

In pl@nta

El Instituto Geográfico Nacional (IGN) dispone de una gran cantidad de documentación geográfica antigua.

Ficheros digitales procedentes del escaneado de planos de edificios singulares de gran valor cartográfico e histórico.

Realizados a diferentes escalas, principalmente 1:250 y 1:500, entre los años 1850 y 1900.

El objeto de este proyecto es recopilar los 427 ficheros en formato *ECW* de los planos manuscritos de edificios singulares, digitalizarlos y procesarlos con software libre y realizar una aplicación web *responsive* gratuita para todo el público.

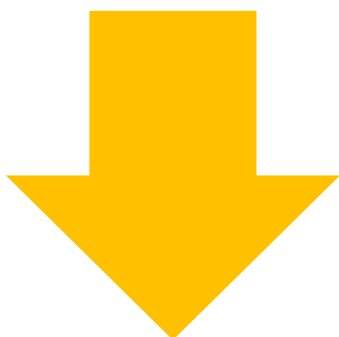


Descarga de Datos

El primer proceso es la descarga masiva del centro de descargas del CNIG (IGN):

Metadatos:

- Planos de edificios
- Descripción: planos manuscritos de edificios singulares.
- *SGR*: ED50. Proyección UTM huso 30 extendido.
- Ud. descarga: edificio.
- Formato: *JPG* (sin georreferenciar con resolución 250 ppp) y *ECW*.
- Ficheros digitales procedentes del escaneado de planos de edificios singulares. Realizados a diferentes escalas, principalmente 1:250 y 1:500, entre los años 1850 y 1900. No existen documentos de todas las provincias, tan sólo existen planos en las provincias de Badajoz, Ciudad Real, Guadalajara, Madrid, Segovia y Toledo.



La descarga automática del Centro de Descargas del IGN y del CNIG se realiza a través de una aplicación *Java* (archivo *.JNLP*) que agiliza y automatiza el proceso de descarga.

El tamaño aproximado es de 9308,65 (MB)

La digitalización de los planos de edificación que se custodian en el Archivo del Instituto Geográfico Nacional y la publicación en el portal de descarga garantiza la preservación de los documentos y el acceso a una fuente de información indispensable para cualquier estudio de evolución del territorio.

La georreferenciación de estas hojas mejora además las consultas geográficas y la realización de estudios comparativos espaciales.

La Georreferenciación de los ficheros a 400 ppp a partir de las coordenadas de las esquinas, se pueden obtener en el mismo Centro de Descargas como parte del Equipamiento Geográfico de Referencia Nacional.

Procesado de Datos

Para el procesado y la digitalización de los archivos *ECW* utilizaremos el software de código libre y gratuito *QGIS*.

1.QGIS

Es un Sistema de Información Geográfica (SIG) de código libre.

El primer proceso es crear la estructura de la base de datos, crearemos tres capas de archivo *shape* con la siguiente estructura:

MultiPolygon		
Nombre	Tipo	Longitud
Id	Integer	10
Nombre	String	80
Provincia	String	80
Region	String	80
Municipio	String	80
Escala	String	80
Plano	String	80
Fecha	Date	10

MultiLineString		
Nombre	Tipo	Longitud
Id	Integer	10
Nombre	String	80
Provincia	String	80
Region	String	80
Municipio	String	80
Escala	String	80
Plano	String	80
Fecha	Date	10
Id Poligono	Integer	10

MultiPoint		
Nombre	Tipo	Longitud
Id	Integer	10
Nombre	String	80
Habