Algún fichero GeoTIFF no era interpretado correctamente por la herramienta y daba como lugar colores extraños en la imagen, *Figura 9*.



Figura 9. Lectura incorrecta de formato GeoTIFF por Virtual Hub

- Únicamente se soporta una versión para cada uno de las interfaces ofrecidas. Esto limita el acceso e interoperabilidad del servicio. Por ejemplo, actualmente la interfaz WMS soporta las versiones 1.1.1 y 1.3.0, debiendo elegir entre una de ellas.
- Política de cacheo. La política de creación de la matriz de tiles usada para satisfacer la interfaz WMTS no era la más adecuada, ya que no almacenaba los tiles producidos en disco, y por lo tanto no era un cacheo real. Se cumple la interfaz WMTS, pero no se aprovecha realmente su potencia. Además, la configuración de los niveles de resolución ofrecidos era fija y de 0 a n, no pudiendo configurar un subconjunto como por ejemplo de 4 al 11.
- Problemas con los sistemas de coordenadas ofrecidos. En el fichero obtenido como respuesta a una petición de *GetCapabilities* proporcionada por la interfaz WMS, informaba soporte a cuatro sistemas de coordenadas, EPSG:3035³⁶, EPSG:25830, EPSG:4326 y EPSG:3857, pero solamente se obtenía respuestas a las peticiones realizadas en el sistema de coordenadas EPSG:4326.

El proceso de instalación de ambos servicios de publicación y de configuración de los mismos se documenta ampliamente en el **Anexo IV**.

³⁶ <http://www.epsg.org/>

2.4. Aplicación web de visualización

2.4.1. Estudio de la tecnología existente

2.4.1.1. Tecnología propia

GeoSLab es una empresa de base tecnológica, especializada en el tratamiento digital de la información geoespacial y georreferenciada y sus ámbitos de aplicación con más de diez años de experiencia. Durante todos estos años ha desarrollado tecnología propia en el ámbito de la visualización web de información geográfica. En lo referente a este proyecto, ha desarrollado una solución tecnológica de visualización compuesta por dos componentes. El primero basado en *OpenLayers*³⁷, orientado a la configuración y creación de visores web. El otro basado en *jQuery*³⁸, usado para la creación de interfaces gráficas en páginas web que facilita la interacción entre el visor y el usuario. El desarrollo de la aplicación web Mapas Históricos se desarrollará apoyándose en esta tecnología previa existente.

2.4.1.2. JavaScript

*JavaScript*³⁹ (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar *ECMAScript*⁴⁰. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de *JavaScript* del lado del servidor. Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. Desde el 2012, todos los navegadores modernos soportan completamente *ECMAScript* 5.1. Los navegadores más antiguos soportan por lo menos *ECMAScript* 3. La sexta edición se liberó en julio del 2015.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y *JavaScript* tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código *JavaScript* integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del *Document Object Model* (DOM). Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX⁴¹. *JavaScript* se interpreta en el navegador web al mismo tiempo que las sentencias van descargándose junto con el código HTML.

³⁷ <https://openlayers.org/>

³⁸ <http://jquery.com/>

³⁹ <https://www.javascript.com>

⁴⁰ <http://www.ecmascript.org>

⁴¹ [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))

2.4.1.3. OpenLayers

OpenLayers es una librería JavaScript de código abierto (utiliza una licencia derivada de BSD) usada para mostrar datos de mapas en navegadores web. La librería permite crear aplicaciones web similares a las ofrecidas por *Google Maps*, o *Bing Maps*. Fue desarrollado originalmente por *Metacarta Labs*, aunque desde el 2007 es un proyecto de la *Open Source Geospatial Foundation*. En la *Figura 10* se puede ver un ejemplo de infraestructura de *OpenLayers* y su interacción con otros servicios (principalmente es compatible con servicios OGC) para mostrar la información al usuario. Hay que tener en cuenta que ésta no es la única infraestructura posible, ya que *OpenLayers* permite una gran libertad a la hora de integrarlo con otros servicios y aplicaciones.

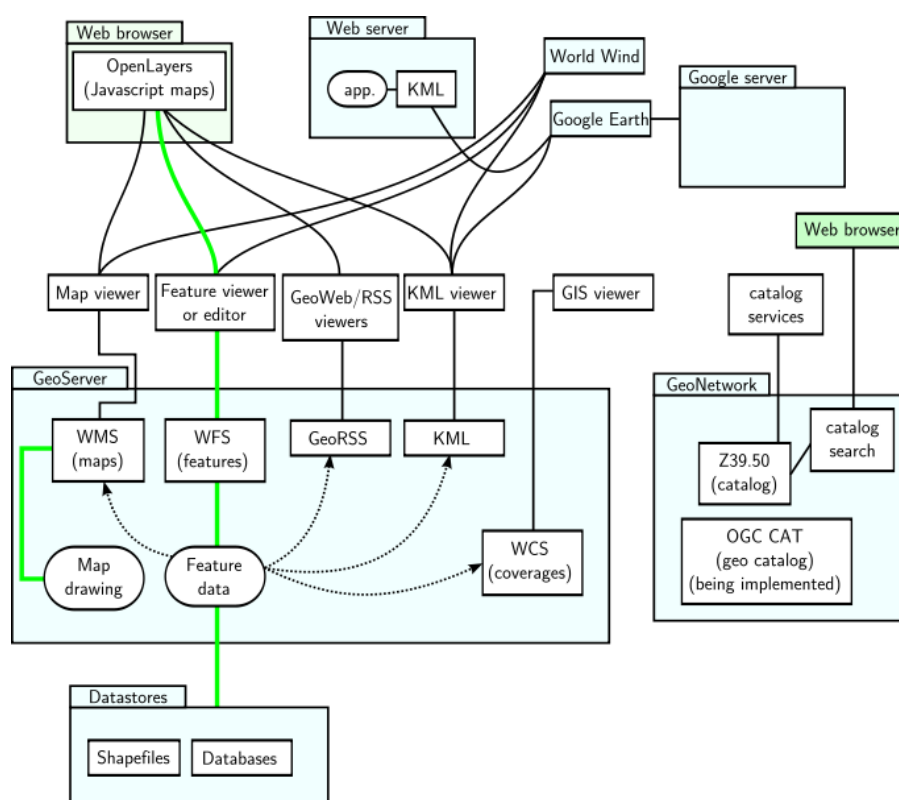


Figura 10. Ejemplo de uso de OpenLayers

2.4.1.4. jQuery

jQuery es una biblioteca multiplataforma de *JavaScript*, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. *jQuery* es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. *jQuery*, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en *JavaScript* que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

jQuery consiste en una serie de ficheros *JavaScript* que contienen las funcionalidades comunes de DOM, eventos, efectos y AJAX. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

Las principales características de *jQuery* son:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un *plugin* básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores *Mozilla Firefox 2.0+*, *Internet Explorer 6+*, *Safari 3+*, *Opera 10.6+* y *Google Chrome 8+*.

2.4.2. Diseño de la Aplicación

A partir de los requisitos del sistema enumerados en el **Anexo II.I** se han extraído características que debería tener la aplicación a desarrollar:

1. El usuario dispone de herramientas para navegar libremente en el mapa: zoom, desplazarse, etc.
2. Acceso a una lista preseleccionada de mapas históricos de la ciudad de Zaragoza ordenada cronológicamente.
3. Poder alternar entre los mapas históricos de la ciudad de Zaragoza preseleccionados.
4. Permitir superponer cualquier mapa histórico sobre el mapa actual de la ciudad.
5. El usuario podrá cambiar el mapa base que muestra la cartografía actual.
6. Se podrá gestionar la transparencia del mapa histórico para facilitar su comparación con el mapa actual.
7. Acceso a información más detallada del mapa.
8. El usuario podrá imprimir una vista del mapa histórico y el mapa actual.
9. El usuario puede seleccionar diferentes capas de otra información útil sobre los recursos que ofrece la ciudad para mostrarlas sobre el mapa histórico seleccionado.

10. Debe ofrecer la mayor usabilidad posible para que pueda ser usado por una gran diversidad de usuarios.
11. Compatibilidad con la mayor cantidad posible de navegadores web.
12. Independencia del dispositivo usado para acceder a la información, para que la aplicación esté siempre disponible y accesible para el usuario.

Con estas características en mente se procedió a plantear los casos de uso de la aplicación y su diseño.

2.4.2.1. Casos de uso

Navegación básica por los mapas históricos

Descripción:

El usuario puede navegar por un conjunto de mapas históricos preseleccionados de la ciudad de Zaragoza, tanto geográfica como temporalmente. Geográficamente, el usuario es capaz de acercar, alejar, mover y establecer el mapa en la extensión original. El usuario es capaz de cambiar libremente entre los mapas históricos preseleccionados, que se presentan ordenados cronológicamente.

Condiciones previas:

El usuario debe acceder a la página web que aloja la aplicación con un navegador web con *JavaScript* habilitado.

Inicio:

El navegador web ha cargado completamente la página web que aloja la aplicación.

Acciones:

1. Aparece un listado de los mapas históricos predefinidos de la ciudad de Zaragoza que permite navegar por ellos temporalmente.
2. El usuario elige el mapa histórico deseado, cargándose este en el visor permitiéndole navegar por el geográficamente.
3. El usuario interactúa libremente con el mapa realizando cualquiera de las actividades siguientes el número de tiempo que desea, en el orden que el usuario decida:
 - Zoom en el mapa alrededor de un punto que el usuario elige.
 - Zoom del mapa alrededor del centro del mapa visualizado.
 - Zoom hacia fuera el mapa alrededor de un punto el usuario elige.
 - Zoom del mapa alrededor del centro del mapa visualizado.
 - Mueva el área visualizada del mapa en cualquier dirección, manteniendo la escala de zoom.
 - Visualizar el mapa en toda su extensión.

- Cambiar a otro mapa histórico preseleccionado desde la zona donde se muestran ordenados cronológicamente
- Solicitud de información (ayuda y leyenda) sobre las herramientas disponibles.
- Solicitud de información sobre la leyenda.

Resultado final:

El usuario ha podido navegar por un conjunto de mapas históricos preseleccionados de la ciudad de Zaragoza, tanto geográfica como temporalmente.

Comparar un mapa histórico con el actual

Descripción:

El usuario puede comparar un mapa histórico con el actual mapa de Zaragoza, teniendo el actual como fondo y el histórico en primer plano con un grado de transparencia establecido. El usuario podrá navegar por ambos mapas geográficamente, pudiendo cambiar en cualquier momento entre todos los mapas históricos y actuales disponibles.

Condiciones previas:

El usuario está ejecutando la exploración básica de mapas históricos caso de uso.

Inicio:

El usuario solicita comparar el mapa histórico en vista con el actual.

Acciones:

1. El mapa histórico visualizado se muestra con un cierto de transparencia, pudiendo ver el mapa con la cartografía actual.
2. El usuario interactúa libremente con el mapa realizando cualquiera de las actividades indicadas en el caso de uso de navegación básica de mapas históricos (que se aplican ahora tanto al primer plano como a los mapas de fondo).
3. Establecer como mapa actual de fondo cualquiera de los disponibles.

Resultado:

El usuario ha podido comparar un mapa histórico con el actual mapa de Zaragoza.

Impresión de un mapa histórico

Descripción:

El usuario podrá imprimir la vista actual del mapa independientemente del mapa histórico o mapa actual seleccionados.

Condiciones previas:

El usuario está ejecutando la exploración básica de mapas históricos caso de uso.

Inicio:

El usuario solicita imprimir el mapa histórico en vista con el mapa actual de fondo.

Acciones:

1. El usuario ha realizado la navegación básica por los mapas hasta obtener la vista que desea imprimir.
2. El usuario pulsa el botón de impresión que le muestra un dialogo de vista previa del resultado de la impresión
3. El usuario acepta lo presentado en la vista previa y decide realizar la impresión.

Resultado:

El usuario ha podido imprimir el mapa histórico con la vista actual de fondo.

2.4.2.2. Diseño de los componentes de la aplicación web

En la *Figura 11* se presenta la arquitectura básica de la aplicación, la cual estará formada por dos componentes. El primero será un visor encargado de proporcionar la funcionalidad que permitirá visualizar los mapas históricos sobre una base cartográfica actual. El segundo será un componente donde se definirá la interfaz gráfica que permitirá la interacción entre el usuario y el visor. La interacción entre estos dos componentes se realizará a través de funcionalidad creada siguiendo el patrón de diseño 'Facade' que ambos componentes ofrecerán. Con esta arquitectura propuesta se produce una separación entre la lógica de la aplicación y su interfaz gráfica, lo que facilita el desarrollo de la misma y reducirá el tiempo de desarrollo de posibles ampliaciones posteriores que se desearan realizar.

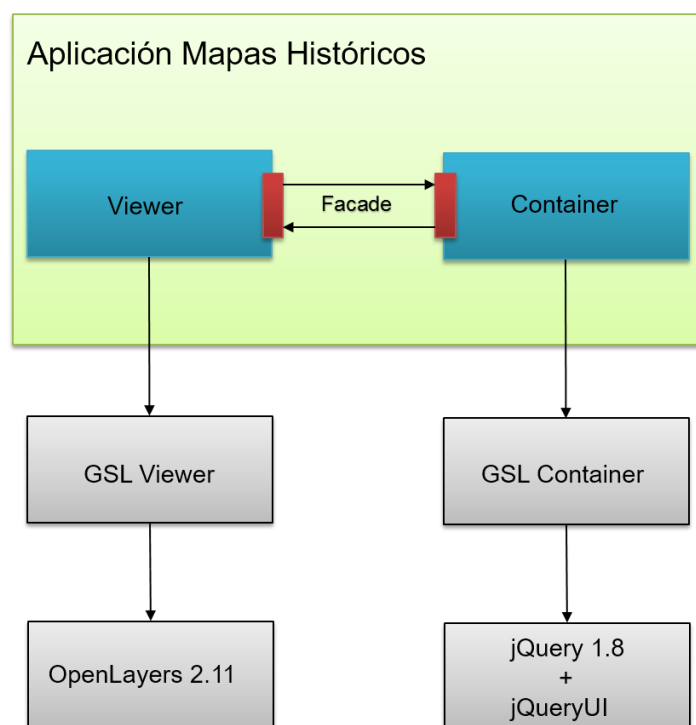


Figura 11. Arquitectura de la aplicación Mapas Históricos

2.4.2.3. Diseño GUI de la aplicación web

En la *Figura 12* se adjunta el diseño conceptual de la aplicación que se planteó. En él se pueden observar tres partes diferenciadas.

En la parte superior se mostrará un carrusel donde aparecen todos los mapas históricos disponibles ordenados cronológicamente con dos controles para navegar entre ellos. Al pulsar sobre cada uno de los mapas será mostrado en el visor.

En la parte central se insertará el visor geográfico con todas las herramientas de navegación estándar. Permitirá la visualización de la cartografía base actual sobre la que se colocaran los mapas históricos.

Por último, en la parte derecha habrá un panel donde se mostrará la información del mapa histórico mostrado y un control para definir el grado de transparencia con el que se visualizará el mapa histórico. Esto permitirá ver los cambios que ha experimentado la ciudad. Además, habrá un enlace que dará acceso a la sección correspondiente del archivo municipal del Ayuntamiento de Zaragoza, en la que se podrá obtener información ampliada

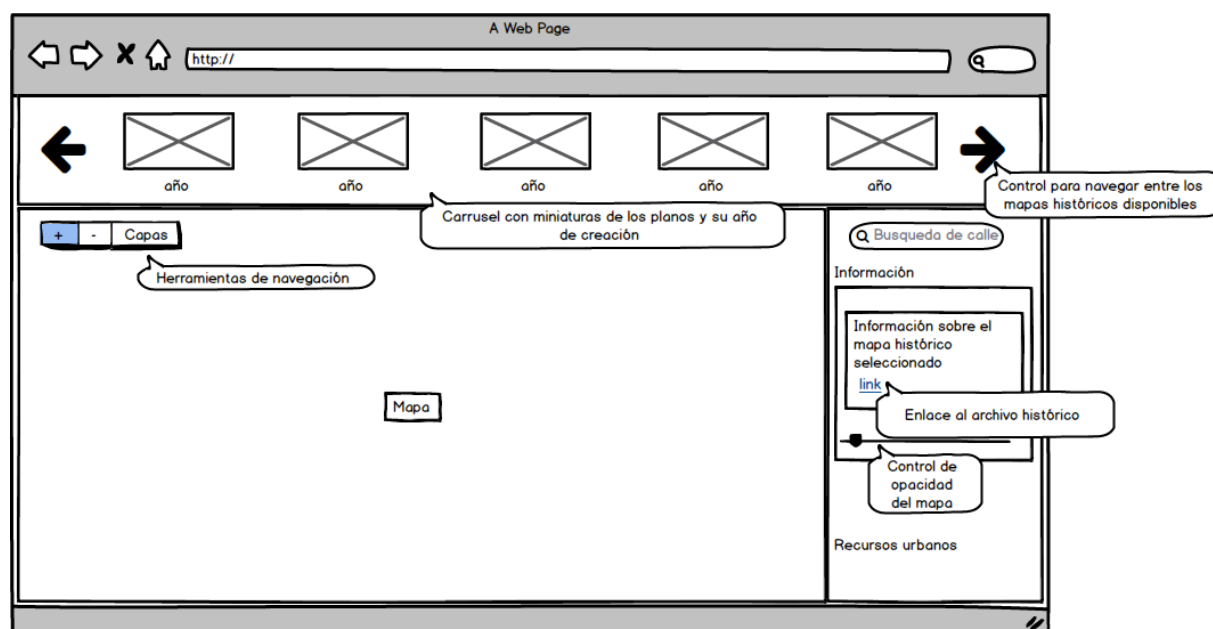


Figura 12. Diseño conceptual del GUI de la aplicación

2.4.3. Desarrollo

El desarrollo de la aplicación web de visualización se ha basado en tecnología de visualización existente previamente. Esta tecnología no consiste en un visor que modificar o similar, si no que provee de controles, componentes y elementos que deben configurar e integrar para construir un visor o la interfaz gráfica que hará de intermediario entre el usuario y el visor. Esto permite construir aplicaciones a medida que resuelven problemas concretos, aunque hay que realizar una formación previa de esta tecnología y sus particularidades.

Finalmente se han construido los dos componentes que aparecen en la *Figura 11*. En primer lugar, basándose en la tecnología de visualización existente se ha construido el componente visor. Se han

configurado los controles y elementos necesarios. Después se ha desarrollado la funcionalidad que permite la interacción entre ellos y ensamblado todo para conseguir un componente de visualización que cumpla nuestros requisitos. En segundo lugar, basándonos en tecnología también existente, se ha seguido el mismo proceso para crear el componente contenedor. Engloba toda la interfaz gráfica que permitirá la interacción entre el usuario y el componente visor desarrollado en primer lugar. Una parte importante de esto es la comunicación entre ambos componentes. Esta interacción se hará mediante la funcionalidad creada expresamente siguiendo el patrón de diseño '*Facade*'.

Durante el proceso de desarrollo se han tenido que resolver problemas debidos a la propia naturaleza del lenguaje *JavaScript*, gestión del desarrollo y distribución del mismo y limitaciones en el uso del *Virtual Hub* del proyecto ENERGIC OD

En el **Anexo V** se presenta en mayor detalle todo lo enumerado en esta sección.

2.5. Plan de pruebas e integración

Una vez realizado el diseño y desarrollo, se llevó a cabo una fase de test exhaustiva en la que se pasaron unas pruebas funcionales y de rendimiento, que garanticen el correcto funcionamiento del sistema resultante. Se proponen los siguientes bloques de test para los que se definirá un plan de pruebas detallado:

- Pruebas funcionales, que garanticen el adecuado funcionamiento del software desarrollado de acuerdo a los requisitos planteados en el presente pliego.
- Testeo de usabilidad y rendimiento en las últimas versiones de los navegadores más utilizados (Internet Explorer, Mozilla Firefox, Google Chrome, Safari y Opera).

Se ha diseñado un plan de pruebas que se ejemplifica en el **Anexo VI**. Para cada una de las pruebas definidas en dicho plan se indica la descripción/propósito de la prueba, acciones previas, acciones a ejecutar y resultados esperados y el navegador web en que se prueba el sistema, realizando las mismas en múltiples navegadores. Si tras su ejecución se detecta que el sistema no cumple con el plan de pruebas establecido, se realizarán las modificaciones necesarias para cumplirlo y se volverá a ejecutar. Asimismo, se han usado distintos entornos de desarrollo y producción para minimizar los problemas de instalación, configuración y funcionamiento del software desarrollado. Se listan a continuación los entornos propuestos:

- Entorno de desarrollo en las dependencias de la empresa GeoSLab, que replicará el entorno de la Sede Electrónica municipal, pudiendo comprobar previamente en desarrollo el correcto funcionamiento de todo el software desarrollado.
- Entorno de producción en los sistemas de la Sede Electrónica municipal, integrado con el entorno habitual de aplicaciones y de base de datos.

La integración en el entorno de producción se realizará una vez que el software haya pasado satisfactoriamente el plan de pruebas en el entorno desarrollo. Para este proceso se siguieron los siguientes pasos en cada una de las máquinas que actuaban como servidor:

1. Copiar los ficheros GeoTIFF que almacenaban los mapas históricos en el servidor.
2. Instalar y configurar *Apache Tomcat*. Este paso se hace si no hay un contenedor de aplicaciones valido.
3. Desplegar y configurar el servicio de publicación de información geográfica deseado.
4. Configurar y desplegar el componente visor de la aplicación web.
5. Configurar y desplegar el componente contenedor de la aplicación web.

3. Conclusiones

A lo largo de esta sección se va a comentar los resultados que se han obtenido tras la realización del proyecto. Se expondrán posibles líneas futuras de trabajo que continúen el desarrollo realizado. También se comentarán las dificultades y problemas que se han encontrado a lo largo del proyecto. Finaliza la sección una breve valoración personal del Proyecto Fin de Carrera.

3.1. Resultado del proyecto

Los objetivos que se han planteado para este proyecto se han cumplido satisfactoriamente. Se ha realizado el desarrollo de un sistema capaz de integrar en un servicio de publicación de datos geográficos estandarizados por el *Open Geospatial Consortium* (OGC), datos geográficos no concebidos con este fin. Concretamente, una serie de planos de la ciudad de Zaragoza creados en los últimos 300 años y escaneados a muy alta resolución. Asimismo, hemos conseguido visualizarlos a través de un visor geográfico con las funcionalidades habituales en este tipo de aplicaciones añadiendo algunas nuevas destinadas a dar relevancia a los documentos históricos.

El sistema desarrollado está actualmente accesible dentro de la plataforma IDEZar. ‘Zaragoza Histórica’ es una nueva herramienta que permite descubrir la evolución de la ciudad a lo largo de su historia. Nace como un nuevo entorno innovador en el que acercar este valioso material a la ciudadanía. Esta nueva herramienta proporciona un visor interactivo con la cartografía actual de la ciudad sobre la que el usuario puede superponer y comparar una selección del catálogo de mapas históricos de la ciudad del archivo municipal de Zaragoza. Estos mapas se muestran permitiendo escoger la transparencia deseada para facilitar la comparación de los mismos y ayudar al ciudadano a descubrir la evolución de la ciudad a lo largo de los años.

Además, se ha cumplido el objetivo propuesto en el proyecto ENERGIC OD al ser una de las diez aplicaciones a desarrollar para demostrar la validez del concepto, diseño, implementación e implementación de *Virtual Hub*, a pesar de los problemas que se han encontrado para su uso e integración en el sistema. Estas aplicaciones debían abordar las necesidades de las empresas, los ciudadanos y las instituciones, haciendo uso de los datos geoespaciales públicos, lo cual se ha conseguido plenamente.

3.2. Líneas futuras

A lo largo del desarrollo del proyecto y durante las pruebas han ido surgiendo nuevas propuestas en las que poder seguir desarrollando y mejorando el sistema. Algunas de las nuevas posibilidades que se han recogido son las siguientes:

- Geolocalización de más planos antiguos del catálogo del Archivo municipal⁴² y su posterior inclusión en el sistema desarrollado, lo que enriquecerá la funcionalidad y experiencia de uso.
- Incorporar fotografías antiguas sobre la cartografía de la ciudad. Al seleccionar uno de los mapas históricos, se visualizarían las imágenes correspondientes a ese año. De esta forma se muestra una visión completa de la ciudad en una época determinada, mostrando las imágenes sobre la cartografía en un año concreto, *Figura 13*.

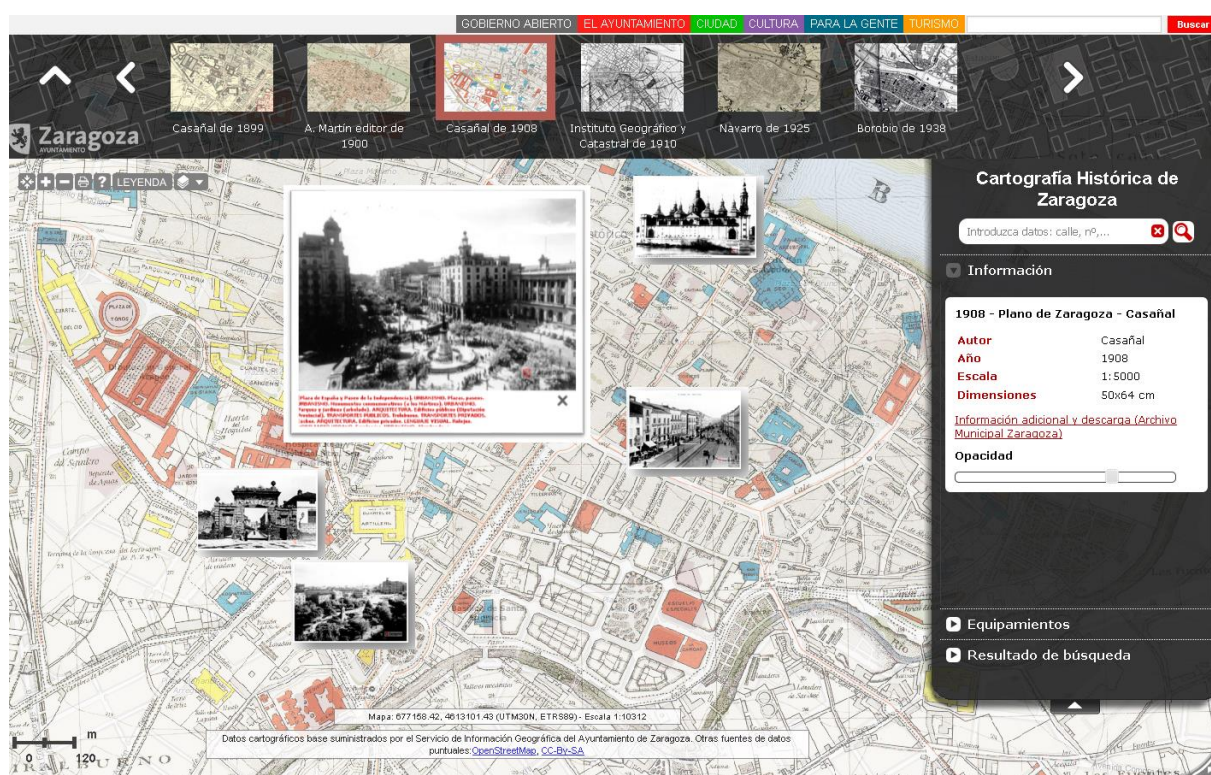


Figura 13. Ejemplo de visualización de fotos antiguas correspondientes a la época sobre un mapa histórico

Para poder ofrecer la mejora anteriormente expuesta se necesitaría en primer lugar, poder georreferenciar las fotografías mediante algún proceso automático y además ofrecer un mecanismo de corrección manual que permitiese corregir los posibles errores de la localización automática de las imágenes, por ejemplo, una interfaz web con un visor.

⁴² <http://www.zaragoza.es/nuba/app/home/>

- Comparativa de dos mapas antiguos visualizándolos simultáneamente. Se dividiría la ventana para mostrar dos planos históricos simultáneamente y facilitar su comparativa. Ambas ventanas estarían sincronizadas (se mueven a la vez en respuesta a la interacción del usuario), pudiéndose desbloquear para que se muevan independientemente si se desea, *Figura 14*.

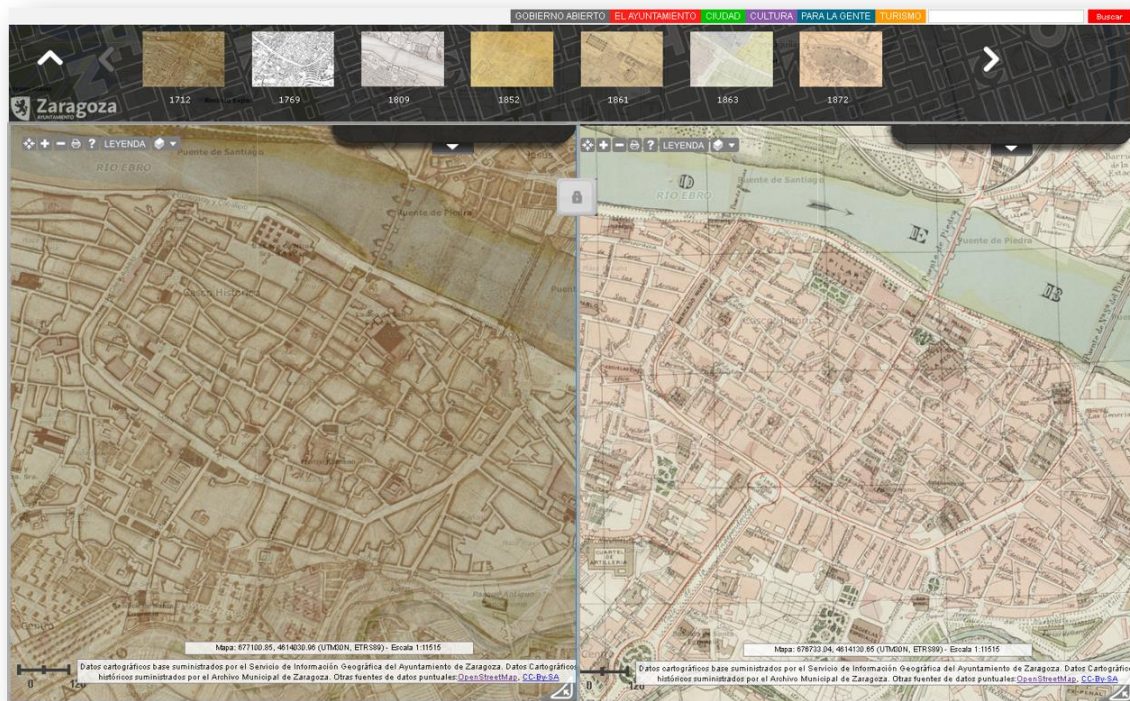


Figura 14. Ejemplo de la visualización de dos mapas históricos simultáneamente

3.3. Incidencias y problemas encontrados

Los problemas que han surgido durante la elaboración del proyecto han sido muchos y de índole muy variada, que han podido ser resueltos consiguiendo así lograr los objetivos establecidos para el proyecto.

El primer problema con el que hubo que lidiar fue la optimización de los datos después de enriquecerlos con información geoespacial. Se pretendía utilizar formatos de almacenamiento eficientes en acceso y de gran compresión como ECW o JPG2000. Pero la necesidad de superponer los planos históricos con una base cartográfica actual impidió su uso, teniendo que optar por el formato GeoTIFF, el cual permite transparencia, pero no tiene las ventajas de los anteriores. Esto provocó que la velocidad de respuesta de los servicios de publicación no fuese suficiente como para ofrecer una experiencia satisfactoria de uso. Esto se solventó con el uso de la suite GDAL/OGR con el fin de encontrar una combinación de opciones de almacenamiento interno del formato GeoTIFF que permitiesen un uso válido de estos datos.

Otro problema relacionado con la velocidad de acceso fue el uso de las interfaces ofrecidas por *Virtual Hub*. Éste está pensado como un elemento homogeneizador de diferentes fuentes y formatos de datos y el precio a pagar por esto es la eficiencia en el acceso a los mismos. *Virtual Hub* realiza al

vuelo el acceso a las fuentes y las diferentes transformaciones necesarias para responder a través de la interfaz seleccionada, lo que provoca un mayor tiempo de respuesta. En este caso tanto la interfaz WMS como la interfaz WMTS no respondían con la suficiente eficiencia para ofrecer una experiencia de uso correcta, por lo que hubo que utilizar *Geoserver* en IDEZar para solventar este problema.

Por último, en el cliente desarrollado para el proyecto ENERGIC OD surgió un problema al utilizar la tecnología existente para visualizar los mapas en la aplicación Mapas históricos. Este problema apareció al usar *Google Maps* como el mapa base de la visualización. En este caso, la capa base de *Google Maps* fuerza el sistema de coordenadas que debe ser utilizado por la herramienta de visualización. En el contexto de esta aplicación, era necesario añadir capas WMTS con otro sistema de coordenadas, haciendo imposible usar *Google Maps* como mapa base de la visualización.

Se pensó en utilizar el servicio WMTS ofrecido por IDEZar, pero tampoco era compatible con la interfaz WMTS proporcionada por *Virtual Hub*. Es necesario que los niveles de zoom sean los mismos en todos los servicios que se van a superponer y en este caso eran diferentes. Debido a esto, se propuso utilizar el WMTS del Instituto Geográfico Nacional como mapa base para solucionar este problema.

3.4. Valoración personal

La realización de este Proyecto Fin de Carrera en GeoSLab me ha aportado una gran cantidad de nuevas experiencias. He participado en el desarrollo de una aplicación que va a formar parte de una infraestructura de prestigio como es IDEZar y que va a ser utilizada por la ciudadanía y expertos relacionados con la historia. Así mismo va a permitir el uso y aprovechamiento de unos datos que hasta ahora no eran explotados en todo su potencial.

He vivido por primera vez la experiencia de desarrollar un sistema completo para un proyecto europeo, conviviendo con los problemas que surgen, como la comunicación con otros compañeros de diferentes países. Junto con estas nuevas experiencias, el desarrollo de este proyecto me ha permitido obtener conocimientos SIG, desconocidos previamente. Además del aprendizaje de nuevas técnicas de programación eficiente, gestión y desarrollo de proyectos.

El haber trabajado utilizando proyectos de *Open Source*, ha conseguido que le dé una mayor importancia a una buena documentación producida por el desarrollador. En el caso de *Virtual Hub* la documentación era escasa, lo que ha supuesto un mayor esfuerzo para el desarrollo del proyecto. Y no solo la documentación, sino también a la creación de un código limpio y estructurado, bien comentado, de forma que facilite su comprensión.

Por tanto, la realización de este proyecto ha sido gratificadora, trabajando con un grupo de personas que me han ayudado con los problemas que me han ido surgiendo y con los cuales no me había enfrentado previamente. Adquiriendo así nuevos conocimientos en temas desconocidos hasta el comienzo del proyecto, además de haber vivido una experiencia que me será muy útil a partir de ahora de cara al mundo profesional.

Índice de figuras

Figura 1. Estructura de capas de un SIG	2
Figura 2. Detalle la estructura de datos de una TileCache	5
Figura 3. Cronograma de planificación del proyecto.....	10
Figura 4. Arquitectura del sistema propuesta para integrar con IDEZar.....	13
Figura 5. Arquitectura en detalle de la solución propuesta para IDEZar.....	14
Figura 6. Detalle de la arquitectura para el caso ENERGIC OD.....	15
Figura 7. Ejemplo de toma de puntos de control	18
Figura 8. Script de optimización de ficheros GeoTIFF	23
Figura 9. Lectura incorrecta de formato GeoTIFF por Virtual Hub.....	25
Figura 10. Ejemplo de uso de OpenLayers	27
Figura 11. Arquitectura de la aplicación Mapas Históricos	31
Figura 12. Diseño conceptual del GUI de la aplicación.....	32
Figura 13. Ejemplo de visualización de fotos antiguas correspondientes a la época sobre un mapa histórico	36
Figura 14. Ejemplo de la visualización de dos mapas históricos simultáneamente	37
Figura 15. Menú Proyecto de QGIS	53
Figura 16. Dialogo de propiedades del proyecto de QGIS.....	54
Figura 17. Dialogo de gestión de WMS/WMTS configurados de QGIS	55
Figura 18. Dialogo de configuración de conexión a WMS.....	55
Figura 19. Dialogo de carga de WMS tras configurar el acceso al WMS IDEZar.....	56
Figura 20. Visualización de la capa base del WMS de IDEZar	57
Figura 21. Selección de la herramienta de georreferenciación.....	58
Figura 22. Herramienta de georreferenciación	58
Figura 23. Mapa añadido a la herramienta de georreferenciación.....	59
Figura 24. Dialogo para introducir coordenada de mapa.....	60
Figura 25. Mapa base de referencia para obtener la coordenada del mundo real.....	60
Figura 26. Ejemplo de puntos de control	61
Figura 27. Dialogo de configuración de la transformación.....	63
Figura 28. Comandos de ejemplo	64
Figura 29. Despliegue de fichero .war en Apache Tomcat	67
Figura 30. Pantalla inicial de Geoserver	70

Figura 31. Pantalla principal de configuración de Geoserver.....	71
Figura 32. Pantalla de workspaces	72
Figura 33. Configuración de workspace	73
Figura 34. Pantalla de drivers de acceso a datos fuentes.....	73
Figura 35. Configuración de driver de acceso a datos fuente GeoTIFF.	74
Figura 36. Selección de datos a publicar	75
Figura 37. Detalle de pantalla de configuración de 'Layer'	75
Figura 38. Detalle de pantalla de configuración de 'Layer'	76
Figura 39. Pantalla de previsualización de layers publicadas	76
Figura 40. Previsualización de capa en el cliente interactivo	77
Figura 41. Esquema gráfico de un Virtual Hub	78
Figura 42. Pantalla inicial del componente GI-axe del Virtual Hub	79
Figura 43. Detalle de la pantalla de login	80
Figura 44. Configuración de sources e interfaces.....	81
Figura 45. Configuración de un source	81
Figura 46. Servicio después de configurar un source	82
Figura 47. Configuración WMS	83
Figura 48. Configuración WMTS	83
Figura 49. Servicio tras su configuración	84
Figura 50. Estructura de los componentes de la aplicación de visualización.....	85
Figura 51. Vista general de la aplicación	88
Figura 52. Carrusel de selección de mapa histórico	89
Figura 53. Visualización del mapa histórico	90
Figura 54. Detalle del elemento 'Información'	90
Figura 55. Detalle del elemento 'Equipamiento'	91
Figura 56. Equipamientos mostrados sobre el mapa histórico	91
Figura 57. Detalle del elemento 'Busqueda'.....	92
Figura 58. Visualización de un resultado de la búsqueda.....	92
Figura 59. Ejemplo de las pruebas realizadas en el plan de pruebas	93
Figura 60. Ejemplo de las pruebas realizadas en el plan de pruebas	94

Anexo I Estado del arte

Como primera tarea del proyecto, se realizó un estudio del estado del arte actual. Este consiste en documentarse sobre las técnicas y programas existentes hoy en día relacionados con la temática del proyecto. Este estudio permitió aportar nuevos puntos de vista al proyecto, así como nuevas ideas no planteadas en un inicio, y la elección de la base del sistema a desarrollar. El estudio comenzó conociéndose los diferentes productos SIG y servicios de publicación de información geográfica que hay en el mercado actualmente.

Anexo I.I Estudio de la situación actual

Este estudio del arte se ha realizado con el objetivo de encontrar las mejores herramientas y/o componentes para:

1. Georreferenciar los mapas históricos proporcionados por el Ayuntamiento de Zaragoza.
2. Publicar y permitir el acceso a estos datos mediante estándares OGC

Además de conseguir estos objetivos fundamentales, durante el transcurso del mismo se han adquirido conocimientos sobre campos como los formatos de almacenamiento de información geográfica ráster, Sistemas de Información Geográfica y estándares OGC que eran necesarios para desenvolverse con soltura durante la duración de este proyecto.

A continuación, se detalla los resultados arrojados tras estas investigaciones. En primer lugar, se describirán las herramientas para el tratamiento inicial de los mapas históricos proporcionados.

Existen una gran cantidad de programas SIG en el mercado, desde los que se encuentran bajo una licencia de software libre hasta los que lo hacen bajo una licencia de carácter comercial. A continuación, se procederá a la descripción de los más reconocidos de los que se han encontrado.

ArcGIS, es un sistema integrado completo, que comparte la misma arquitectura de componentes (*ArcObjects*) con el fin de poder manipular, distribuir, crear y analizar la información geográfica. Este software fue diseñado por la empresa ESRI (*Environmental Systems Research Institute*). Usa estándares abiertos: COM, XML, SQL... para comunicarse con bases de datos y servidores. Gracias a la funcionalidad que le proporciona el soporte de sus clientes (*ArcView*, *ArcInfo*, *ArcEditor*...) y servidores (*ArcSDE* y *ArcIMS*), facilita la gestión de datos, planificación, operaciones comerciales y análisis de datos. Está disponible únicamente para Windows y Linux. A parte de la aplicación de escritorio, posee también *ArcGIS Server*, para la publicación y gestión web y *ArcGIS Mobile*, para la captura y gestión de información en campo. Esta última herramienta permite aumentar la precisión en los puntos georreferenciados y mejorar la actualización de los datos. Es fácil de usar por personal de campo que no tiene por qué tener experiencia en SIG. Está asociado a licencias de pago bastantes elevadas

OpenJUMP, es un SIG de código libre que permite realizar consultas, crear y editar datos geográficos de formatos diferentes. Está programado en Java y es multiplataforma, es decir, puede ejecutarse en entornos distintos como Windows, Linux o Mac OS. Es un sistema altamente intuitivo que basa su funcionalidad en módulos. Esta arquitectura modular permite crear numerosos *plugins* para numerosos formatos que añaden funcionalidades distintas a la aplicación. Así, para cargar cualquier

tipo de imagen o dato vectorial sólo habrá que cargar o programar el módulo en cuestión. Su interfaz es parecida a la de *ArcView*. Tiene herramientas de edición muy interesantes para la modificación de datos vectoriales y otras herramientas básicas de geoprocésado (zonas de influencia, uniones, etc.). Cualquier nueva función es fácil de añadir. Con este software, es posible conectarse a servidores de cartografía WMS. Existen versiones internacionalizadas y varios sitios con información sobre *OpenJUMP*, tanto para la creación de nuevos *plugins* como para generar nuevos proyectos o programas basados en este software que aporten nuevas funcionalidades y más específicas.

uDIG, tiene su origen en la empresa Refracctions (creadores de *PostGIS*) y es considerado en gran parte por eso como el sucesor de *OpenJUMP*, además de porque lo utiliza como base de algoritmos para el análisis de los datos espaciales. *uDIG* también utiliza *Geotools* como librería para la entrada y salida de datos, gracias a lo que es capaz de trabajar con un gran número de formatos. Este software también está basado en los estándares del OGC (*Open Geospatial Consortium*) que regula los estándares abiertos e interoperables de los Sistemas de Información Geográfica. Está programado en Java. De entre sus ventajas cabe destacar que permite la conexión a servidores WFS (Web Feature Service) o acceso a través de Internet a objetos geográficos no sólo a efectos de su visualización en un mapa (WMS) sino además para consulta y descarga, es decir, en modo lectura y escritura. Eso le hace convertirse en un proyecto cada día más prometedor, aunque actualmente aún esté en fase de desarrollo. Se caracteriza también por su capacidad de impresión y productos gráficos en los más diversos formatos, además de por su diseño modular que permite ser reutilizado también en diversos proyectos o programas.

gvSIG, es un Sistema de Información Geográfica que tiene su origen en la Generalitat Valenciana. Es gratuito y con licencia GNU/GPL (Licencia Pública General de GNU o sistema operativo totalmente libre). Es un software integrador, capaz de trabajar con información de cualquier tipo u origen, tanto en formato ráster como vectorial, y comparte algunas características con *OpenJUMP*, como su arquitectura modular o su carácter multiplataforma. Además, permite trabajar con formatos de otros programas como *Autocad*, *Microstation* o *ArcView*, de acuerdo con los estándares OGC (*Open Geospatial Consortium*) que regula los estándares abiertos e interoperables de los Sistemas de Información Geográfica. Las herramientas que implementa permiten una gran precisión en edición cartográfica. Posee otras funciones básicas como diseño de impresión y soporte de los formatos más populares, tanto vectoriales como de imágenes. *gvSIG* es una aplicación de la que ya existen varias versiones, y aunque su funcionalidad está prácticamente cubierta y se ha convertido en una referencia dentro de las tecnologías SIG, que continúa actualmente en fase de desarrollo y perfeccionamiento, siempre bajo los principios de compartir y elaborar. Incluye principalmente las aplicaciones *gvSIG Desktop* y *gvSIG Mobile*.

- *gvSIG Desktop* tiene las herramientas propias de un completo cliente SIG de escritorio.
- *gvSIG Mobile* ideal para proyectos de captura y actualización de datos en campo. Se caracteriza por disponer de una interfaz amigable, siendo capaz de acceder a los formatos más comunes y cuenta con un amplio número de herramientas SIG y GPS ideales para trabajar con información de naturaleza geográfica.

QGIS, es un Sistema de Información Geográfica de código libre, licencia GNU, multiplataforma. Una de sus mayores ventajas es la posibilidad de usar *QGIS* para interactuar con GRASS y GDAL, utilizando toda la potencia de análisis de estos en un entorno de trabajo más amigable. *QGIS* está desarrollado en C++, usando la biblioteca Qt para su interfaz de usuario. Permite la integración de *plugins*

desarrollados tanto en C++ como Python. Una de las principales ventajas de *QGIS* es su facilidad de conexión a base de datos, en particular *PostGIS* (vectorial y ráster), respecto a otros GIS donde la conexión es más tediosa. Se encuentra en constante desarrollo debido a que es un software bajo licencia GNU, por lo tanto, existe una gran comunidad de usuarios desarrollando nuevas funciones y mejorando las existentes. Existe una gran cantidad de documentación disponible.

GRASS GIS (*Geographic Resources Analysis Support System*), es un software libre, de carácter gratuito y con el código fuente a disponibilidad de los usuarios. El *Open Source Geospatial Foundation*, inició este proyecto complejo hace 28 años, con el fin de crear una herramienta eficiente en análisis de datos geoespaciales, procesamiento de imágenes, mapas y modelos espaciales. Su ámbito académico y gubernamental unido a su manejabilidad y potencia, ha permitido su uso a escala mundial. Uno de los inconvenientes principales de GRASS es el hecho de que está diseñado para entornos UNIX, lo cual ha frenado su expansión hacia el público general y su uso se limita a centros universitarios y de investigación. Hoy en día existen versiones de GRASS que se pueden instalar en entornos Windows a través de emulación de *Cygwin*. Otro de los inconvenientes es que este software es demasiado complejo para personas sin conocimientos SIG.

GDAL, *Geospatial Data Abstraction Library* o GDAL es una biblioteca de software para la lectura y escritura de formatos de datos geoespaciales, publicada bajo la *MIT License* por la fundación geoespacial de código abierto OGC (*Open Source Geospatial Foundation*). Como biblioteca, presenta un único modelo abstracto de datos al uso que llama para todos los formatos soportados. También viene con una variedad de utilidades en línea de comando para la traducción y el proceso de datos geoespaciales. GDAL es una biblioteca de software para lectura y escritura de formatos geoespaciales ráster y vectoriales, publicada bajo la licencia de software libre por la *Open Source Geospatial Foundation*. Presenta un modelo de datos abstracto único que homogeniza el acceso a los distintos formatos compatibles, facilitando el trabajo con ellos. Las proyecciones y transformaciones son apoyadas por la biblioteca Proj.4. La biblioteca OGR relacionada (*OGR Simple Features Library*), que forma parte de GDAL, proporciona es utilizada para trabajar con datos vectoriales. GDAL fue desarrollado principalmente por Frank Warmerdam hasta el lanzamiento de la versión 1.3.2, cuando el mantenimiento fue transferido oficialmente al Comité de Gestión del Proyecto GDAL / OGR bajo la *Open Source Geospatial Foundation*. GDAL / OGR es considerado un importante proyecto de software libre por sus "amplias capacidades de intercambio de datos" y también en la comunidad comercial de SIG debido a su uso generalizado y un amplio conjunto de funcionalidades. Son muy numerosas las aplicaciones que usan GDAL para el acceso a datos geográficos entre las que se destacan:

- *MapServer*
- *GRASS GIS*
- *QGIS*
- *Google Earth*
- *Kosmo*

Las ventajas más reseñables de GDAL son una gran solidez por los orígenes y edad del proyecto, poseer herramientas potentes de análisis y tratamiento ráster. Como inconvenientes nos encontramos con una interfaz no muy amigable y una gran curva de aprendizaje para usuarios noveles en el mundo SIG.

Respecto a los servicios de publicación de información geográfica, se siguió el mismo procedimiento que en el caso de la herramienta SIG. A continuación, se destacan los softwares encontrados.

Geoserver, un servidor de código abierto escrito en Java que permite a los usuarios compartir datos geospaciales. Diseñado para la interoperabilidad, publica datos de las principales fuentes de datos espaciales usando estándares abiertos. GeoServer ha evolucionado hasta llegar a ser un método sencillo de conectar información existente a globos virtuales tales como *Google Earth* y *NASA World Wind*, así como mapas basados en web como *OpenLayers*, *Google Maps* y *Bing Maps*. *GeoServer* sirve de implementación de referencia del estándar *Open Geospatial Consortium Web Feature Service*, y también implementa las especificaciones de *Web Map Service* y *Web Coverage Service*.

Mapserver, es un motor de renderizado de datos geográficos de código abierto escrito en lenguaje C. Más allá de exploración de datos SIG, MapServer permite crear “mapa de imágenes geográficos”, es decir, los mapas que pueden dirigir contenido al usuario. Por ejemplo, el *DNR Minnesota Recreation Compass* provee a los usuarios con más de 10.000 páginas Web, reportes y mapas a través de una única aplicación. Esta misma aplicación sirve como “motor de mapas” para otra parte del sitio, suministrando el contexto espacial donde sea necesario. *MapServer* fue originalmente desarrollado por la Universidad de Minnesota (UMN) con el proyecto ForNet en cooperación con la NASA, y el departamento de recursos naturales (MNDNR). Más tarde fue recibido por el proyecto TerraSIP, un proyecto patrocinado entre la NASA y la UMN y un consorcio de intereses en el manejo de la tierra.

Deegree, es un software de código abierto (GNU Lesser General Public License) para infraestructuras de datos espaciales y web geoespacial. Incluye componentes para la gestión de datos geoespaciales, incluyendo acceso a datos, visualización, y seguridad. Los estándares son unos de los pilares de Deegree, cumpliendo con los estándares *del Open Geospatial Consortium* (OGC) y del Comité Técnico ISO 211. Incluye la implementación de referencia del servicio de mapas web de OGC (WMS), un servicio de features web (WFS) totalmente compatible, así como paquetes para servicio de catálogo (CSW), servicio de cobertura web (WCS), servicio de procesamiento web (WPS) y mosaico de mapas web Servicio (WMTS). Desde 2000 Deegree ha sido desarrollado por lat/lon, con la firme intención de convertirlo en un proyecto impulsado por la comunidad. Un paso importante para este efecto fue la aceptación de ser un proyecto *OSGeo* en 2010. Hoy, Deegree es mantenido por varias organizaciones e individuos con una gran base de usuarios en todo el mundo.

Anexo II Análisis del sistema

En este anexo se va a explicar con más detalle los requisitos del sistema a construir. A partir de los requisitos propuestos por el cliente se han obtenido los requisitos del sistema. Estos requisitos serán divididos en requisitos funcionales, no funcionales y datos.

Anexo II.I Requisitos del sistema

Requisitos funcionales:

Esta sección muestra los requisitos funcionales del sistema. Cada requisito en esta sección podrá ser buscado por id completo y único. Para facilitar esto se usará la siguiente notación:

ID:

- F<número del requisito>.
- Todos los números que identifiquen un requisito deberán tener la misma cantidad de dígitos, rellenando con 0 a la izquierda.
- La numeración de los requisitos se iniciará en 0.

Requisito:

- Nombre del requisito.

Descripción:

- Breve descripción del requisito.

ID	F01
Requisito	Acceso de los usuarios a una lista preseleccionada de mapas históricos de la ciudad de Zaragoza
Descripción	El usuario tiene acceso a un listado de los mapas históricos preseleccionados ordenados cronológicamente.

ID	F02
Requisito	Seleccionar un mapa histórico.
Descripción	El usuario dispone de herramientas para cambiar entre los mapas históricos preseleccionados.

ID	F03
Requisito	Navegación libre por el mapa histórico.
Descripción	El usuario dispone de herramientas para navegar libremente en el mapa.

ID	F04
Requisito	Imprimir el mapa histórico.
Descripción	El usuario tiene una herramienta para imprimir la vista actual del mapa independientemente del mapa histórico o mapa actual seleccionados.

ID	F05
Requisito	Herramienta de ayuda.
Descripción	El usuario tiene una herramienta para consultar un menú de ayuda que muestra la leyenda e información útil para la interacción con la aplicación.

ID	F06
Requisito	Comparar el mapa histórico con el mapa actual.
Descripción	El usuario tiene una herramienta para gestionar la transparencia del mapa histórico. Esta herramienta facilitará comparar el mapa histórico y el mapa actual.

ID	F07
Requisito	Cambiar la capa base del mapa actual.
Descripción	El usuario tiene la posibilidad de cambiar libremente la capa base entre todas las disponibles.

Requisitos no funcionales:

Esta sección muestra los requisitos no funcionales del sistema. Cada requisito en esta sección podrá ser buscado por id completo y único. Para facilitar esto se usará la siguiente notación:

ID:

- N<número del requisito>.
- Todos los números que identifiquen un requisito deberán tener la misma cantidad de dígitos, rellenando con 0 a la izquierda.
- La numeración de los requisitos se inicia en 0.

Requisito:

- Nombre del requisito.

Descripción:

- Breve descripción del requisito.

ID	N01
Requisito	Usabilidad.
Descripción	El sistema será utilizado por ciudadanos. Esto implica que debe ser fácil de usar y su interfaz intuitivo.

ID	N02
Requisito	Usabilidad. Ayuda al usuario.
Descripción	Se facilitará el uso de la aplicación mediante <i>tooltips</i> . Se considerará la ayuda en línea y los tutoriales si durante las pruebas aparecen dificultades de uso.

ID	N03
Requisito	Accesibilidad.
Descripción	Se proporcionarán textos alternativos para los mapas.

ID	N04
Requisito	Exactitud .
Descripción	Los datos ofrecidos serán de la mayor calidad posible.

ID	N05
Requisito	Mantenimiento.
Descripción	La aplicación no requerirá, o serán los mínimos posibles, cambios en el caso de añadir nuevos mapas históricos, cambien interfaces o protocolos de acceso.

ID	N06
Requisito	Escalabilidad. Usabilidad.
Descripción	La solución de visualización será desarrollada como una aplicación web que funcionará en un navegador para permitir su uso en todos los sistemas operativos posible.

ID	N07
Requisito	Usabilidad.
Descripción	Los mensajes mostrados al usuario sobre advertencias, errores o cualquier otro tipo tienen que ser claros.

ID	N08
Requisito	Usabilidad.
Descripción	Soporte a diferentes idiomas.

ID	N10
Requisito	Restricción.
Descripción	Integración en infraestructuras ya existentes. El despliegue de la aplicación se hará considerando y respetando el entorno tecnológico ya existente.

Requisitos de los datos:

Esta sección muestra los requisitos de los datos del sistema. Cada requisito en esta sección podrá ser buscado por id completo y único. Para facilitar esto se usará la siguiente notación:

ID:

- D<número del requisito>.
- Todos los números que identifiquen un requisito deberán tener la misma cantidad de dígitos, rellenando con 0 a la izquierda.
- La numeración de los requisitos se inicia en 0.

Requisito:

- Nombre del requisito.

Descripción:

- Breve descripción del requisito.

ID	D01
Requisito	Exactitud.
Descripción	La precisión y exactitud de los mapas históricos georreferenciados debe ser suficiente para tener transiciones suaves al cambiar de un mapa histórico a otro y para permitir su comparación con el mapa actual o entre ellos.

ID	D02
Requisito	Resolución.
Descripción	La resolución de los mapas históricos debe ser la mayor posible para permitir realizar zoom sobre ellos.

ID	D03
Requisito	Contenido.
Descripción	Los mapas históricos mostrados corresponderán a mapas proporcionados por el archivo histórico de la ciudad de Zaragoza.

ID	D04
Requisito	Cobertura.
Descripción	La extensión geográfica de los mapas históricos corresponderá como máximo a la de la ciudad de Zaragoza.

ID	D05
Requisito	Formato.
Descripción	Los mapas históricos deberán convertirse a un formato que permita su georreferenciación.

Anexo III Tratamiento y procesamiento de los mapas históricos

Anexo III.I Instalación QGIS

Linux Ubuntu

La mayoría de las distribuciones de Linux dividen *QGIS* en varios paquetes. Los paquetes básicos que son necesarios son *qgis* y *qgis-python* (para ejecutar *plugins*). Otros paquetes como como *qgis-grass* (o *qgis-plugin-grass*) y *qgis-server* se pueden omitir inicialmente o instalarse sólo cuando los necesite.

En los repositorios predeterminados suelen encontrarse versiones antiguas de QGIS, por eso es recomendable seguir estos pasos para obtener la herramienta desde los repositorios oficiales del proyecto.

- Añadir el repositorio a los *sources* del sistema:

```
$ deb *repository* *codename* main
```

```
$ deb-src *repository* *codename* main
```

Con el parámetro *repository* seleccionaremos entre las distintas versiones disponibles de QGIS y con el parámetro *codename* indicaremos la distribución de Linux Ubuntu en la que se va a instalar el componente. Por ejemplo, para instalar la última versión de QGIS en Ubuntu 16.04 se ejecutarían los siguientes comandos:

```
$ deb http://qgis.org/debian xenial main
```

```
$ deb-src http://qgis.org/debian xenial main
```

- Actualizar el listado de paquetes disponibles e instalar los paquetes básicos y los opcionales deseados:

```
$ sudo apt-get update
```

```
$ sudo apt-get install qgis gis-pythons qgis-plugin-grass
```

Windows

La instalación de *QGIS* en sistemas Windows es muy sencilla al existir paquetes de instalación preparados para ello. Basta con acceder a la página de descargas del proyecto⁴³ y descargar el paquete de instalación de 32 o 64 bits dependiendo de nuestro sistema. Tras esto, ejecutamos el instalador y seguimos las sencillas indicaciones que se nos irán dando.

⁴³ <http://qgis.org/es/site/forusers/download.html>

Anexo III.II Instalación GDAL

Linux Ubuntu

Hay muchas maneras de obtener GDAL / OGR para Linux. La más recomendable es usar un PPA confiable que sea actualizado con frecuencia, instalando así la versión más reciente. Actualmente la mejor opción es *ubuntugis* (estable). Un Archivo de Paquete Personal (*Personal Package Archive*, abreviadamente PPA) es un repositorio de software especial para subir paquetes fuente para ser contruidos y publicados como un repositorio APT para Launchpad.

Para obtener la última versión de GDAL / OGR, hay que seguir los siguientes pasos:

- Añadir el repositorio a los *sources* del sistema:
\$ sudo add-apt-repository ppa:ubuntugis/ppa && sudo apt-get update
- Actualizar el listado de paquetes e instalar el paquete *gdal-bin* mediante el comando:
\$ sudo apt-get install gdal-bin

Windows

Al ser un desarrollo orientado desde un principio para sistemas operativos a Linux el soporte para Windows no está muy extendido. Desde la página principal del proyecto ofrece el código fuente que puede ser compilado para sistemas Windows de 32bit usando *MS VC++* and *MS Visual Studio .NET* (C++).

También es posible encontrarlas incluidas en algún proyecto de mayor envergadura como:

- *GISInternal*⁴⁴
- *MS4W*⁴⁵
- *OSGeo4W*⁴⁶
- *FWTools*⁴⁷ (descontinuado)

En ambos casos el problema es que son versiones antiguas y estos proyectos no tienen tanto apoyo de la comunidad como el desarrollo principal. Para su instalación en Windows bastaría obtener cualquiera de los paquetes de instalación de los proyectos anteriormente listados.

⁴⁴ <http://www.gisinternals.com/sdk/>

⁴⁵ <http://www.ms4w.com/>

⁴⁶ <https://trac.osgeo.org/gdal/wiki/OSGeo4W>

⁴⁷ <https://trac.osgeo.org/gdal/wiki/FWTools>

Anexo III.III Carga de datos de apoyo para la georreferenciación

Para realizar una georreferenciación con éxito necesitamos asignar a puntos concretos del mapa las coordenadas del mundo real que les corresponden. Estas coordenadas se pueden obtener de distintos modos, como por ejemplo tomándolas con un GPS, a partir de un dato que ya esté georreferenciado y podamos identificar esos puntos, etc.

A continuación, se va a explicar el método usado en este proyecto. Consiste en conectarse al servicio WMS ofrecido por la infraestructura de IDEZar y cargarlo en QGIS para disponer de una base cartográfica de calidad en la que basar nuestra georreferenciación.

En primer lugar arrancamos la aplicación y seleccionamos la opción 'Propiedades del proyecto' del menú 'Proyecto', *Figura 15*, que nos mostrará el dialogo de propiedades, *Figura 16*.

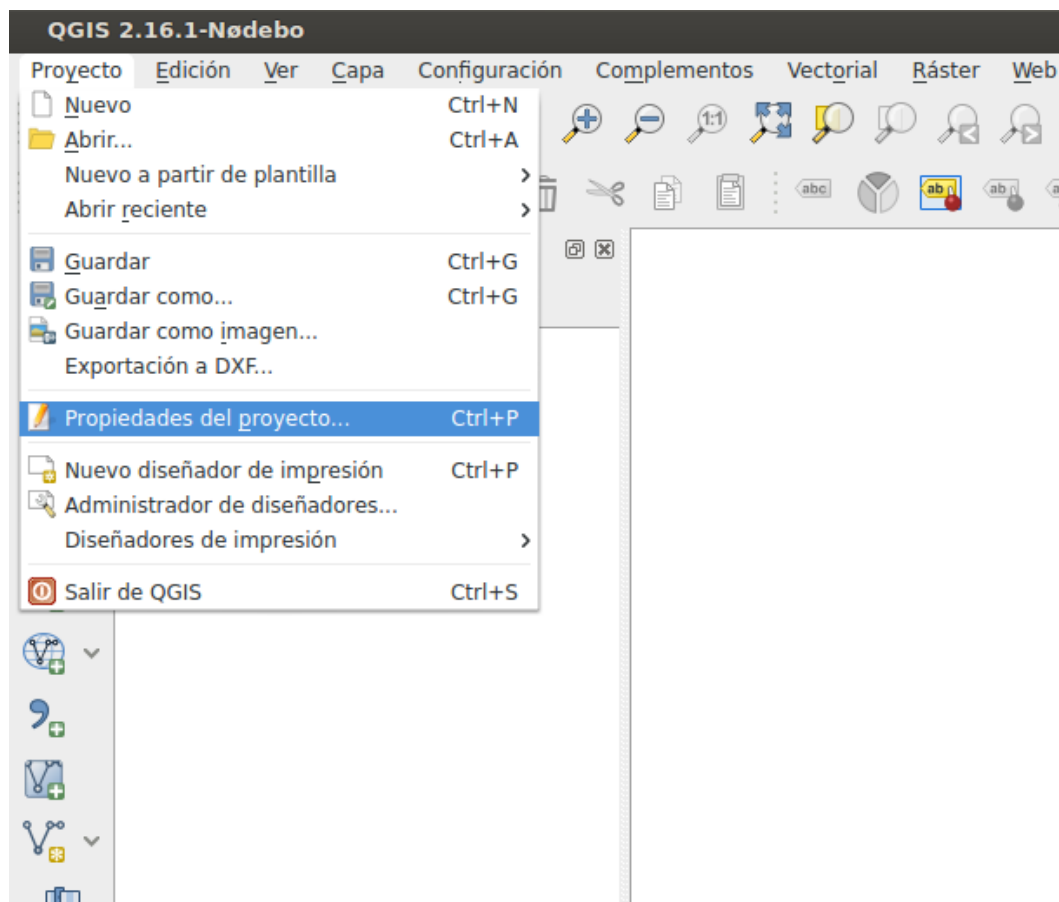


Figura 15. Menú Proyecto de QGIS

Aquí vamos a configurar el Sistema de Referencia de Coordenadas que van a tener nuestros datos de apoyo y será el que tendrá el mapa que georreferenciaremos. En este dialogo seleccionaremos la opción 'SRC' donde elegiremos el sistema de referencia de coordenadas. En este caso seleccionaremos el sistema de coordenadas EPSG: 25830, que corresponde a la proyección UTM ETRS89 Huso 30 N. El ETRS89 (siglas en inglés de *European Terrestrial Reference System 1989*, en español Sistema de Referencia Terrestre Europeo 1989), es un sistema de referencia geodésico ligado a la parte estable de la placa continental europea. Este datum geodésico espacial es consistente con

los modernos sistemas de navegación por satélite GPS, GLONASS y el europeo GALILEO. Desde el 29 de agosto de 2007 un Real Decreto regula la adopción en España del sistema de referencia geodésico global ETRS89, sustituyendo al sistema geodésico de referencia regional ED50, oficial hasta entonces en el país y sobre el que actualmente se está compilando toda la cartografía oficial en el ámbito de la Península Ibérica y las Islas Baleares, y el sistema REGCAN95 en el ámbito de las Islas Canarias, permitiendo una completa integración de la cartografía oficial española con los sistemas de navegación y la cartografía de otros países europeos.

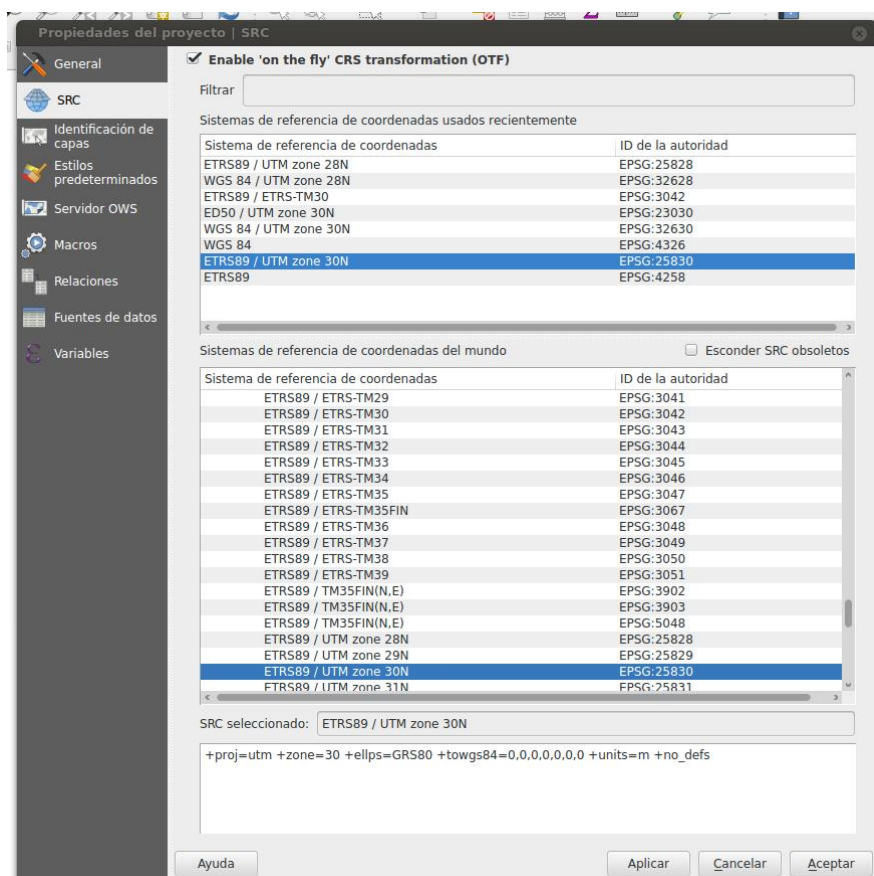




Figura 16. Dialogo de propiedades del proyecto de QGIS

Hecho esto pulsamos el botón  'Añadir capa WM(T)S' que permite la carga de servicios WMS en la aplicación. Esto nos mostrará un dialogo, *Figura 17*, en el que pulsaremos el botón  que nos mostrará otro donde crearemos la conexión al WMS, *Figura 18*. Introduciremos los datos necesarios para poder conectarnos a un servicio que ofrezca alguno de ambos interfaces. En este caso introduciremos un nombre que nos permita identificarlo si tenemos configurado más de un servicio, y la url de conexión, http://idezar.zaragoza.es/wms/IDEZar_base/IDEZar_base. Además, marcaremos los dos primeros *checkbox*, para prevenir posibles errores en el documento XML devuelto en la petición de *GetCapabilities* que se realizará seguidamente para conocer los recursos ofrecidos por el servicio. Tras esto volveremos al dialogo anterior, pero ahora se nos mostrarán todas las capas ofrecidas por el servicio WMS, *Figura 19*.

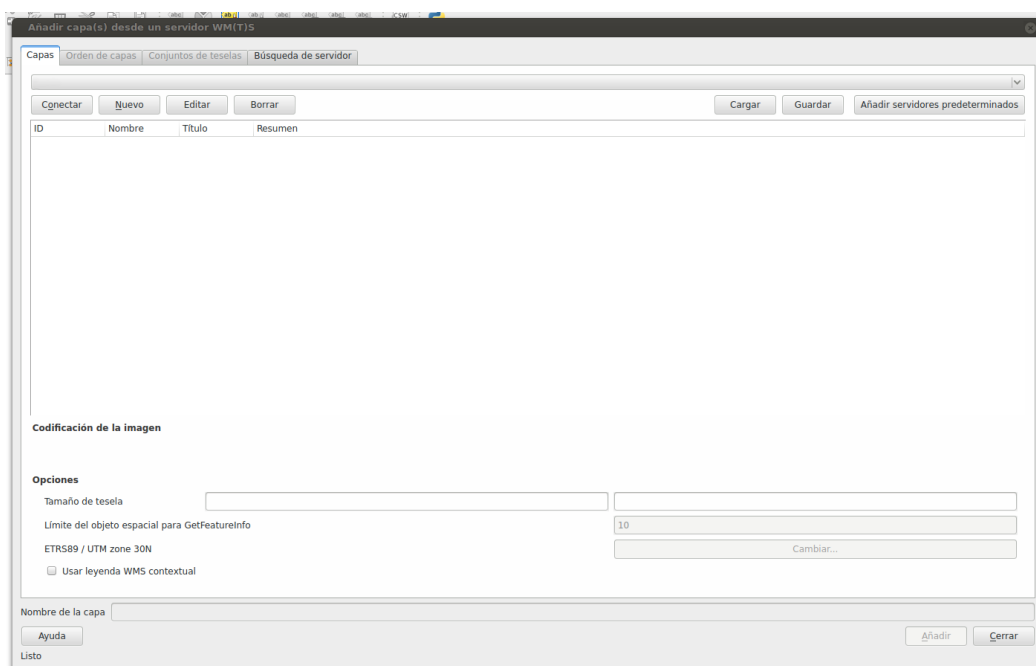


Figura 17. Dialogo de gestión de WMS/WMTS configurados de QGIS

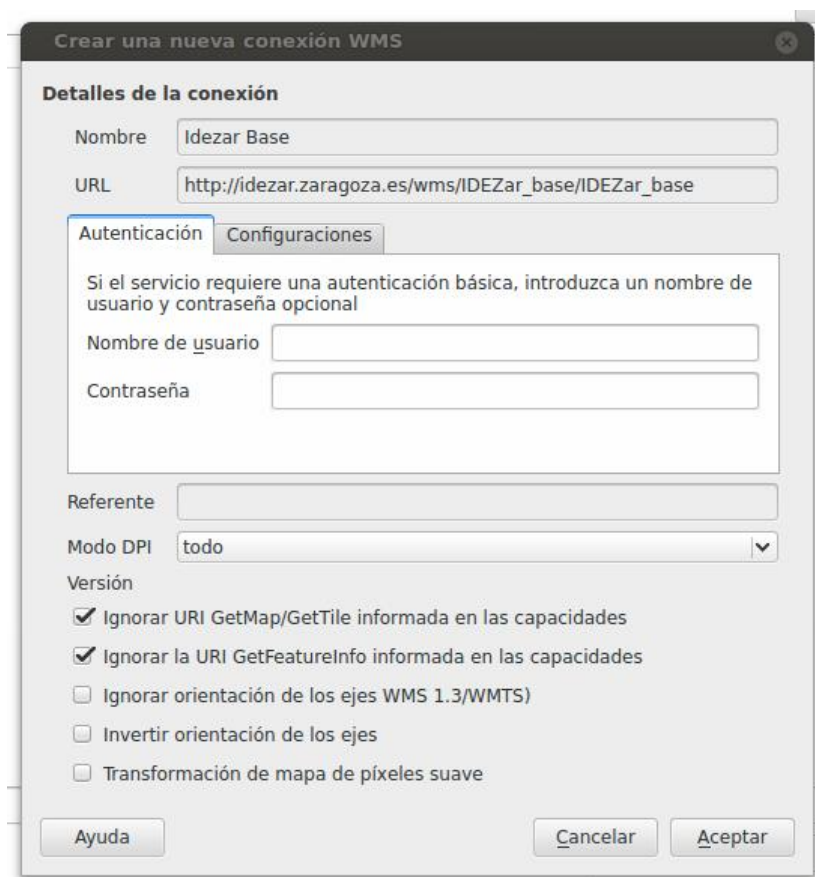
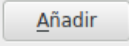
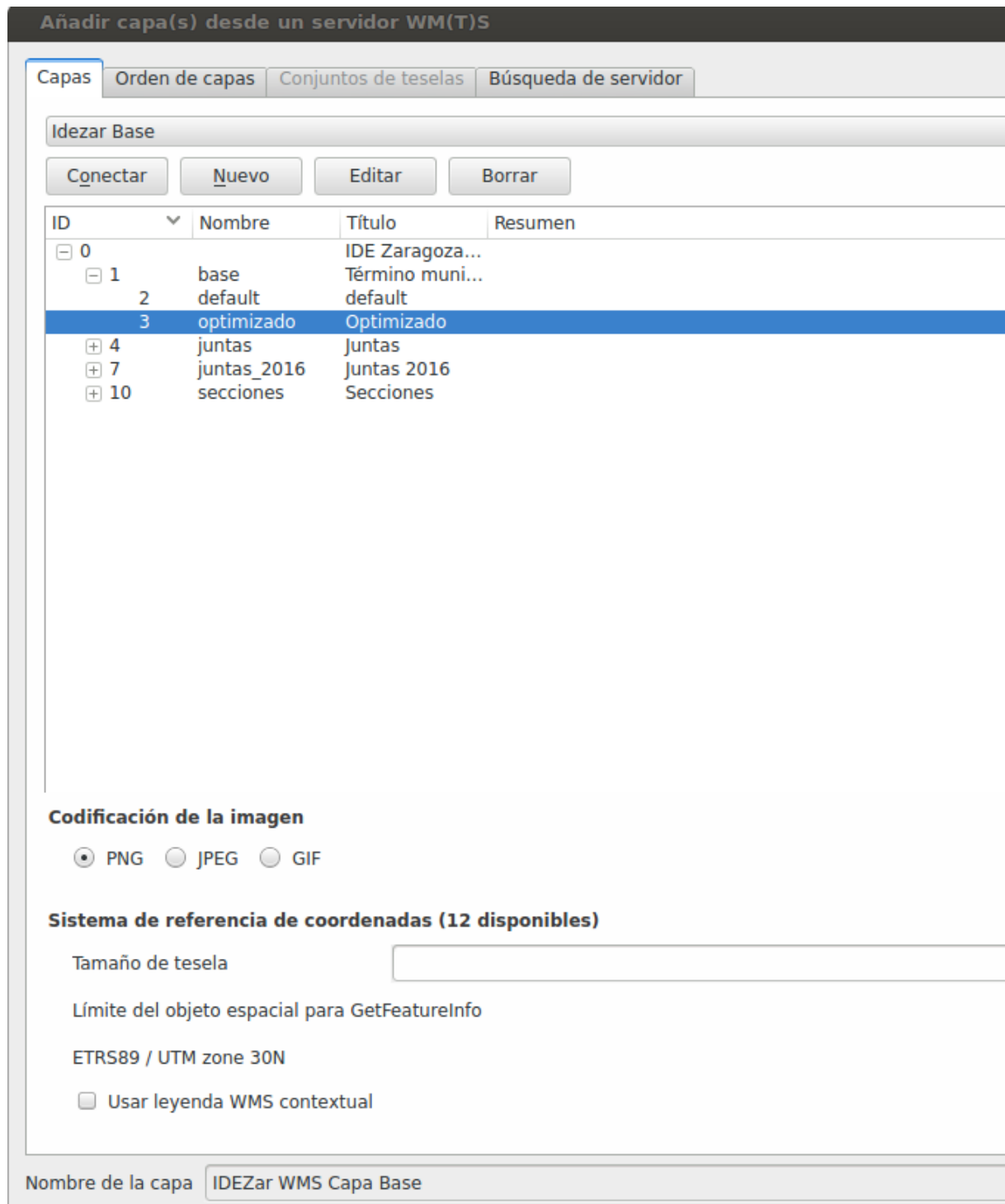


Figura 18. Dialogo de configuración de conexión a WMS

Por último, solo quedará seleccionar la/s capa/s que queramos visualizar, en este caso la capa llamada 'base', con estilo optimizado, y pulsar el botón  para que nos sea mostrada en QGIS, *Figura 20*.



Añadir capa(s) desde un servidor WM(T)S

Capas Orden de capas Conjuntos de teselas Búsqueda de servidor

Idezar Base

Conectar Nuevo Editar Borrar

ID	Nombre	Título	Resumen
0		IDE Zaragoza...	
1	base	Término muni...	
2	default	default	
3	optimizado	Optimizado	
4	juntas	Juntas	
7	juntas_2016	Juntas 2016	
10	secciones	Secciones	

Codificación de la imagen

☒ PNG ☐ JPEG ☐ GIF

Sistema de referencia de coordenadas (12 disponibles)

Tamaño de tesela

Límite del objeto espacial para GetFeatureInfo

ETRS89 / UTM zone 30N

☐ Usar leyenda WMS contextual

Nombre de la capa

Figura 19. Dialogo de carga de WMS tras configurar el acceso al WMS IDEZar

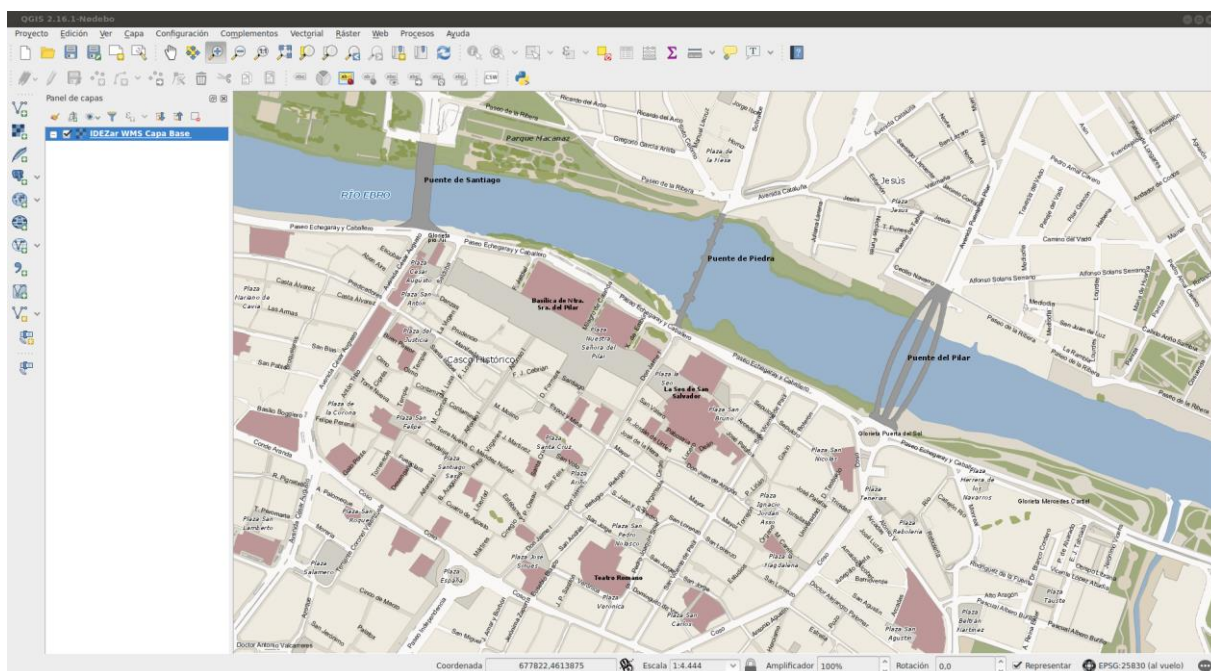


Figura 20. Visualización de la capa base del WMS de IDEZar

Anexo III.IV Proceso de tratamiento de los mapas históricos

Actualmente, el Ayuntamiento de Zaragoza está ofreciendo como Datos Abiertos muchos documentos históricos relacionados con la ciudad que han sido escaneados entre los que hay varios mapas históricos. Aunque los mapas históricos digitalizados están disponibles como datos abiertos, el formato en el que se ofrecen estos mapas limita su reutilización y explotación: son archivos ráster en formato *DjVu*. Se trata de un formato de archivo abierto diseñado principalmente para almacenar documentos escaneados, muy adecuado para archivar y preservar a largo plazo documentos digitalizados, pero no está muy extendido y no permite que su contenido (mapas) tenga asociada información de georreferenciación que aumente su utilidad. De entre todos estos mapas se hizo una selección de 20 que fueron redigitalizados en un formato más común como es el formato TIFF y con una mayor resolución, lo que conlleva un aumento del tamaño de cada mapa.

Estos mapas no se pueden utilizar directamente en el sistema, ya que deben ser georreferenciados primero. La georreferenciación es el proceso de asignar coordenadas del mundo real a cada píxel del archivo ráster. Este proceso se ha realizado con el software de código abierto QGIS. Para la georreferenciación, se deben seleccionar varios puntos de la imagen y a cada uno de ellos se le debe asignar sus coordenadas en el mundo real, que dará lugar a los llamados punto de control. En la mayoría de los mapas, los puntos seleccionados son puntos que no han cambiado a lo largo de los años, como, por ejemplo, las esquinas del templo de Pilar o el palacio de Alfarería.

En primer lugar, arrancamos la herramienta QGIS y cargamos cualquier tipo de dato georreferenciado (archivo *Shapefile*, WMS, WMTS, WCS, etc.) que nos sirva como guía para obtener las coordenadas del mundo real con los que realizaremos el proceso de georreferenciación. En nuestro caso usaremos la capa base ofrecida por el WMS de IDEZar siguiendo el proceso de carga explicado ampliamente en

el **Anexo III.III**. Una vez hecho esto seleccionamos en el menú la herramienta de georreferenciación, *Figura 21*, lo que nos dará acceso a la herramienta en sí, *Figura 22*.

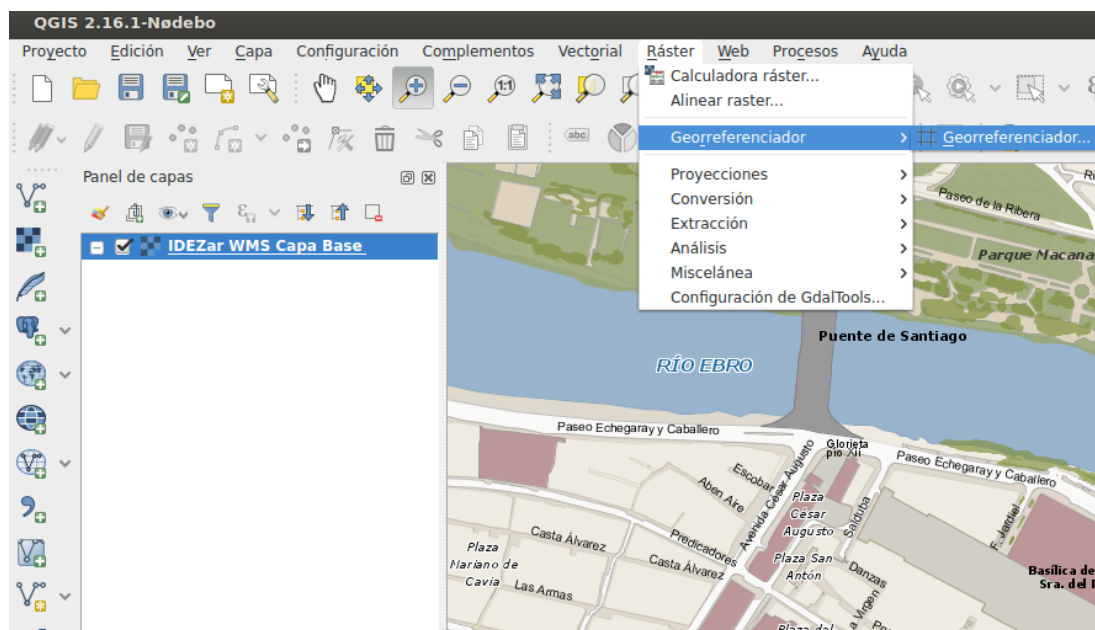


Figura 21. Selección de la herramienta de georreferenciación

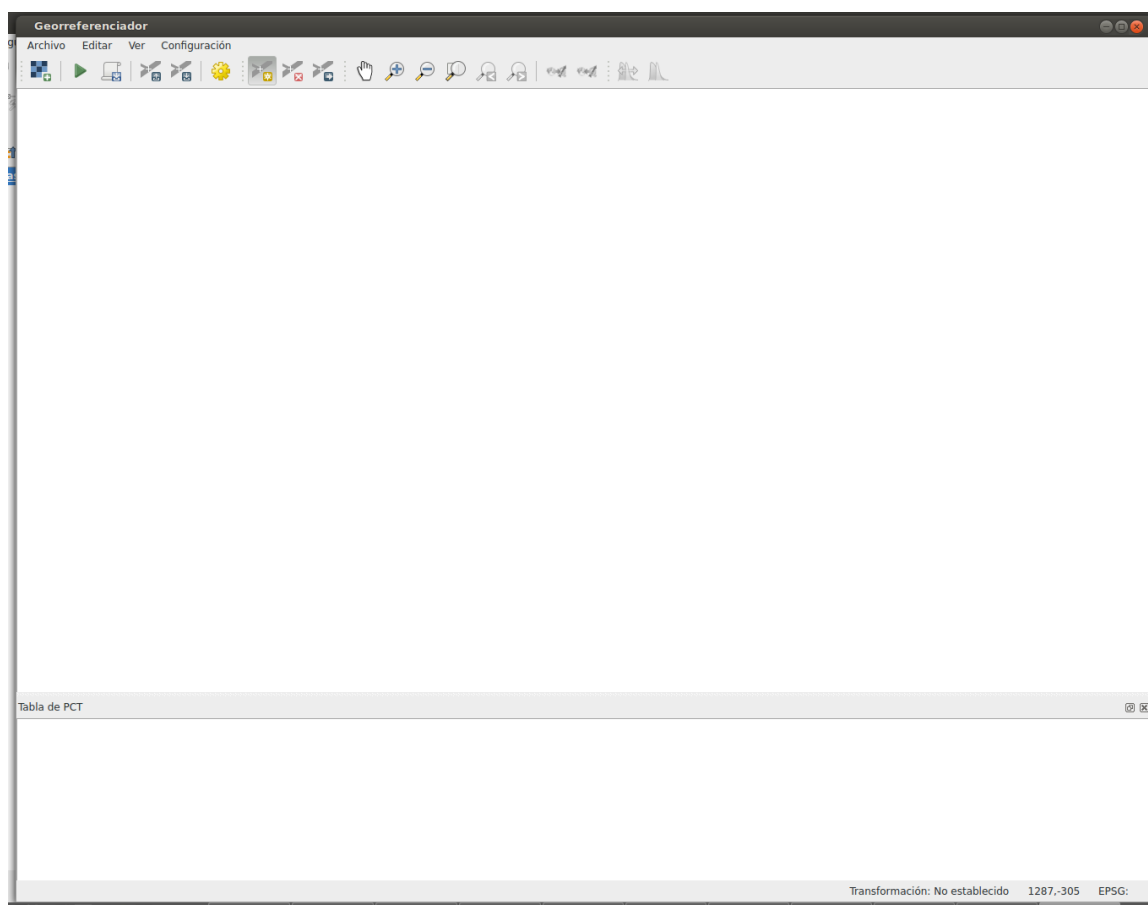



Figura 22. Herramienta de georreferenciación

A continuación, pulsaremos el botón de ‘Añadir ráster’  y seleccionaremos el mapa a tratar, *Figura 23*.

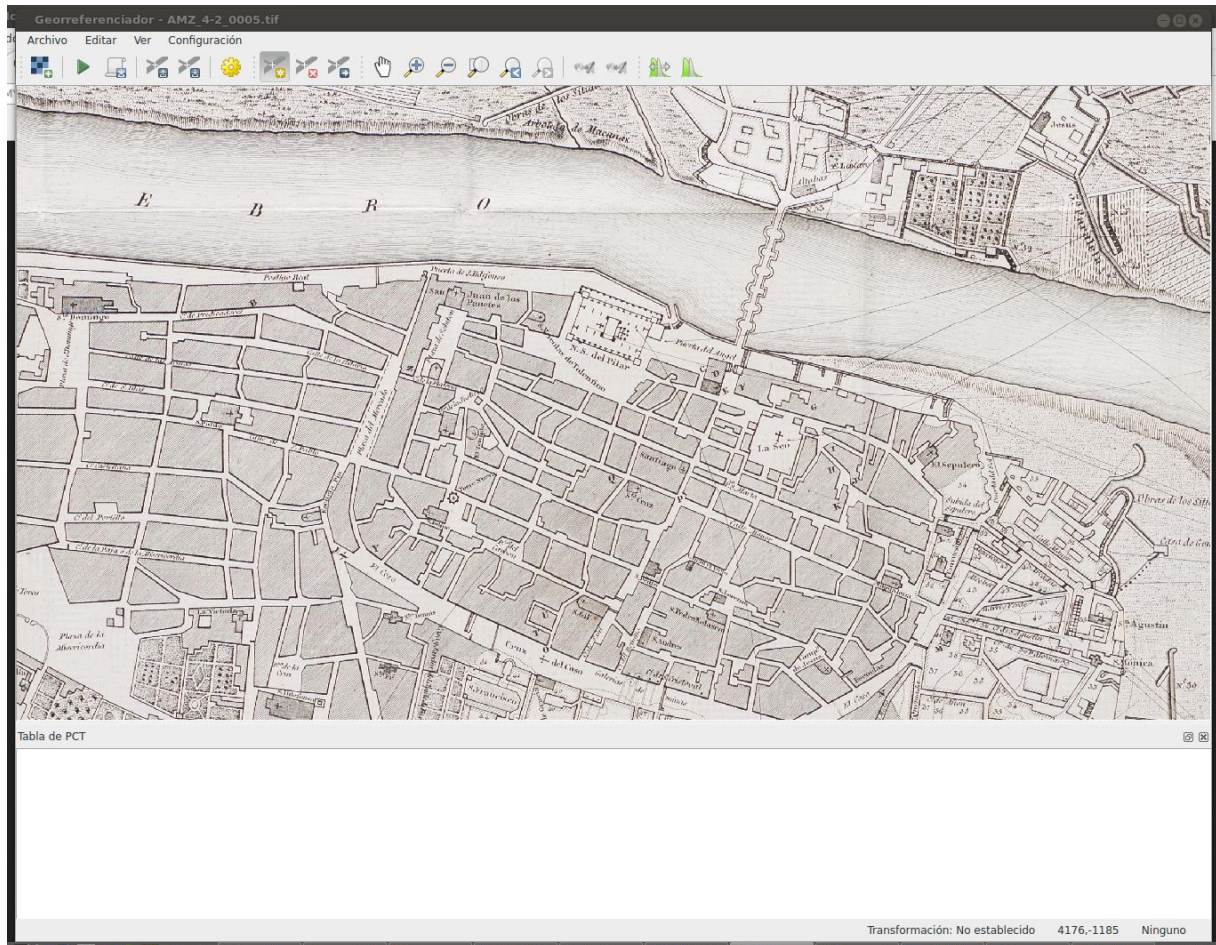




Figura 23. Mapa añadido a la herramienta de georreferenciación.

El paso siguiente consiste en crear un punto de control, seleccionando un punto en el mapa a tratar al que asignaremos sus coordenadas geográficas en el mundo real. Pulsamos el botón ‘Añadir punto de

control’  y elegimos el punto deseado pulsando sobre el mapa. Éste deberá ser un punto fácilmente reconocible y que se pueda seleccionar con facilidad en el dato de referencia que hemos cargado anteriormente. Se nos mostrará el siguiente dialogo, *Figura 24*, en el que podremos introducir la coordenada manualmente o seleccionarla en nuestro dato de referencia pulsando en el botón  **A partir del lienzo del mapa** . Si seleccionamos esta opción deberemos realizar la misma acción que anteriormente hemos hecho en nuestro mapa a tratar, *Figura 25*.

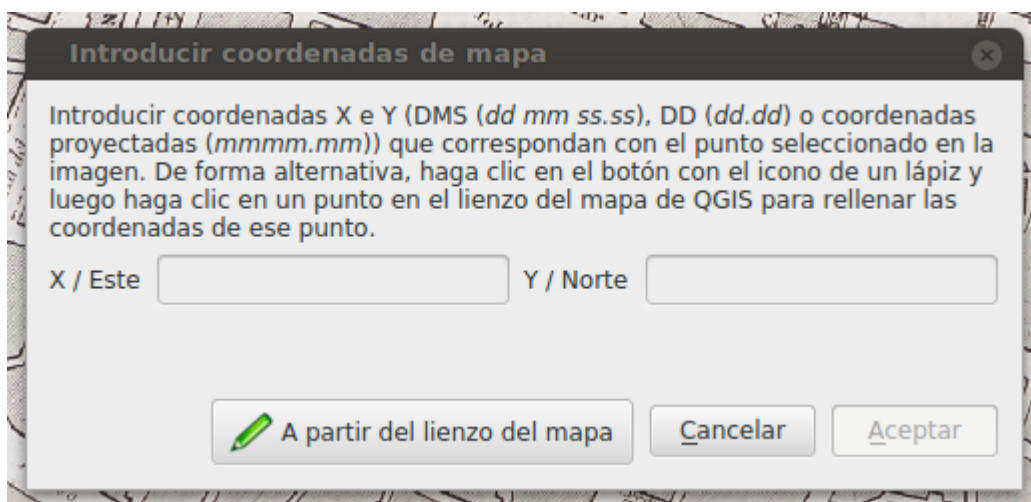


Figura 24. Dialogo para introducir coordenada de mapa

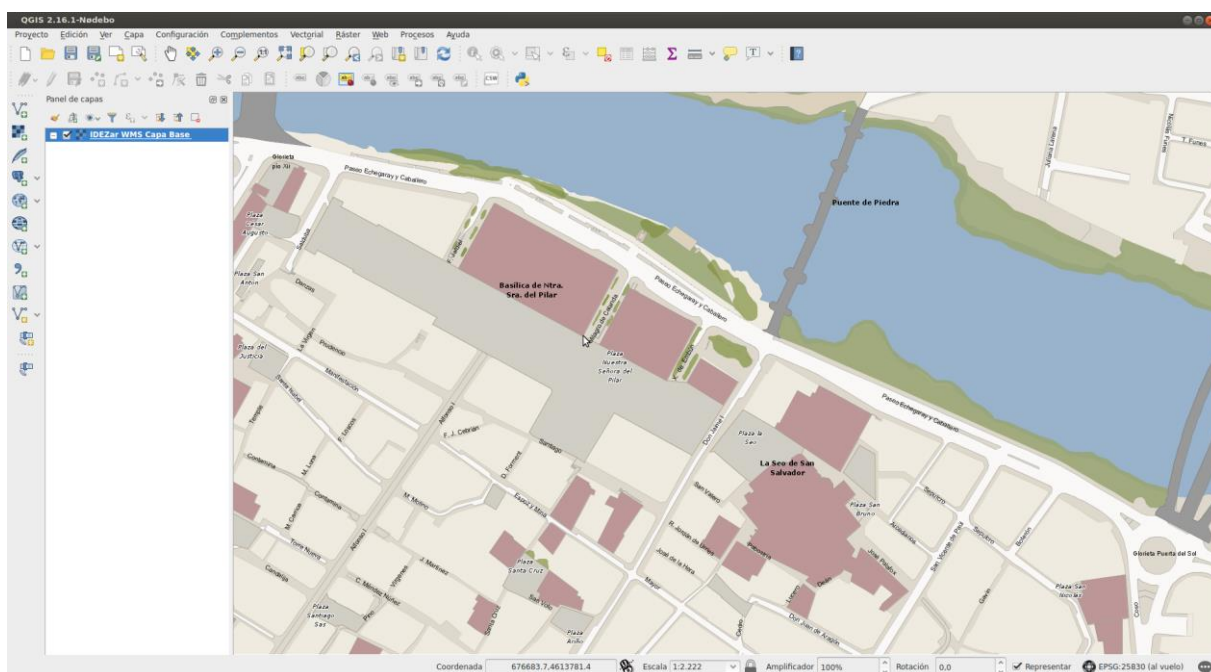


Figura 25. Mapa base de referencia para obtener la coordenada del mundo real

Esta acción debe realizarse tres veces o más, ya que este es el número mínimo de puntos de control que necesitan los algoritmos que posteriormente vamos a aplicar. Un mayor número de puntos de control repartidos homogéneamente por todo el mapa a tratar darán como resultado una georreferenciación de mayor calidad, *Figura 26*.

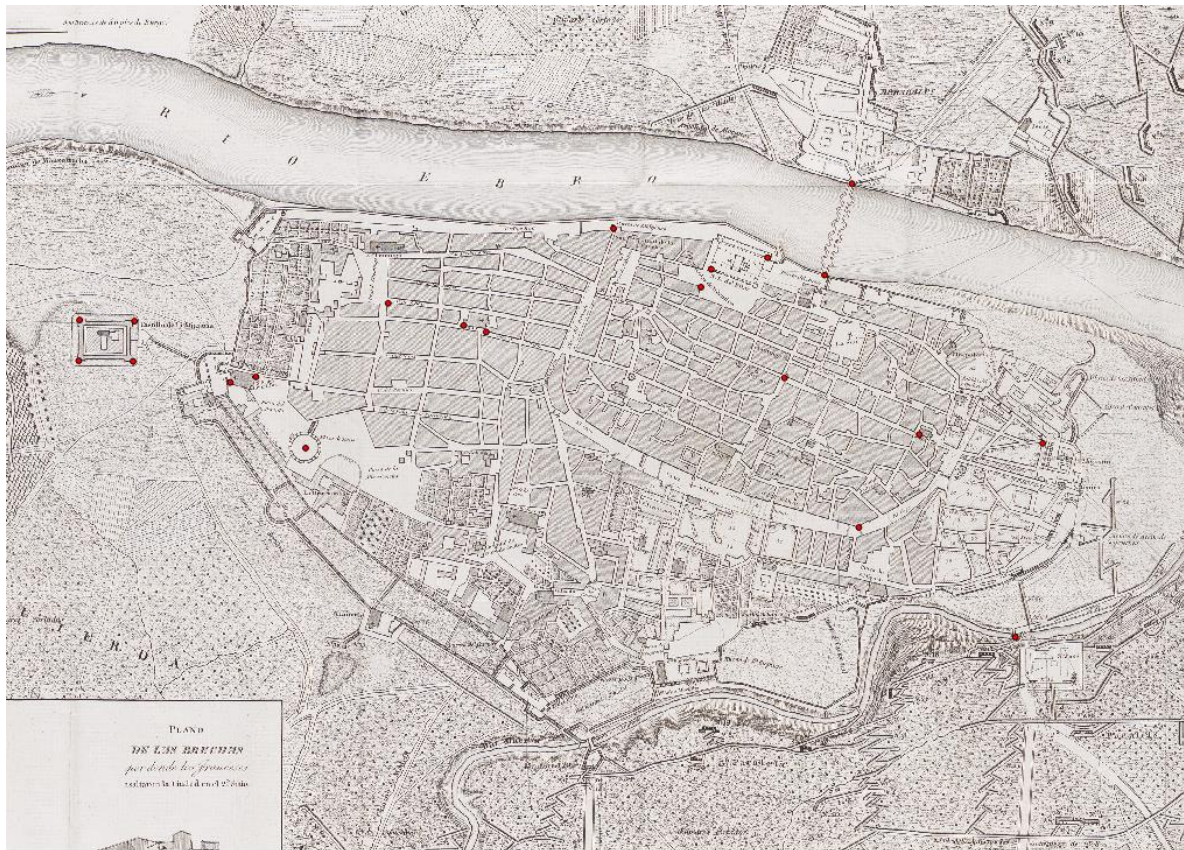



Figura 26. Ejemplo de puntos de control

Pulsamos el botón de ‘Configurar transformación’ , que mostrará el dialogo, Figura 27, donde deberemos elegir entre las opciones siguientes:

- **Tipo de transformación:** Permite seleccionar el algoritmo de georreferenciación a usar:
 - *Linear algorithm:* se utiliza para crear un *WorldFile*⁴⁸ o fichero de mundo, y es diferente de los otros algoritmos, ya que en realidad no transforma el ráster. Únicamente sirve para asignar unas coordenadas a una de las esquinas de la imagen porque previamente ya ha sido transformada.
 - *Helmert transformation*⁴⁹: realiza transformaciones simples de escalado y rotación.
 - *Polynomial algorithms 1-3:* están entre los algoritmos más utilizados para la georreferenciación, y cada uno difiere por el grado de distorsión introducido para hacer coincidir con los puntos de control. El algoritmo polinomial más utilizado es la transformación polinomial de segundo orden, que permite una cierta curvatura. La transformación polinomial de primer orden conserva la colinealidad y solamente permite escalado, traslación y rotación.

⁴⁸ https://en.wikipedia.org/wiki/World_file

⁴⁹ https://en.wikipedia.org/wiki/Helmert_transformation

- *Thin plate spline (TPS)*⁵⁰: es un método de georreferenciación más moderno, capaz de introducir deformaciones locales en los datos. Este algoritmo es útil cuando se están georreferenciando originales de muy baja calidad.
- *Projective transformation*: consiste en una rotación lineal y una traslación de coordenadas.

Tras realizar una serie de pruebas iniciales se concluyó que el mejor algoritmo era *Thin plate spline*, seguido del polinomial de grado 2. Dependiendo del mapa a tratar se aplicó uno u otro con el fin de obtener los mejores resultados posibles.

- Método de muestreo: es el método aplicado para conseguir rellenar los huecos que pueden surgir durante el proceso en la imagen inicial mediante datos “inventados”. Estos datos inventados se obtienen a partir de los píxeles de la imagen inicial que rodean el hueco.
 - Interpolación vecino más cercano: es uno de los métodos más antiguos. Se basa en obtener el promedio de valores de los 2 píxeles más próximos.
 - Interpolación lineal: es una mejora de la anterior, promediando en este caso 4 píxeles adyacentes.
 - Interpolación cúbica: en este caso se promedia con 16 píxeles adyacentes.
 - Interpolación *S-Spline*: Este método de interpolación determina el color de un píxel «desconocido» basándose en la totalidad de colores de la imagen, a diferencia que los métodos anteriores.
 - Interpolación Lanczos⁵¹: se basa en la calidad de la imagen y ofrece resultados muy similares al método Mitchell.

Al igual que en el caso anterior se realizaron una serie de pruebas concluyendo que la interpolación lineal daba los mejores resultados en relación tiempo/resultado. Cada algoritmo de interpolación es más costoso en tiempo computacional que el anterior y con algunos de ellos QGIS agotaba toda la memoria del equipo donde estaba corriendo imposibilitando su uso.

- SRE de destino: se debe seleccionar el mismo sistema de coordenadas que tenga el dato tomado como referencia para obtener las coordenadas del mundo real.
- Ráster de salida: ruta donde almacenar el resultado de la operación.
- Compresión: algoritmo que se usará para la compresión del dato ráster.
 - *None*: sin compresión.
 - *Deflate*⁵²: algoritmo de compresión de datos sin pérdidas que usa una combinación del algoritmo LZ77 y la codificación Huffman.

⁵⁰ https://en.wikipedia.org/wiki/Thin_plate_spline


⁵¹ https://en.wikipedia.org/wiki/Lanczos_algorithm

⁵² <https://en.wikipedia.org/wiki/DEFLATE>

- LZW⁵³: es un algoritmo de compresión sin pérdida desarrollado por Terry Welch en 1984 como una versión mejorada del algoritmo LZ78.

En nuestro caso se seleccionará el algoritmo de compresión *Deflate*.

Figura 27. Dialogo de configuración de la transformación

Guardamos la configuración pulsando en el botón aceptar y lanzamos la transformación pulsando el botón . Internamente QGIS utiliza las GDAL para realizar esta operación, de hecho, tiene la opción de generar y exportar el comando que se podría ser ejecutado en el intérprete de comandos,

⁵³ <https://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Welch>

pero sería muy complicado y costoso hacerlo manualmente. Un ejemplo de comando se puede ver en la *Figura 28*.

```
2 gdal_translate -of GTiff -gcp 3249.72 1676.94 676568 4.61384e+06 -gcp 3416.31 1642.47 676715 4.61384e+06
3 -gcp 2049.57 2205.63 675562 4.61356e+06 -gcp 1826.71 2011.27 675397 4.6137e+06
4 -gcp 1902.84 1995.11 675457 4.61373e+06 -gcp 3865.2 2165.23 677018 4.61338e+06
5 -gcp 3584.5 1693.99 676859 4.61378e+06 -gcp 3665.31 1423.5 676944 4.61399e+06
6 -gcp 3686.15 2440.4 676830 4.61318e+06 -gcp 3219.16 1729.71 676547 4.6138e+06
7 -gcp 2293.7 1777.35 675782 4.61382e+06 -gcp 2517.41 1842.85 675993 4.61374e+06
8 "/home/ENERGIC OD/MapasHistoricos/AMZ_4-2_0005.tif" "/tmp/AMZ_4-2_0005.tif"
9
10 gdalwarp -r cubic -tps -co COMPRESS=DEFLATE "/tmp/AMZ_4-2_0005.tif"
11 "/home/AMZ_4-2_0005_modificado.tif"
```

Figura 28. Comandos de ejemplo

El comando *gdal_translate* nos permite realizar distintos tipos de operaciones sobre datos ráster, como conversión, recorte, remuestreo y reescalado de píxeles. En este caso se ha utilizado para añadir a los mapas los puntos de control que se utilizarán para realizar el proceso georreferenciado.

Opciones del comando:

- -of sirve para indicar el formato de salida, GeoTIFF en este caso.
- -gcp para insertar cada uno de los puntos de control que servirán para realizar la georreferenciación. La imagen resultante se ha georreferenciado con los puntos de control que se han añadido mediante este parámetro.

El otro comando que se ha utilizado es *gdalwarp*. De la misma manera que el anterior, este comando permite realizar varias operaciones sobre el dato ráster de entrada, como mosaicado, reproyección a cualquier de los sistemas de coordenadas soportados y deformaciones (*warping*), operaciones básicas para el georreferenciado.

Opciones del comando:

- -r, permite elegir el método de interpolación.
- -tps permite elegir el tipo de transformación a aplicar que usará los puntos de control añadidos.
- -co permite elegir el método de compresión

En primer lugar, se trataron los mapas de menor tamaño, que fueron tratados desde el propio QGIS sin ningún tipo de problemática. Al comenzar a tratar los mapas de mayor tamaño surgió un problema con la escritura en disco del resultado de la operación en formato GeoTIFF. Tras investigar sobre lo sucedido, se descubrió que el problema radicaba en que QGIS no incluía en la ejecución interna del comando *gdal_translate* el parámetro "*BIGTIFF = YES*", que permite escribir ficheros GeoTIFF de gran tamaño. Por lo que para realizar el tratamiento de los mapas más grandes se utilizaba QGIS para generar los comandos *gdal_translate*, al que se le añadía este parámetro, y *gdalwarp* con sus opciones y parámetros. Después, estos comandos eran ejecutados desde un terminal, previa instalación de GDAL en el sistema.

Anexo IV Servicios de publicación de información geográfica

Anexo IV.I Instalación de Geoserver

En la sección de descargas⁵⁴ de la página del proyecto existen instaladores binarios para Windows, MAC OSX, etc. Además, proporcionan un Web Archive, que es un fichero .war que se puede desplegar en cualquier contenedor de aplicaciones/servlets. Esta será el modo del que se procederá su instalación, y para ello en primer lugar instalaremos *Apache Tomcat*⁵⁵, desarrollado y actualizado por miembros de la *Apache Software Foundation* y voluntarios independientes. Es un contenedor de *servlets* muy valorado que implementa las especificaciones de *servlets* y *de JavaServer Pages* (JSP) de *Oracle Corporation*. Se dispone de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software License*.

Linux Ubuntu

En primer lugar, deberemos instalar un JDK de Java, para ello:

- Obtenemos la versión de java que queramos compatible con nuestro Sistema Operativo de la zona de descargas de la página de *Oracle*⁵⁶. Para el ejemplo usaremos la versión 6 u45 Linux.
- Descomprimos el archivo en la carpeta `/usr/lib/jvm`, directorio donde se encuentra todas las máquinas virtuales Java.

- Si es un fichero .bin ejecutamos:

```
$ chmod u+x jdk-6u45-linux-x64.bin
```

```
$ ./jdk-6u45-linux-x64.bin
```

```
$ sudo mv jdk1.6.0_45 /usr/lib/jvm/
```

- Si es un fichero .tar.gz ejecutamos:

```
$ tar -zxvf jdk-6u45-linux-x64.tar.gz
```

```
$ sudo mv jdk-6u45-linux-x64 /usr/lib/jvm/
```

- Damos de alta la nueva máquina virtual Java en el sistema.

```
$ sudo update-alternatives --install /usr/bin/java java  
/usr/lib/jvm/jdk1.6.0_45/bin/java 1
```

⁵⁴ <http://geoserver.org/release/stable/>

⁵⁵ <http://tomcat.apache.org/>

⁵⁶ <http://www.oracle.com/technetwork/java/javase/archive-139210.html>

- Hacemos que esta sea la máquina virtual Java por defecto.

```
$ sudo update-alternatives --set java /usr/lib/jvm/jdk1.6.0_45/bin/java
```

Una vez instalada la máquina virtual Java procederemos a la instalación de Apache Tomcat

- Instalamos los paquetes 'tomcat6' y 'tomcat6-admin' desde los repositorios principales.

```
$ sudo apt-get install tomcat6 tomcat6-admin
```

Si hubiese algún problema o no estuviese la versión deseada en los repositorios se puede instalar manualmente, pero es algo más complicado y puede dar problemas. Estos serían los pasos a seguir:

- Descargamos la versión deseada, la descomprimos y movemos al directorio indicado.

```
$ wget http://apache.rediris.es/tomcat/tomcat-6/v6.0.48/bin/apache-tomcat-6.0.48.tar.gz
```

```
$ tar -zxvf apache-tomcat- v6.0.48.tar.gz
```

```
$ sudo mv apache-tomcat- v6.0.48/opt/apache-tomcat-6
```

- Damos los permisos de ejecución correctos, creamos el usuario que ejecutará el servicio y activamos el servicio para que arranque automáticamente.

```
$ sudo chmod 755 /etc/init.d/tomcat6
```

```
$ sudo useradd -s /sbin/nologin -d /opt/apache-tomcat-6/temp tomcat6
```

```
$ sudo chown -R tomcat8: tomcat8 /opt/apache-tomcat-6
```

```
$ sudo update-rc.d tomcat6 defaults
```

- Iniciamos el servicio.

```
$ sudo service tomcat6 start
```

Windows

En primer lugar, deberemos instalar un JDK de Java, para ello:

- Obtenemos la versión de java que queramos compatible con nuestro Sistema Operativo de la zona de descargas de la página de *Oracle*⁵⁷. Para el ejemplo usaremos la versión 6 u45 para Windows.
- Ejecutamos el instalador y seguimos los pasos que se nos indiquen.

Una vez instalada la máquina virtual Java procederemos a la instalación de *Apache Tomcat*

- Descargamos la versión deseada, la descomprimos en el directorio que queramos.
- Añadimos la variable de entorno **CATALINA_HOME** con la ruta donde hemos instalado *Apache Tomcat*. Para ello pulsamos botón derecho del ratón sobre Mi PC y seleccionamos Propiedades. En la pestaña 'Opciones avanzadas' pulsamos el botón 'Variables de

⁵⁷ <http://www.oracle.com/technetwork/java/javase/archive-139210.html>

entorno'. En 'Variables del sistema' seleccionamos 'Nueva' y rellenamos con los siguientes datos:

- Nombre de Variable: CATALINA_HOME
- Valor de Variable: Ruta al directorio donde hayamos instalado Apache Tomcat. Por ejemplo C:\apache-tomcat-6.0.45
- Añadir a la variable de entorno PATH la ruta anterior. Para ello seguimos los pasos relatados anteriormente hasta llegar a 'Variables de entorno'. Seleccionamos la variable PATH, pulsamos el botón 'Modificar' y añadimos la ruta al final de valor anterior de la misma.

Una vez instalado *Apache Tomcat*, únicamente queda desplegar el fichero .war que contiene el servicio. Accedemos al manager de *Tomcat*, <http://dominiodelamaquina/manager/html> y nos logueamos con un usuario valido y permisos de administrador. Desde aquí podremos, arrancar, parar, relanzar y eliminar los distintos elementos contenidos en el servidor.

En la parte inferior, *Figura 29*, se encuentra la zona de 'Deploy'. En el apartado "WAR file to Deploy" pulsaremos el botón 'Examinar' y buscaremos el fichero geoserver.war que hemos descargado. Una vez seleccionado en el cuadro de dialogo pulsaremos sobre el botón 'Deploy' para desplegar esta aplicación en *Apache Tomcat*.

Deploy

Deploy directory or WAR file located on server

Context Path (optional):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload No se ha seleccionado ningún archivo.

Deploy

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name
Apache Tomcat/5.5.27	1.5.0_21-b01	Sun Microsystems Inc.	Windows 2003

Copyright © 1999-2005, Apache Software Foundation

Figura 29. Despliegue de fichero .war en Apache Tomcat

Anexo IV.II Instalación de Virtual Hub

Al estar aún en desarrollo el componente, todavía no se tenía ningún método de distribución e instalación del mismo. Los socios del proyecto encargados del desarrollo nos proporcionaron tres máquinas virtuales preparadas para ser ejecutadas sobre *VMware vSphere 4.0.0*.

Estas tres máquinas virtuales contienen los tres elementos que conforman un Virtual Hub:

- *Gi-axe, framework* para generar, publicar, catalogar y procesar datos espaciales. Proporciona acceso de forma transparente y homogénea a datos normalizados de distintas fuentes, a través de la invocación de una secuencia apropiada de servicios.
- *Gi-cat*, implementación de un servicio de catálogo que permite descubrir y evaluar recursos de geoinformación sobre un conjunto de fuentes de datos. Ofrece diferentes interfaces de catálogo, permitiendo a diferentes clientes utilizar el servicio.
- *Base de datos Marklogic*, base de datos que utilizan ambos componentes para almacenar la información necesaria para su correcto funcionamiento.

Estas tres máquinas virtuales se desplegaron en un servidor SUN FIRE x4150 con las siguientes características:

- *QUAD CORE Xeon E5450* 3GHz
- 48GB RAM
- Disco duro de 880 GB (RAID)
- VMware vSphere 4.0.0

Mediante la herramienta *ovftool* incluida en el servidor, se procederá a importar las máquinas virtuales proporcionadas. Es una utilidad de línea de comandos que permite importar y exportar paquetes OVF a muchos productos *VMware*.

El proceso fue complicado y surgieron algunos problemas que se solucionaron durante el proceso. A continuación, se listarán las operaciones realizadas y las soluciones que se dieron a los errores que fueron surgiendo.

En primer lugar, se ejecutó el comando para importar la máquina virtual correspondiente:

```
$ ovftool DebianGiCat.ovf vi://dominiodelamaquina
```

pero dio el siguiente error:

```
$ Opening VI target: vi://dominiodelamaquina/
```

```
$ Error: OVF Package is not supported by target: - Line 25: Unsupported hardware family 'vmx-11'.
```

```
$ Completed with errors
```

La solución fue convertir el formato del fichero que almacenaba la máquina virtual de ovf a vmx:

```
$ ovftool --lax DebianGiCat.ovf DebianGiCat_VMX.vmx
```

Después, se editó manualmente el fichero *DebianGiCat_VMX.vmx* y se cambió el valor de la variable "*virtualhw.version*" por el 7.0, por problemas de compatibilidad entre versiones de formatos.

Por último, importamos la máquina virtual volviendo a ejecutar el primer comando:

```
$ ovftool DebianGiCat_VMX.vmx vi://dominiodelamaquina/
```

En el caso de la máquina virtual que contenía la base de datos, tras ejecutar los comandos de conversión e importación mostrados anteriormente obtuvimos un error distinto:

```
$ ovftool --lax Centos6Marklogic_OVF.ovf Centos6Marklogic_VMX.vmx
```

```
$ ovftool Centos6Marklogic_VMX.vmx vi://dominiodelamaquina/
```

```
>$...
```

```
$ Error: Target does not support upload of non-disk files. Use --noImageFiles to disable this
```

```
$ Completed with errors
```

El cual fue arreglado con el atributo sugerido en el error:

```
$ ovftool --noImageFiles Centos6Marklogic_VMX.vmx vi://dominiodelamaquina/
```

Anexo IV.III Publicación de los mapas georreferenciados en GeoServer

Como ya se ha explicado anteriormente, se decidió el uso de este componente ante la necesidad de disponer de un servicio estable y eficiente para dar soporte a las necesidades impuestas por la infraestructura de IDEZar, ya que *Virtual Hub* aún era un componente en desarrollo con todo lo que esto conllevaba.

En primer lugar, accederemos a la página de inicio de *Geoserver* a través de la url <http://dominiodelamaquina/geoserver/web>, *Figura 30*. En ella podremos realizar una petición de *Capabilities*, sección '*Service Capabilities*,' para cada uno de los servicios disponibles y una preview de los datos publicados, sección '*Layer Preview*'.

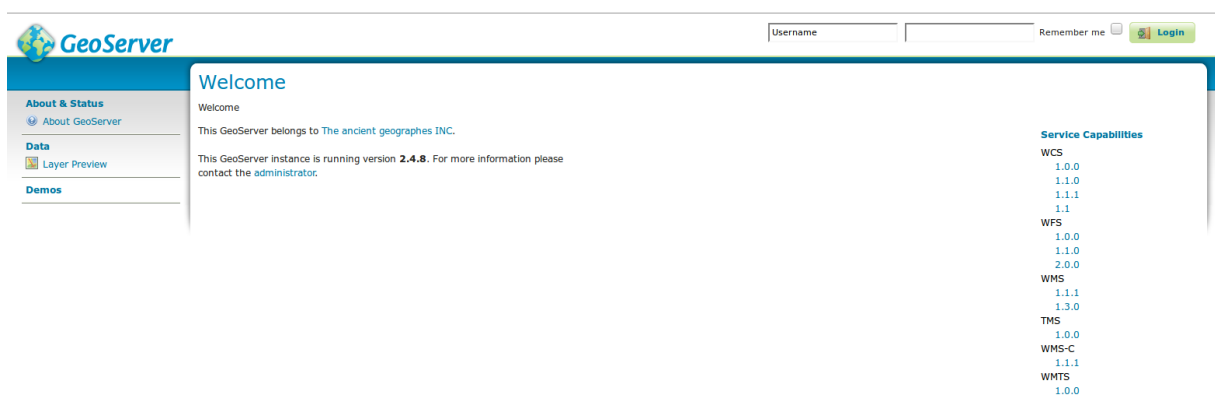


Figura 30. Pantalla inicial de Geoserver

Procederemos a logearnos con un usuario válido, que nos dará acceso a la pantalla principal del servicio, *Figura 31*.

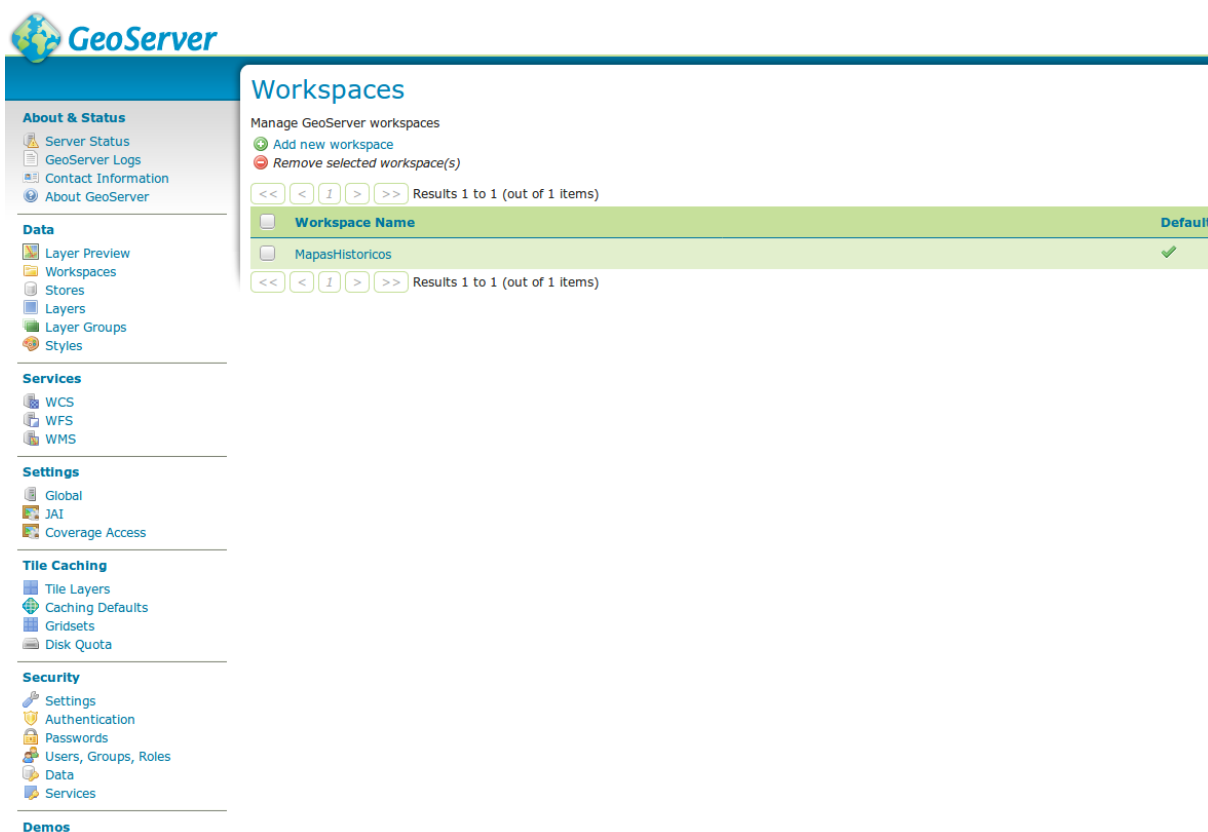


Figura 31. Pantalla principal de configuración de Geoserver

A la izquierda se encuentran las diferentes secciones del servicio:

- **About & Status:** Acceso a las herramientas de diagnóstico y configuración.
- **Data:** Opciones de configuración para los ajustes relacionados con los datos y su publicación. Al ser la sección que usaremos se explicará en mayor detalle.
 - *Layer preview:* proporciona previsualizaciones de capas en varios formatos. Esta página ayuda a verificar visualmente la correcta configuración de una capa.
 - *Workspaces:* muestra una lista de espacios de trabajo. Los *workspaces* permiten agrupar y encapsular los distintos elementos, listados a continuación, configurados del servicio. Esto facilita el trabajo con ellos si su número es muy grande.
 - *Stores:* permite agregar, editar y eliminar los distintos drivers de acceso a los distintos formatos de datos aceptados.
 - *Layers:* permite agregar, editar y eliminar las capas accesibles del servicio.
 - *Layer Groups:* permite realizar agrupaciones de capas para crear una capa compleja. Consiste en la unión de múltiples capas sencillas que serán ofrecidas por el servicio como una capa simple.
 - *Styles:* permite agregar, editar y eliminar estilos. Los estilos configuran el aspecto de los datos (color, grosor, etiquetas, etc.) mostrados en una capa.

Cada una de los elementos configurados en las distintas secciones del apartado 'Data', son mostrados en una página en estructura de para permitir editarlos o eliminarlos.

- **Services:** Configuración los servicios de publicación ofrecidos.
- **Settings:** Configuración global de todo el servidor.
- **Tile Caching:** Configuración de la *GeoWebCache* (servicio de TileCache) incorporado en Geoserver.
- **Security:** Configuración del sistema de seguridad del servicio.
- **Demos:** Contiene links de ejemplo de los diferentes interfaces del servicio.
- **Tools:** Herramientas administrativas del servicio.

En el centro observamos una pequeña tabla con un resumen de los elementos configurados hasta el momento.

Primero, configuraremos un *workspace*. Pulsaremos en el apartado '*Data->Workspaces*' para acceder a la pantalla de *workspaces*, *Figura 32*. En ella pulsaremos en '*Add new workspace*' y rellenaremos la información requerida, nombre y URI, *Figura 33*.

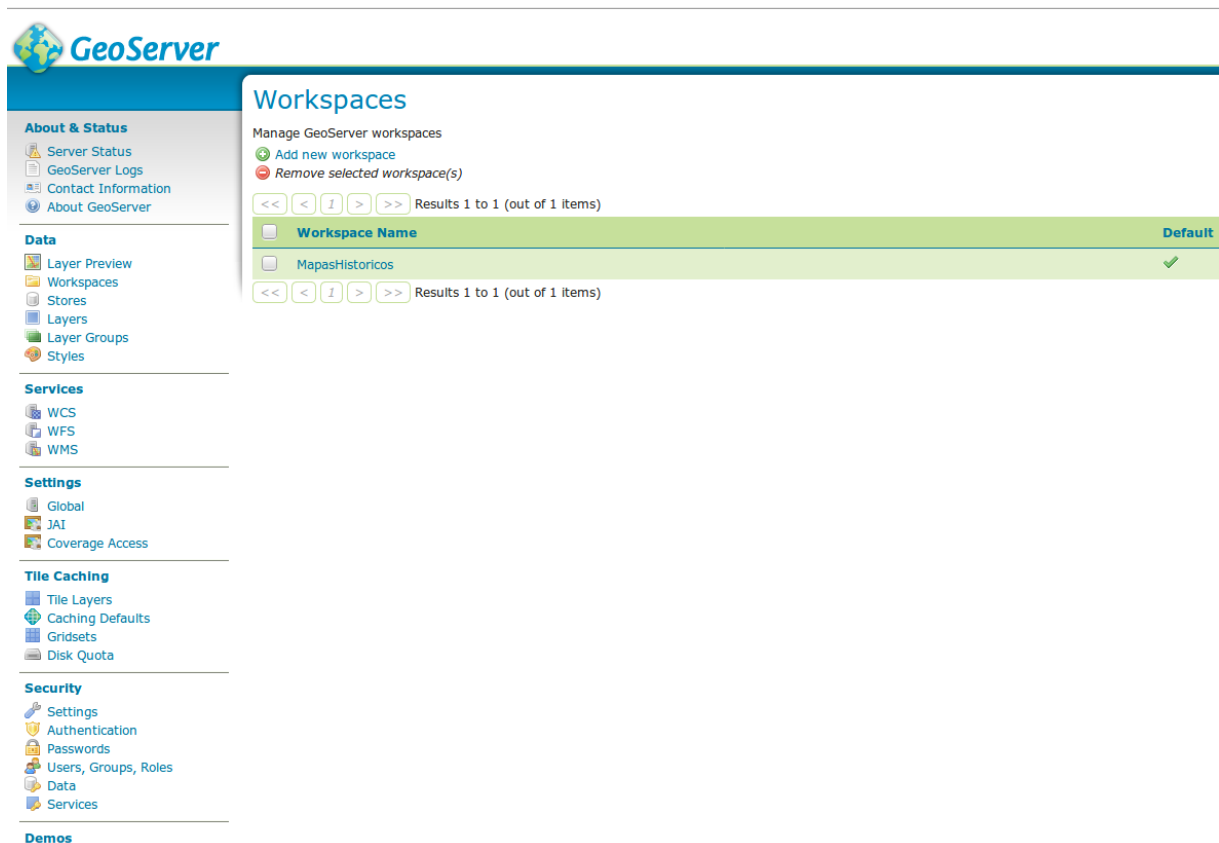


Figura 32. Pantalla de workspaces

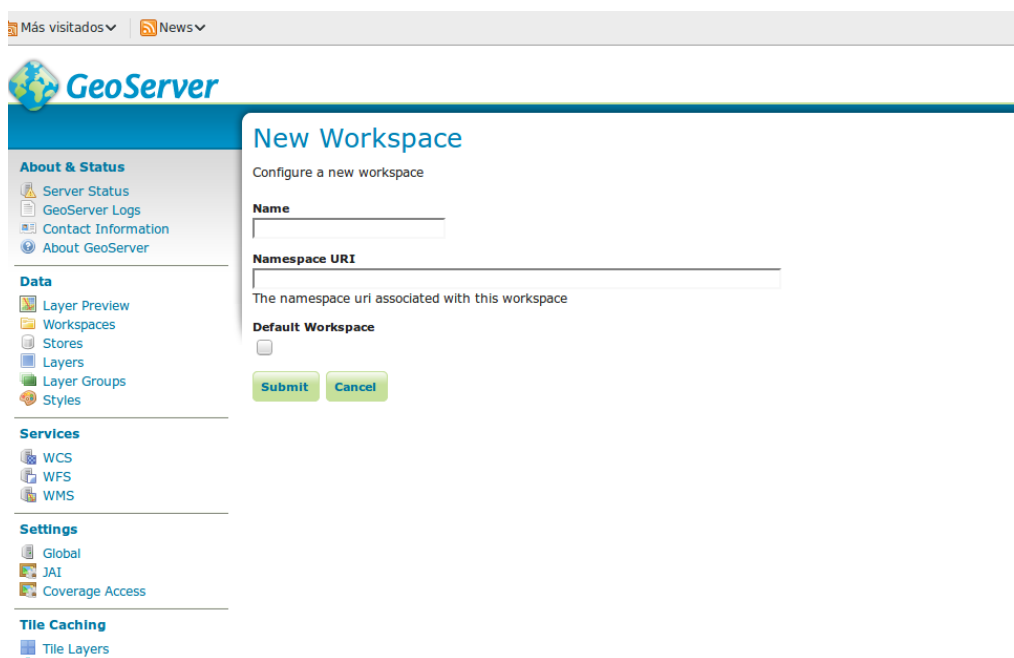


Figura 33. Configuración de workspace

Después, configuraremos el driver para acceder a los datos que deseamos publicar pulsando en 'Data->Stores'. En esta pantalla, *Figura 34*, están las distintas fuentes de datos soportadas. Seleccionaremos 'GeoTIFF' y accederemos a la configuración del driver en concreto, *Figura 35*. Rellenaremos los datos que se nos pidan, en este caso el *workspace* al que será asignado, nombre, descripción y la ruta donde se encuentran el fichero GeoTIFF.

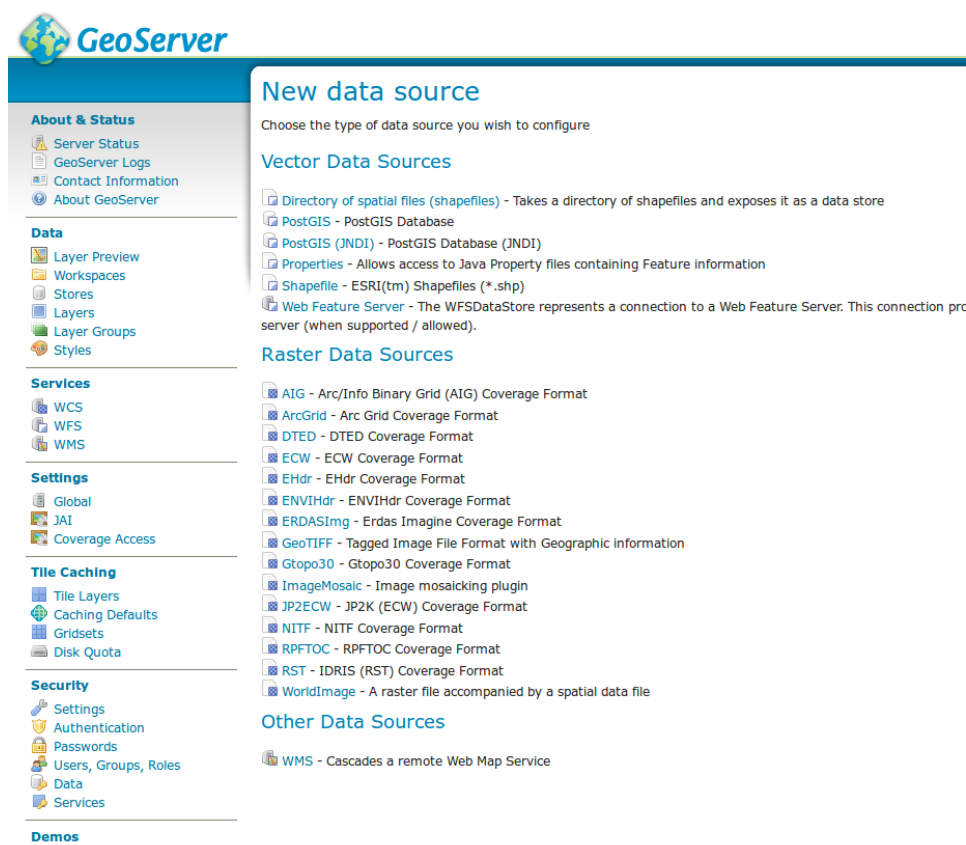


Figura 34. Pantalla de drivers de acceso a datos fuentes

Figura 35. Configuración de driver de acceso a datos fuente GeoTIFF.

A continuación, configuraremos la *Layer* que servirá estos datos en el servicio pulsando en la opción '*Data->Layers*' y después '*Add new Layer*'. Accederemos a la página, Figura 36, donde podremos elegir entre los distintos '*Stores*' que hayamos configurado previamente. Elegimos un '*Store*' y se nos mostrarán todos los elementos publicables que ofrece. Seleccionaremos uno y pulsaremos sobre '*Publish*', que nos llevará a la página de configuración de la '*Layer*', Figura 37 y Figura 38.

En esta pantalla hay varias opciones a completar:

- **Name:** Nombre interno de la capa en el servicio.
- **Enabled:** Disponibilidad de la capa en el servicio.
- **Advertised:** Añade la capa en el fichero de *Capabilities*. Si no seleccionamos esta opción, la capa podrá ser accedida si se conoce su nombre y está activada.
- **Title:** Descripción de la capa.
- **Cordinate Reference Systems:** Sistema de coordenadas de los datos a publicar.
- **Bounding Boxes:** Extensión geográfica que cubren los datos servidos.

Al ser datos de tipo ráster no es necesario configurar nada más para que la capa sea funcional. Si los datos servidos fuesen de tipo vectorial sería necesario asignarles un estilo en la pestaña '*Publishing*'. Este estilo se ha debido crear previamente, sección '*Styles*' y definirá como se representarían. Por ejemplo, si fuesen los ríos de un país se les podría asignar una representación lineal de color azul con un grosor dependiente de su tamaño.

Ya solo falta pulsar el botón ‘Save’ para terminar la configuración de la capa. Este proceso se hará para cada uno de los mapas que se desee publicar.

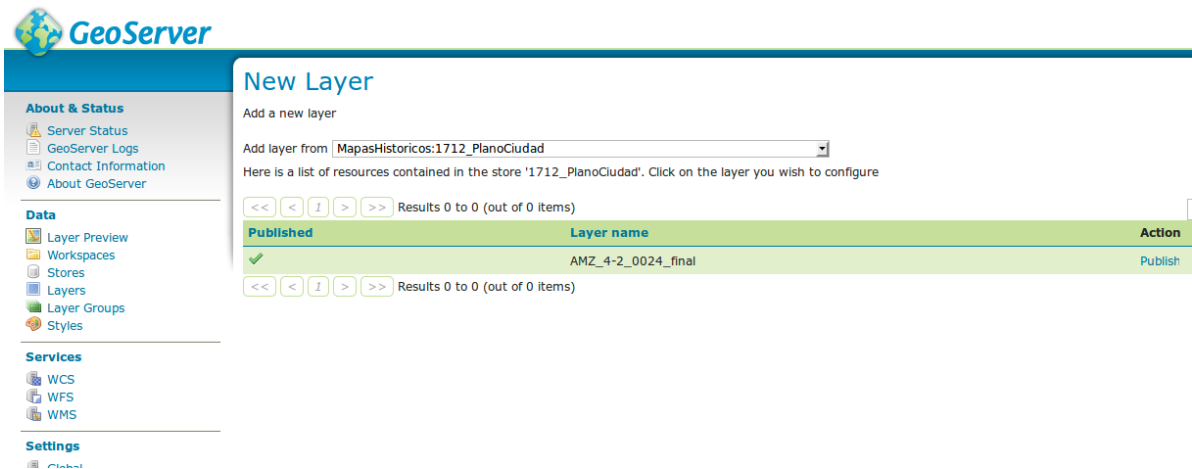


Figura 36. Selección de datos a publicar

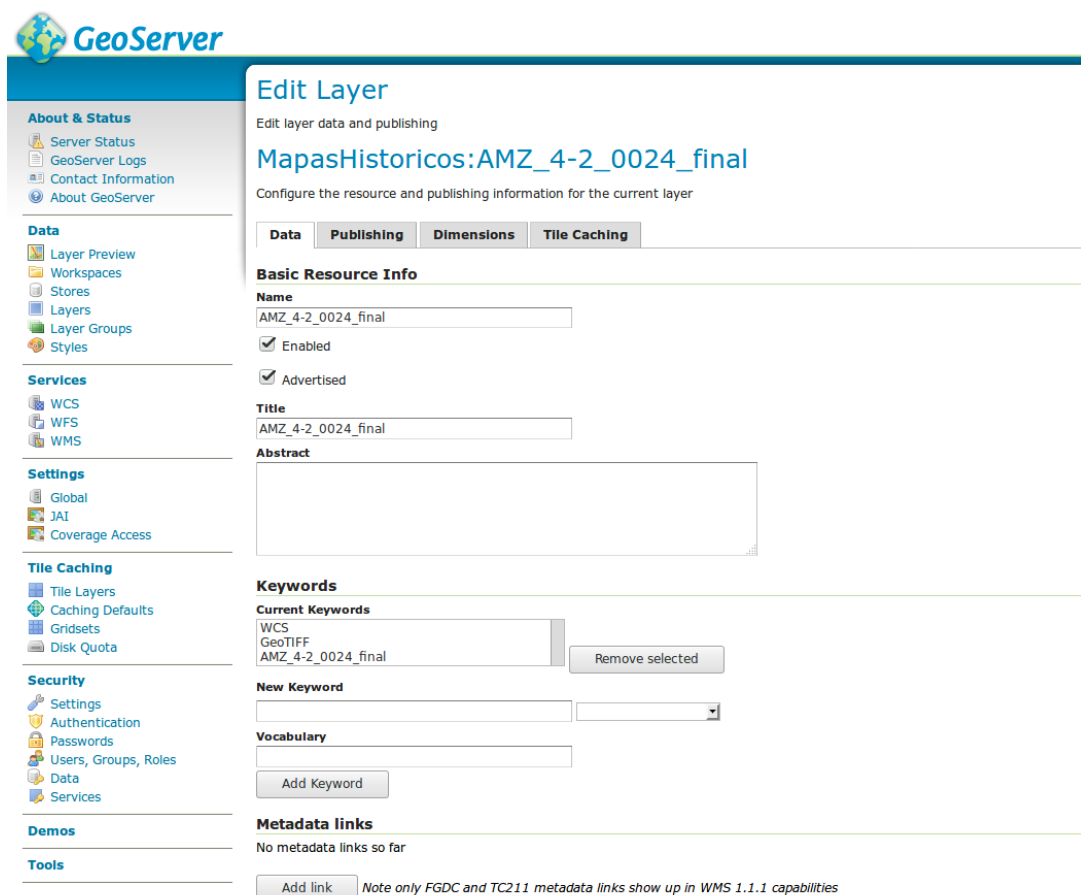


Figura 37. Detalle de pantalla de configuración de ‘Layer’

Demos

Tools

No metadata links so far

Add link

Note only FGDC and TC211 metadata links show up in WMS 1.1.1 capabilities

Coordinate Reference Systems

Native SRS

EPSG:25830

EPSG:ETRS89 / UTM zone 30N...

Declared SRS

EPSG:25830

Find...

EPSG:ETRS89 / UTM zone 30N...

SRS handling

Reproject native to declared

Bounding Boxes

Native Bounding Box

Min X

Min Y

Max X

Max Y

671.967.59179

4.609.526.9264

681.882.99636

4.617.520.2105

Compute from data

Lat/Lon Bounding Box

Min X

Min Y

Max X

Max Y

-0.9358613464

41.6167428490

-0.8144952386

41.6908911798

Compute from native bounds

Coverage Parameters

InputTransparentColor

SUGGESTED_TILE_SIZE

512,512

Coverage Band Details

Band	Data type	Null Values	Unit
RED_BAND	Unsigned 8 bits	-	W.m-2.5r-1
GREEN_BAND	Unsigned 8 bits	-	W.m-2.5r-1
BLUE_BAND	Unsigned 8 bits	-	W.m-2.5r-1
ALPHA_BAND	Unsigned 8 bits	-	

Reload band definitions

Save

Cancel

Figura 38. Detalle de pantalla de configuración de 'Layer'

Por último, podremos ver una lista de todas las capas publicadas en el servicio y realizar una previsualización de las mismas para comprobar su correcto funcionamiento accediendo a la sección 'Data->Layer Preview', Figura 39. Se ofrecen varios formatos de salida, pero lo más eficaz y sencillo es pulsar sobre 'Openlayers'. Acto seguido se nos dará acceso a una pequeña demo de visualización interactiva, Figura 40.

GeoServer

Username

Remember me

Login

About & Status

About GeoServer

Data

Layer Preview

Demos

Layer Preview

List of all layers configured in GeoServer and provides previews in various formats for each.

<<

<

1

>

>>

Results 1 to 19 (out of 19 items)

Search

Type	Name	Title	Common Formats	All Formats
	MapasHistoricos:AMZ_4-2_0024_final	1712_PlanoCiudad	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0696_final	1796_VistaSeptentrion	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0005_final	1809_PlanoTopografico	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0275_0001_01_final	1852_PlanoZaragoza	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0274_final	1861_PlanoZaragoza	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0697_final	1863_PlanoZaragoza	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0459_1_final	1872_PlanoZaragoza	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0072_002_final	1880_PlanoZaragoza	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0072_001_final	1899_PlanoZaragoza_Casañal	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0451_002_final	1910_PlanoParcelarioZaragoza_InstitutoGeograficoYCatastral_p2	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0451_004_final	1910_PlanoParcelarioZaragoza_InstitutoGeograficoYCatastral_p4	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0565_001_final	1938_planoZaragoza_Borobio_p1	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0565_002_final	1938_planoZaragoza_Borobio_p2	OpenLayers KML	Select one
	MapasHistoricos:Planimetria-42_final	1950_PlanoGrafico	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0421_001_final	1958_PlanoGula	OpenLayers KML	Select one
	MapasHistoricos:459_tiled_deflate	459_tiled_deflate	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0005_final_ECW	AMZ_4-2_0005_final_ECW	OpenLayers KML	Select one
	MapasHistoricos:AMZ_4-2_0024_final_deflate_tiled	AMZ_4-2_0024_final_deflate_tiled	OpenLayers KML	Select one

Figura 39. Pantalla de previsualización de layers publicadas



Figura 40. Previsualización de capa en el cliente interactivo

Anexo IV.IV Publicación de los mapas georreferenciados en Virtual Hub

Los *Virtual Hub* de ENERGIC OD son puntos de acceso únicos a datos abiertos geoespaciales heterogéneos, permitiendo a los usuarios descubrir y acceder a datos proporcionados por fuentes dispares, de una manera sencilla. Se ocupan de todas las acciones necesarias, actuando como intermediarios entre datos/metadatos heterogéneos e interfaces de servicio, y armonizando conjuntos de datos geográficos con distintas características, como, series temporales, sistema de referencia de coordenadas, resolución y formato de datos. Las funcionalidades del *Virtual Hub* se proporcionan a través de *APIs restful* y una biblioteca de *JavaScript* fácil de usar para desarrolladores de aplicaciones web y móviles. Todo esto pretende facilitar en gran medida el desarrollo de aplicaciones nuevas y multidisciplinarias alrededor de datos geoespaciales abiertos, estimulando la innovación y las actividades empresariales basadas en la explotación completa de información geográfica y Big Data.

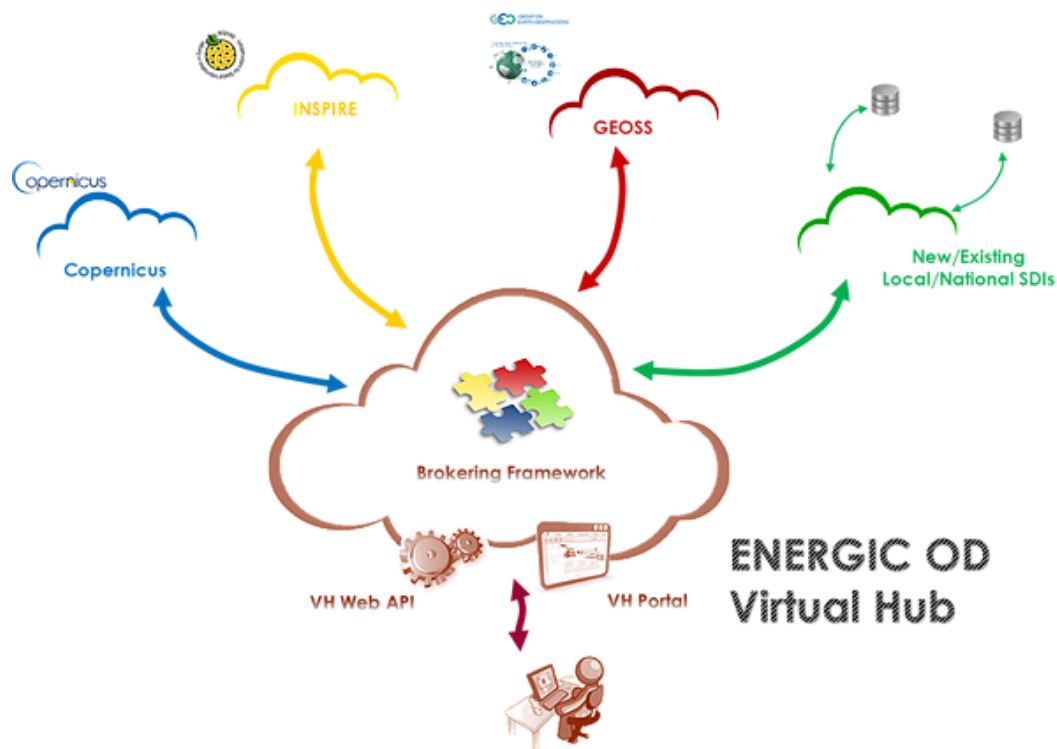


Figura 41. Esquema gráfico de un Virtual Hub

Unos de nuestros socios europeos del proyecto ENERGIC OD nos proporcionó una instancia de un *Virtual Hub* que constaba de tres máquinas virtuales preparadas para la plataforma *VMware vSphere 4.0.0*⁵⁸, ya que el proyecto aún estaba en fase de desarrollo y no se había definido un protocolo o herramienta de instalación. Todos los detalles sobre el entorno y la puesta en marcha del *Virtual Hub* podrán encontrarse en el **Anexo IV.II**.

En primer lugar accederemos a la página de inicio del *Virtual Hub* a través de la url <http://dominiodelamaquina/gi-axe>, Figura 42. En ella pulsaremos sobre 'Configure GI-Axe sources and profilers' que nos dará acceso a la pantalla de login, Figura 43.

⁵⁸ <http://www.vmware.com/es/products/vsphere.html>

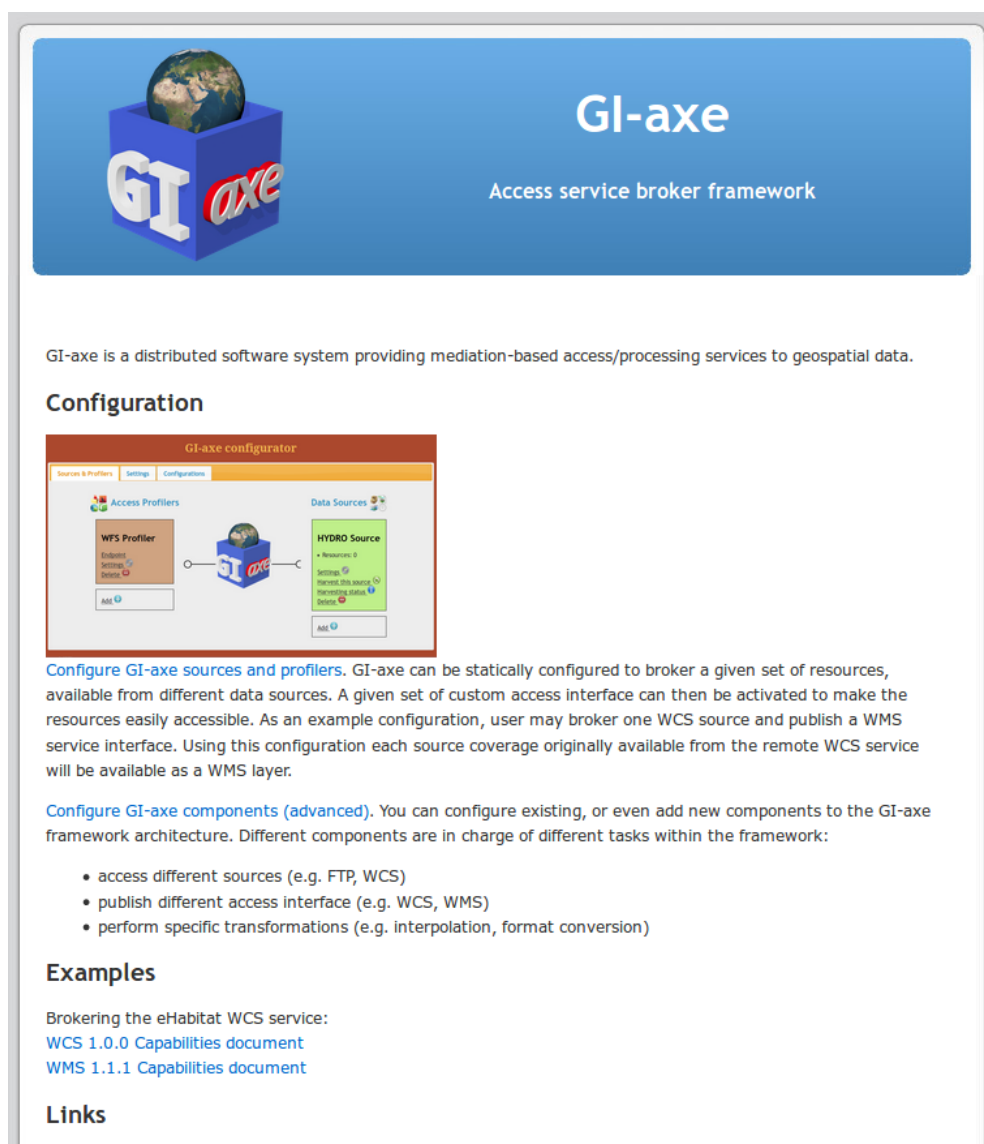


Figura 42. Pantalla inicial del componente GI-axe del Virtual Hub

Tras autenticarnos correctamente, el componente nos dará acceso a la pantalla de configuración de fuentes de datos e interfaces que nos permitirán acceder a éstas, *Figura 44*

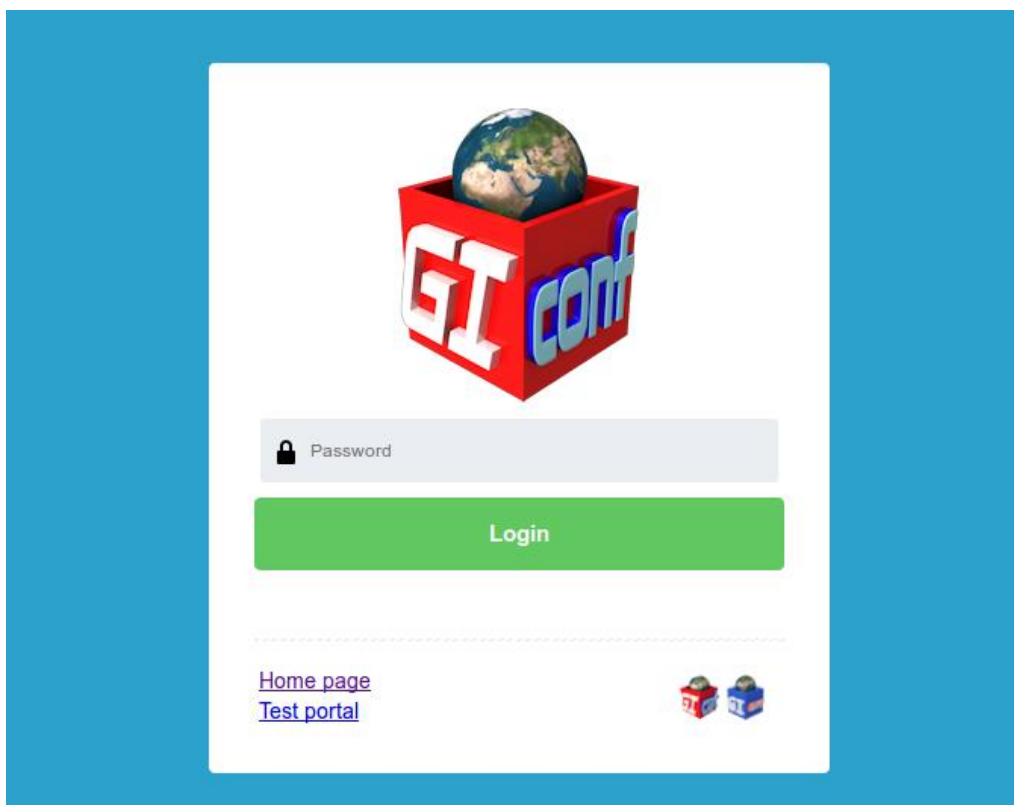


Figura 43. Detalle de la pantalla de login

Desde esta pantalla, *Figura 44*, realizaremos la configuración básica el componente de publicación. En primer lugar, configuraremos el acceso a los ficheros que almacenan los mapas históricos pulsando sobre el botón 'Add' correspondiente a 'Sources'. Tras esto, completaremos todos los campos que aparecen en el dialogo de la *Figura 45*.

Los datos básicos a proporcionar son:

- **Label:** Nombre de la fuente de datos.
- **Protocol:** Seleccionaremos 'FILE', ya que nuestros mapas se encuentran almacenados en ficheros en el árbol de directorios de la máquina local donde se encuentra instalado en servicio.
- **Endpoint:** Ruta local al directorio que contiene los ficheros. Es importante respetar el formato que se observa en la imagen, para evitar errores de acceso.
- **Resource mappers:** Elegiremos la opción 'GEOTIFF', para que únicamente tenga en cuenta este formato de ficheros y agilizar la carga.
- **Connection:** Aquí tenemos dos opciones, seleccionaremos 'Harvested'.
 - *Harvested:* A intervalos regulares, que puede ser establecido, escanea el directorio indicado en el parámetro *EndPoint* para comprobar si hay algún dato nuevo, incorporándolo al servicio de manera automática y transparente.
 - *Distributed:* Se escanea siempre bajo demanda y al vuelo. Es indicada cuando hay un gran movimiento en los datos fuentes. Es más lento, pero asegura el acceso a todos los datos disponibles en ese momento.

Por último, pulsaremos el botón 'Submit' para acabar la configuración de la fuente de datos.

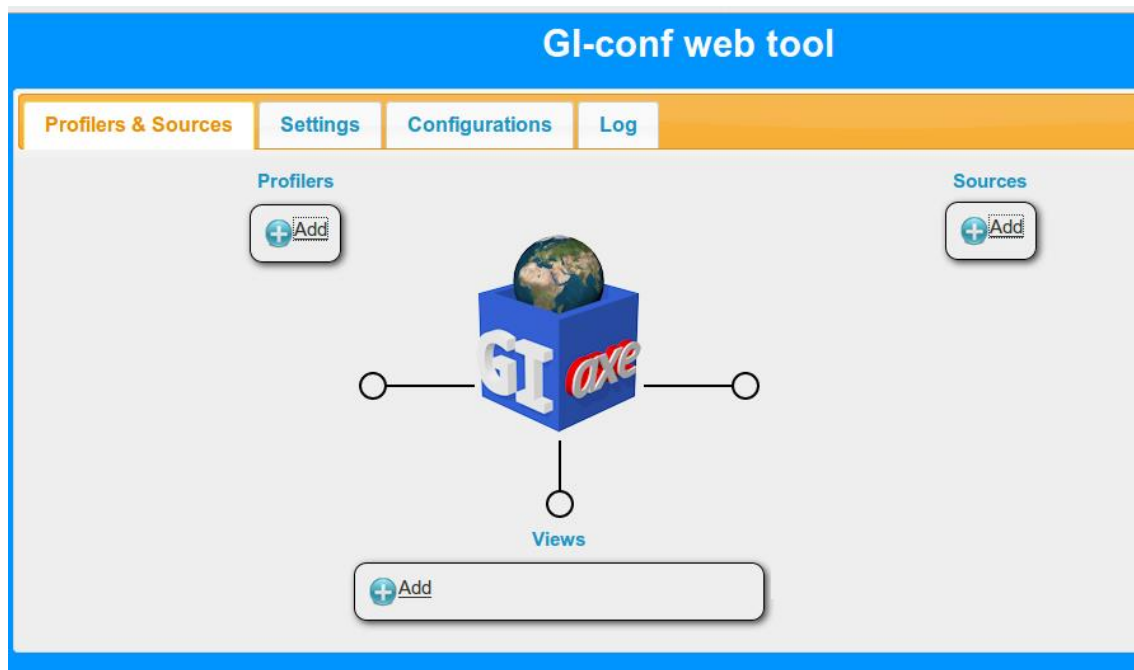


Figura 44. Configuración de sources e interfaces

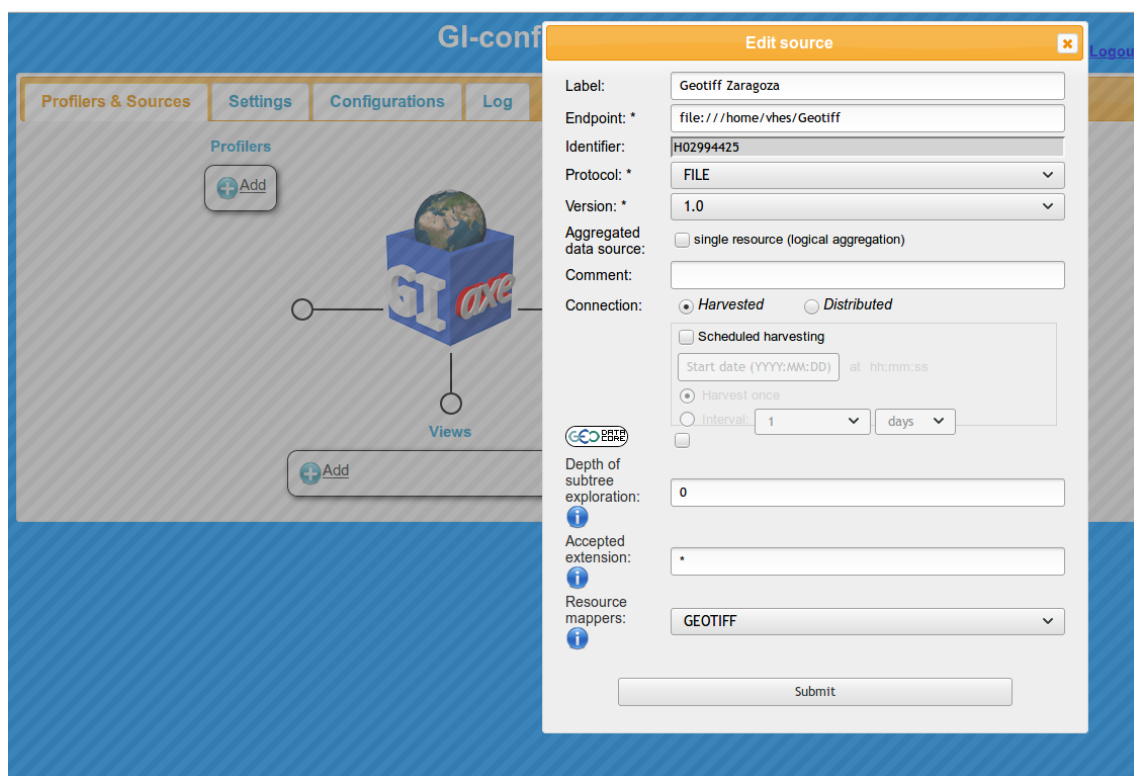


Figura 45. Configuración de un source

En la *Figura 46*, se puede ver la nueva fuente de datos añadida e información sobre la misma. Para terminar, pulsaremos sobre 'Harvest this source' que realizará un barrido del directorio y cargará todos los ficheros GEOTIFF.

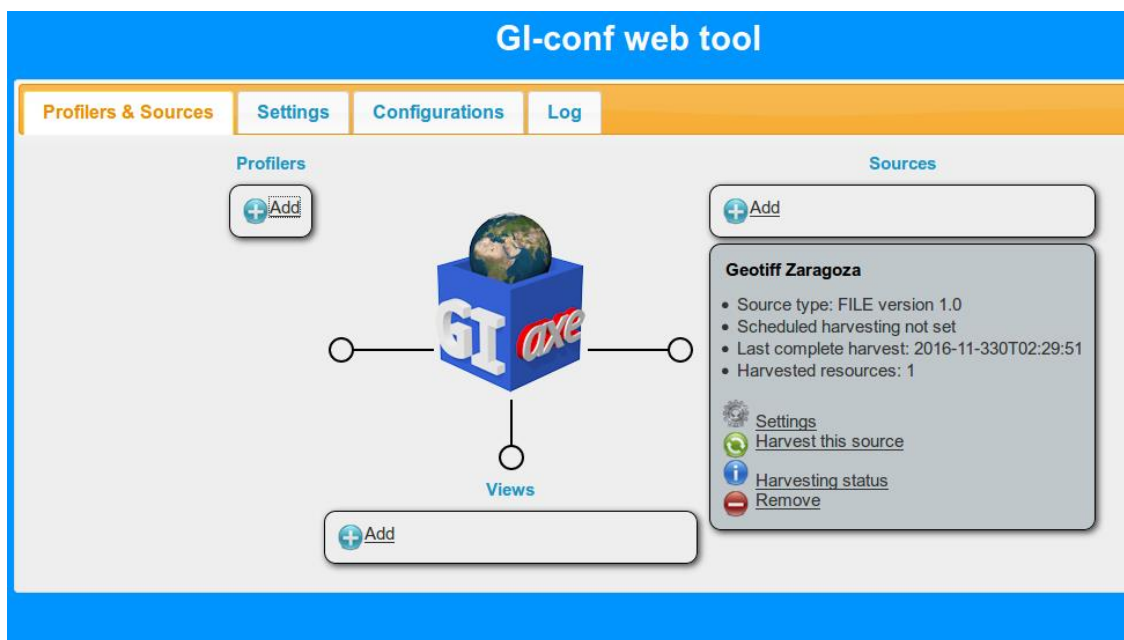


Figura 46. Servicio después de configurar un source

El siguiente paso a realizar es añadir las interfaces que permitirán acceder a las fuentes configuradas, pulsando el botón 'Add' de 'Profilers'. En nuestro caso vamos a configurar los interfaces de WMS y WMTS, que permiten la publicación de mapas en Internet. El dialogo que permite configurar la interfaz deseada presentará diferentes opciones según el tipo de seleccionado. En el dialogo tras seleccionar 'WMS', *Figura 47*, únicamente podremos elegir la versión del interfaz ofrecida por el servicio. En el dialogo tras seleccionar 'WMTS', *Figura 48*, podremos seleccionar el nivel máximo de zoom soportado (niveles de escala de la TileCache) y la versión del interfaz ofrecida por el servicio.

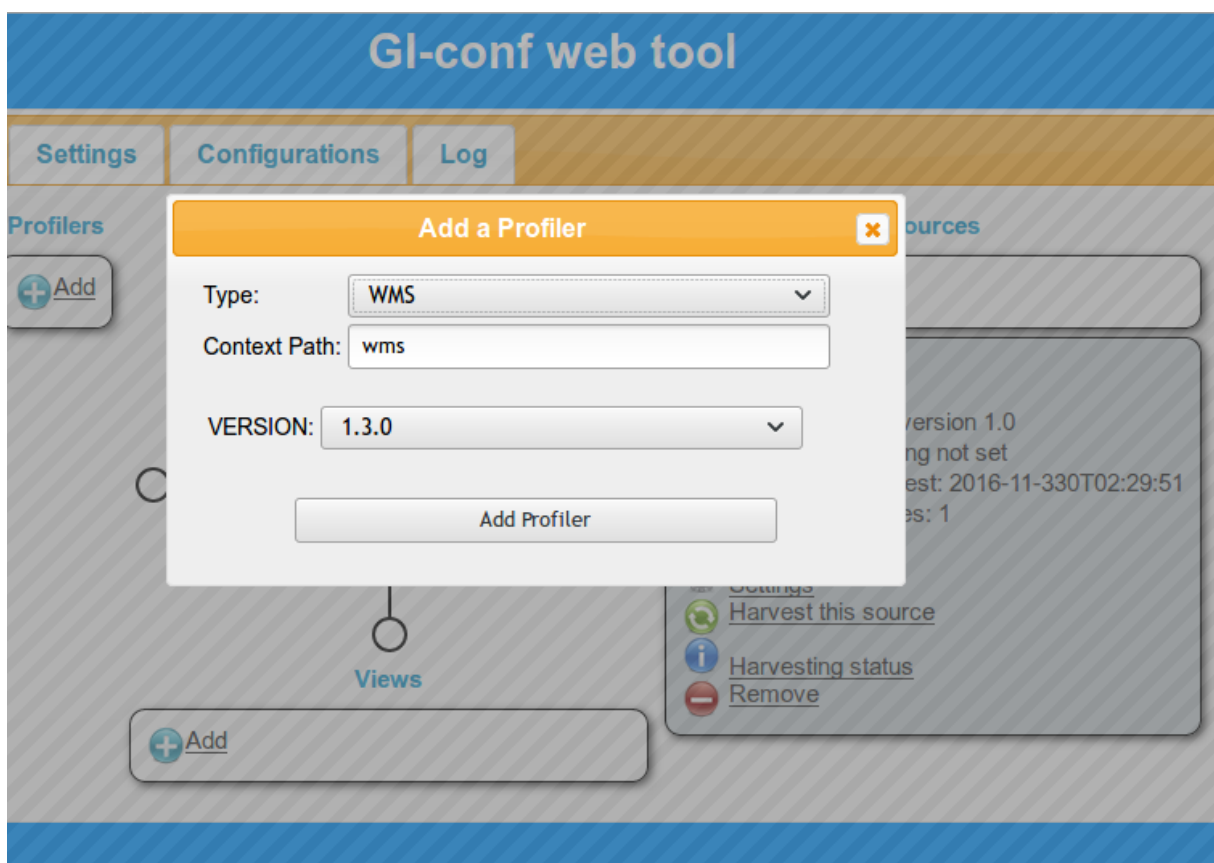


Figura 47. Configuración WMS

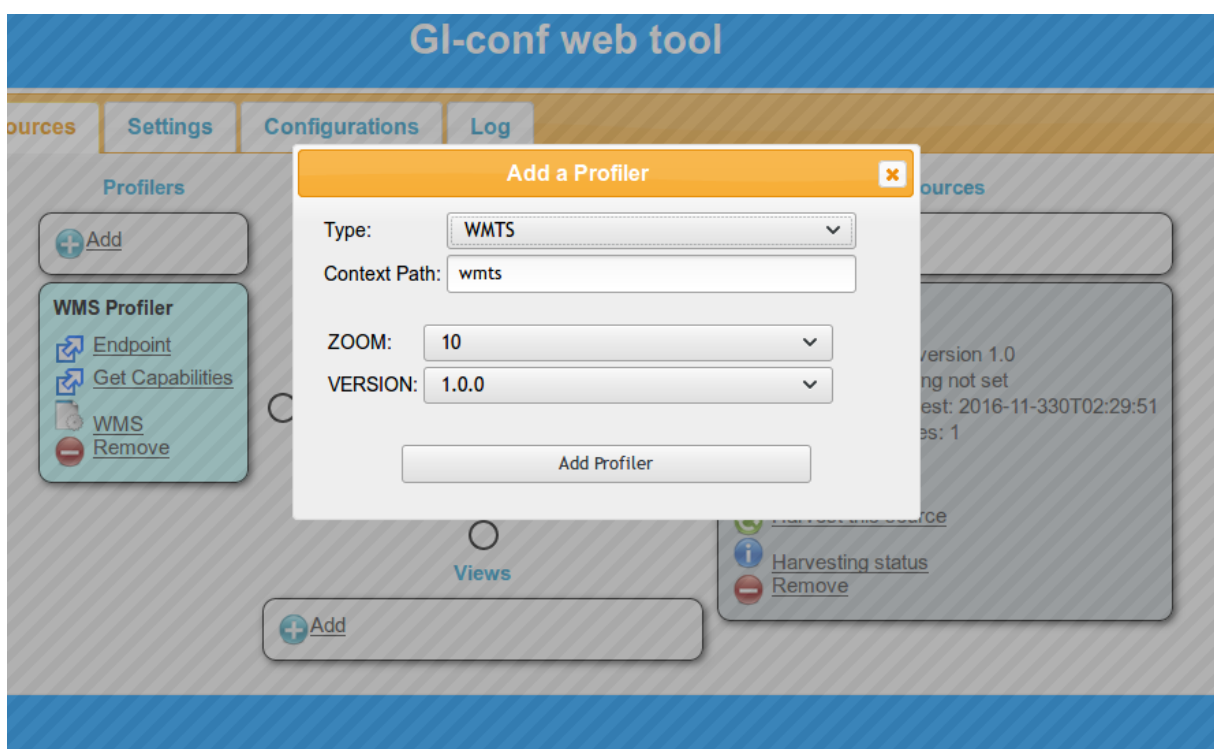


Figura 48. Configuración WMTS

En la *Figura 49* vemos el resultado final de la configuración del servicio. Para cada uno de los interfaces añadidos tenemos cuatro opciones:

- **EndPoint:** Localización de los recursos servidos.
- **GetCapabilities:** Petición que informa sobre las características de cada una de las capas ofrecidas por el servicio. En nuestro caso será una por cada uno de los mapas históricos depositados en el directorio configurado en la fuente.
- **Test:** Se lanza una petición de prueba para comprobar su correcto funcionamiento.
- **Remove:** Elimina la interfaz.

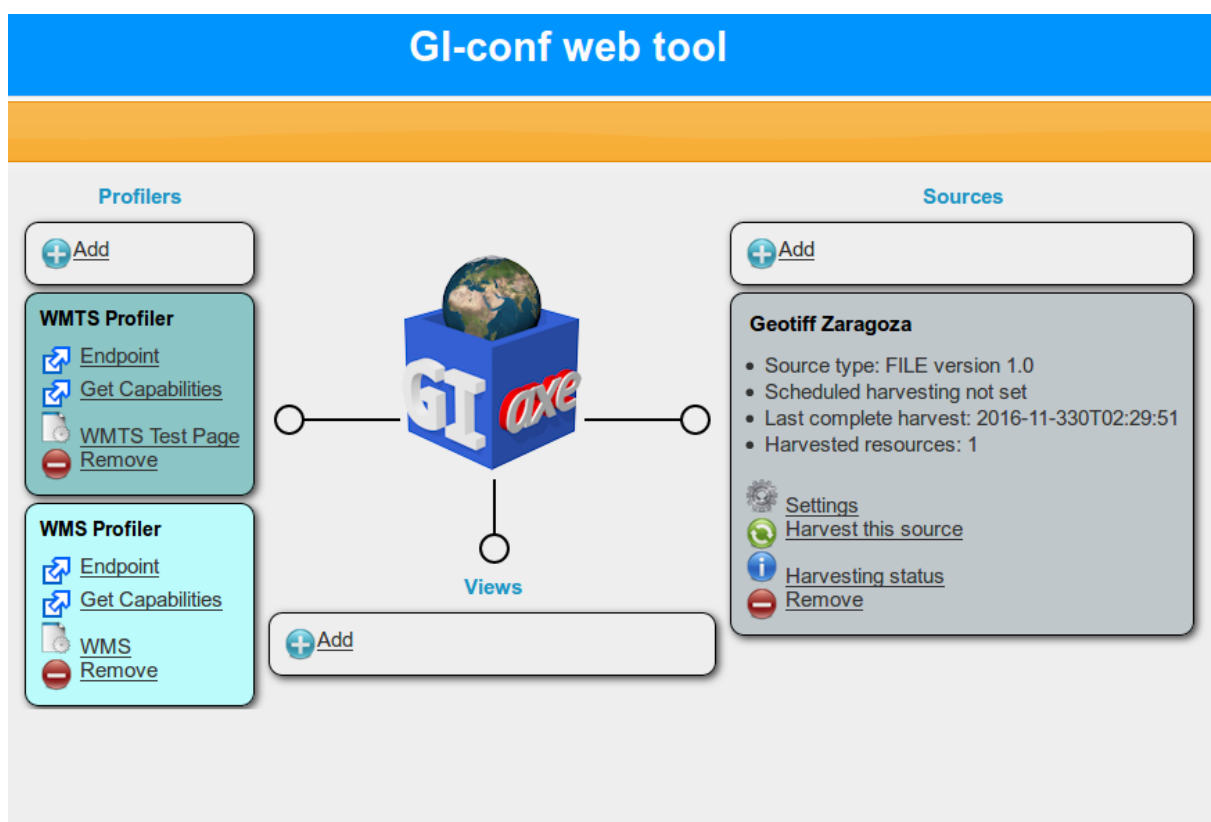


Figura 49. Servicio tras su configuración

Con esto ya tendríamos una configuración básica del componente y estaría listo para su uso.

Anexo V Aplicación de visualización de mapas históricos

Anexo IV.V Desarrollo de la aplicación web

Esta aplicación estará formada por dos componentes que necesitarán interactuar entre ellos. Esta comunicación se hará mediante el *facade* que ambos ofrecen. El patrón de diseño *Facade* permite estructurar un entorno de programación y reducir su complejidad con la división en subsistemas, minimizando las comunicaciones y dependencias entre estos. Se utiliza cuando se necesite proporcionar una interfaz simple para un subsistema complejo, o cuando se quiera estructurar varios subsistemas en capas, ya que las fachadas serían el punto de entrada a cada nivel. Otro escenario proclive para su aplicación surge de la necesidad de desacoplar un sistema de sus clientes y de otros subsistemas, haciéndolo más independiente, portable y reutilizable (esto es, reduciendo dependencias entre los subsistemas y los clientes).

El componente visor, configurado y desarrollado se puede utilizar independientemente del componente contenedor, pero sería un visor común ya que no se explotarían sus capacidades para mostrar mapas históricos ofrecidas mediante su *facade*.

Las características de *JavaScript*, un lenguaje funcional que a pesar de que casi todos sus elementos objetos, han influido en la manera de organizar el desarrollo. Debido a que el lenguaje no da tantas facilidades para estructurar su desarrollo, como por ejemplo Java y su concepto de clase y herencia, se han organizado los componentes físicamente de manera que las diferentes funcionalidades que los conforman estén separadas y el desarrollo sea sencillo, *Figura 50*.

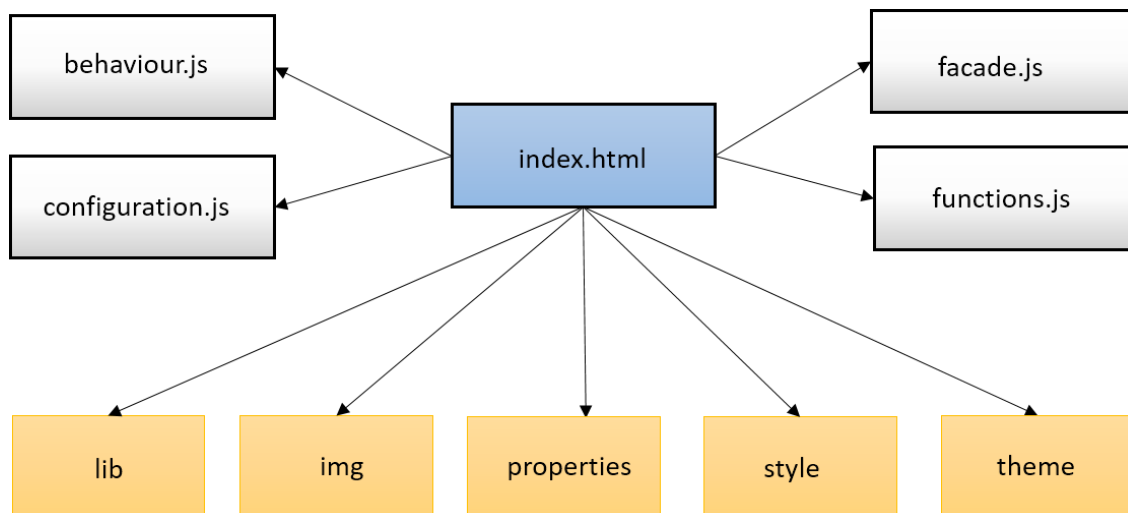


Figura 50. Estructura de los componentes de la aplicación de visualización

- index.html: En este fichero mediante el lenguaje HTML se crea el esqueleto de la interfaz gráfica del componente. En él se enumeran las distintas partes que lo conformaran y se hará referencia a los ficheros .js que contienen el código con las funcionalidades concretas

desarrolladas. Cuando se cargue en el navegador todas estas partes serán configuradas dinámicamente al ejecutarse las funciones de JavaScript previamente referenciadas.

- **behaviour.js:** En este fichero están las funcionalidades concretas desarrolladas para este componente. Desde su método `init()` se inicializan todos los elementos definidos en el fichero `index.html` como se ha mencionado anteriormente. Además, se instancian y configuran todos los elementos de *OpenLayers* o *jQuery* utilizados para conformar el visor o el contenedor. También incluye toda la lógica desarrollada que define el comportamiento del componente y sus reacciones a los distintos eventos que pueden suceder.
- **configuration.js:** En este fichero se guarda la configuración específica que definirá el componente y almacena los parámetros que definen cada elemento que lo conforma. Por ejemplo, el tamaño y posición de los distintos controles, las capas base que se utilizarán para mostrar la cartografía actual en el visor y la extensión inicial que se mostrará, los mapas históricos mostrados en el carrusel del contenedor, etc.
- **facade.js:** En este fichero se define la interfaz que ofrece el componente para interactuar con él. Mediante las funciones ofrecidas se realizará toda la interacción necesaria entre el visor y contenedor. Por ejemplo, están definidas las funciones que permiten mostrar u ocultar un mapa histórico determinado, cambiar la opacidad del mapa histórico mostrado, etc.
- **functions.js:** En este fichero se encuentran funciones auxiliares que pueden ser útiles en distintos puntos del componente y se sobrescriben funciones para cambiar su comportamiento por defecto.
- **lib:** directorio que contiene las librerías necesarias, como *OpenLayers* 2.11 y *jQuery* 1.8
- **img:** directorio que contiene los recursos en forma de imagen utilizados.
- **properties:** directorio que contiene ficheros de *properties* utilizados en el componente, como los ficheros de internacionalización que permiten mostrarlo en distintos idiomas.
- **style, theme:** directorios que contienen ficheros `.css`. CSS⁵⁹ es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de etiquetas. Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML o XHTML. Estos ficheros contienen la estética usada para los distintos elementos del componente. Por ejemplo, se define una única vez el tamaño y color para un botón, que podrá ser usado para representar múltiples botones similares.

Los dos componentes son distribuidos mediante ficheros `.war` creados con la herramienta *Maven*⁶⁰ que pueden ser desplegados en cualquier contenedor de aplicaciones, como *Apache Tomcat*. *Maven* es una herramienta de software para la gestión y construcción de proyectos Java en 2002. Es similar en funcionalidad a *Apache Ant*, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML. Es un proyecto de nivel superior de la *Apache Software Foundation*.

⁵⁹ <https://www.w3.org/Style/CSS/>

⁶⁰ <https://maven.apache.org/>

Maven utiliza un *Project Object Model* (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado. Otra de las características de *Maven* que se ha usado son sus '*profiles*'. *Maven* nos permite crear diferentes perfiles para la construcción de nuestros proyectos. El perfil nos va a permitir definir propiedades concretas para la construcción del proyecto en distintas situaciones. Por ejemplo, si estamos construyendo el proyecto que deberá ejecutarse en distintos entornos.

Como se comenta en la sección 2.3.3 y 3.3, la configuración de la interfaz WMTS de *Virtual Hub* conlleva problemas y su rigidez de diseño obligó a cambiar la capa base ofrecida por IDEZar por una compatible con los niveles de escala fijos que iba a ofrecer *Virtual Hub*, ya que no iban a ser configurables como sería deseable. Este problema se pudo solucionar fácilmente debido a la estructuración del componente y los perfiles de *Maven*. Se definieron dos perfiles distintos en los que se especifican los parámetros de conexión necesarios para conectarse a uno u otro servicio WMTS. Esto consiste básicamente en definir una serie de variables y en cada definición de perfil asignarles el valor adecuado. En el fichero *configuration.js* en lugar de insertar, por ejemplo, la url del servicio se pone el nombre de esta variable. Posteriormente *Maven* buscará en todos los ficheros este nombre de variable y la sustituirá por el valor que se la haya asignado en el perfil.

Además, se realiza un proceso de ofuscación mediante una serie de *scripts* propios que se apoyan en librerías externas. Este proceso se hace antes del despliegue de una aplicación web para eliminar todo lo superfluo de tu código con el fin de hacerlo lo más compacto posible y que pese muy poco. Este proceso elimina todos los espacios innecesarios, cambios de línea, tabuladores, comentarios y otros caracteres similares reduciendo mucho su tamaño y permitirá descargarlo más rápido desde el servidor. Además de lo anterior, hace cambios en el código para complicar depurarlo y reutilizarlo. El objetivo es que no lo copien o por lo menos dificultarlo. Para ello se llevan a cabo múltiples operaciones, como cambio de nombre a variables y funciones, recodificación de cadenas, sustitución de parte del código por otro equivalente, pero más complejo.

Anexo IV.VI Manual de usuario de la aplicación web

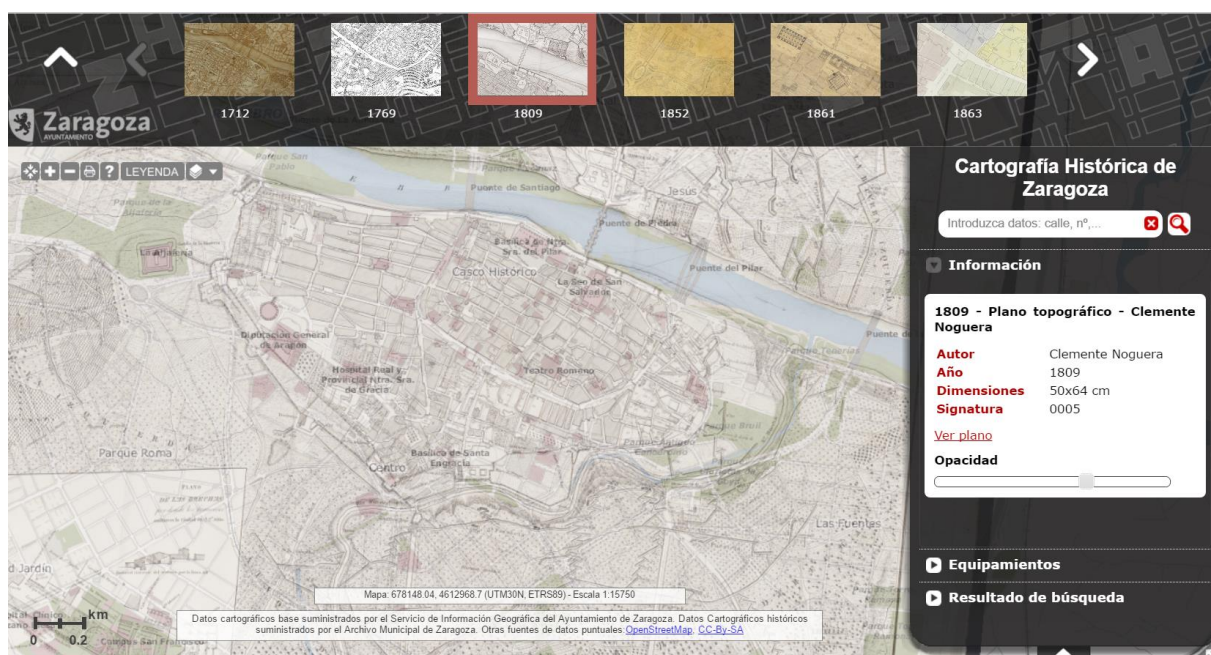


Figura 51. Vista general de la aplicación

Herramientas

La aplicación ofrece las herramientas básicas para la navegación en el mapa. Se pueden realizar movimientos básicos mediante el ratón, pero también hay una barra de herramientas con los siguientes elementos.



Zoom in sobre el centro del mapa.



Zoom out sobre el centro del mapa.



Volver al zoom inicial del mapa



Mostrar dialogo de ayuda.

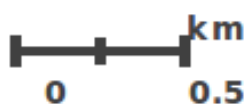


Mostrar leyenda del mapa.



Cambiar la capa base.

También se incluyen algunas herramientas que informan acerca del estado actual del mapa.



Escala actual del mapa.

Mapa: 675867.32, 4612700.28 (UTM30N, ETRS89) - Escala 1:39157

Coordenadas del cursor.

Datos cartográficos base suministrados por el Servicio de Información Geográfica del Ayuntamiento de Zaragoza. Datos Cartográficos históricos suministrados por el Archivo Municipal de Zaragoza. Otras fuentes de datos puntuales: [OpenStreetMap](#), [CC-BY-SA](#)

Información sobre los proveedores de la cartografía.

Visualización de los mapas históricos

El objetivo principal de la aplicación es facilitar el acceso a un conjunto de mapas históricos de la ciudad de Zaragoza. Cada mapa se representa con una imagen de vista previa en el carrusel incluido en la parte superior del visor. Al hacer clic en una de las vistas previas, se resaltará y el mapa histórico correspondiente aparecerá en el visor.



Figura 52. Carrusel de selección de mapa histórico

Es posible navegar entre todos los mapas históricos preseleccionados mediante las flechas situadas a ambos lados del carrusel.

Al añadir un mapa histórico al visor aparecerá sobre la capa base cartográfica actual de la ciudad de Zaragoza, teniendo el mapa actual como fondo. Para facilitar la comparación entre ambos se podrá asignar al mapa histórico el grado de transparencia deseado. Será posible navegar en ambos mapas simultáneamente y cambiar tanto el mapa histórico mostrado en primer plano como el mapa actual en segundo plano.

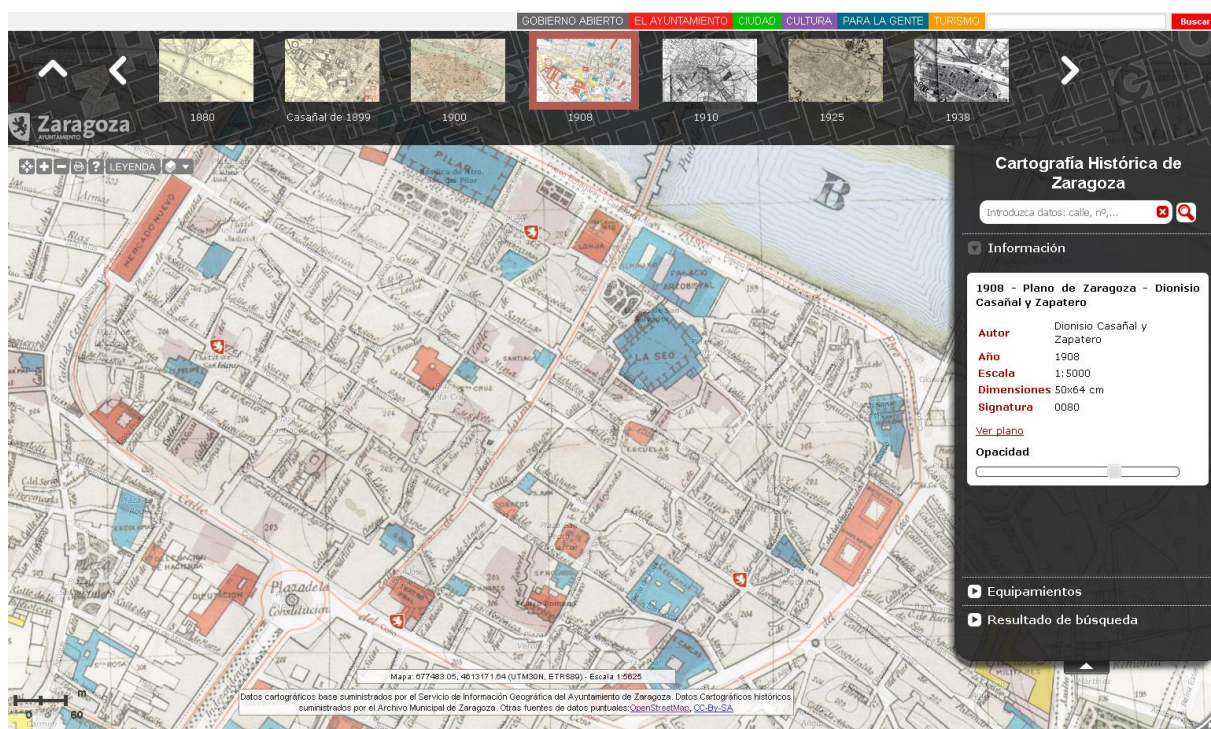


Figura 53. Visualización del mapa histórico

Herramientas avanzadas

Además de las herramientas de navegación y las opciones para seleccionar los datos a visualizar en el mapa, existen características adicionales para aumentar la funcionalidad de la aplicación. Éstas se encuentran situadas en el panel lateral.

El elemento 'Información', proporciona datos adicionales sobre el mapa histórico visualizado. Desde este elemento es posible cambiar la opacidad aplicada al mapa histórico.



Figura 54. Detalle del elemento 'Información'

El elemento 'Equipamientos', permite visualizar distintos tipos de equipamientos y puntos de interés para los ciudadanos de carácter público simultáneamente o no con los mapas históricos.



Figura 55. Detalle del elemento 'Equipamiento'

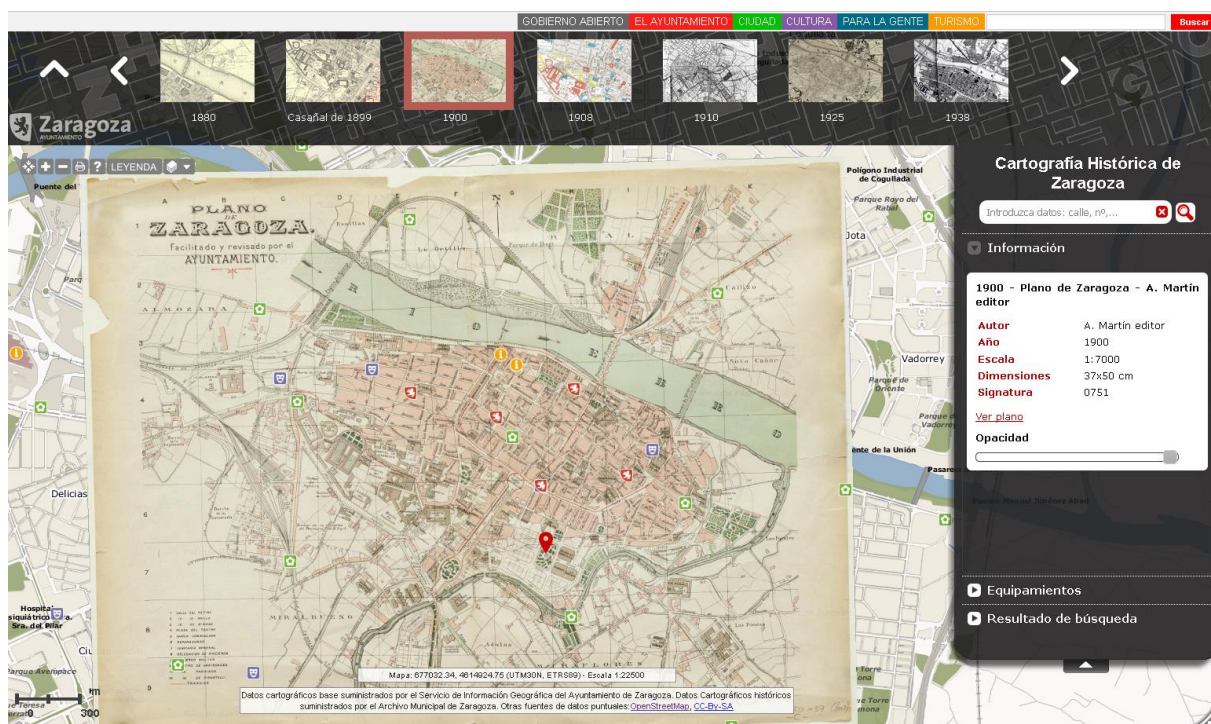


Figura 56. Equipamientos mostrados sobre el mapa histórico

Por último, también existe la posibilidad de realizar una búsqueda toponímica a través introduciendo en el cuadro de búsqueda los términos deseados. Aparecerán los resultados de esta búsqueda y al seleccionar cualquiera de ellos se centrará el mapa sobre él, añadiéndose un marcador para señalar su posición.



Figura 57. Detalle del elemento 'Busqueda'

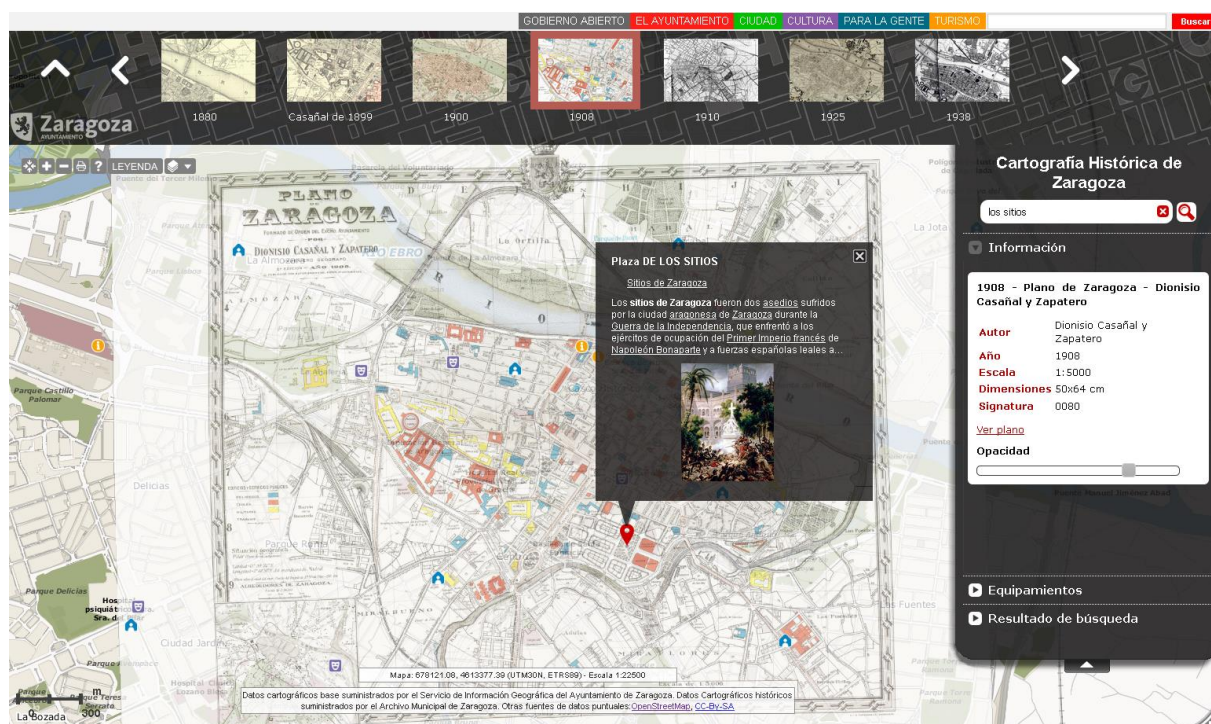


Figura 58. Visualización de un resultado de la búsqueda.

Anexo VI Plan de pruebas

En la *Figura 59* y la *Figura 60* se puede ver un ejemplo de las pruebas definidas en el plan de pruebas. Para cada prueba se indica la descripción/propósito de la prueba, acciones previas, acciones a ejecutar y resultados esperados y el navegador web en que se prueba el sistema y el resultado de dicha prueba.

PRUEBAS									
Precondiciones/acciones previas				Acciones de ejecución		Resultados esperados			
Pruebas generales del Aplicación de mapas históricos									

Figura 59. Ejemplo de las pruebas realizadas en el plan de pruebas

Anexo VII Herramientas y programas utilizados en el proyecto

A continuación, se detallan las herramientas utilizadas para la realización del presente proyecto. Las herramientas han sido clasificadas según sus ámbitos de utilización.

- **Desarrollo:**

- *Virtual Hub* ejecutándose en *VMware vSphere 4.0.0*
- *Mozilla Firefox 4+*, *Google Chrome*, *Safari 6+*, *Internet Explorer 9+*, *Opera 11+*. Es necesario que estos navegadores soporten *HTML5 & CSS3*.
- *JavaScript 1.8.5*
- *jQuery 1.8 + jQueryUI*
- *OpenLayers 2.11*
- *Apache Tomcat 5* con *Java 1.5*

- **Documentación del proyecto:**

- *Adobe Acrobat XI Pro*: lectura de manuales y guías.
- *Microsoft Word 2010*: documentación y memoria del proyecto.
- *Microsoft Excel 2010*: pruebas del sistema.
- *Microsoft PowerPoint 2010*: presentación.
- *Gantt Project 2.7.1*: planificación de tareas.

- **Gestión del proyecto y copias de seguridad:**

- *SVN*: control de versiones.
- *Maven*: herramienta para la gestión y construcción de proyectos.
- *Bugzilla*: gestión de tareas.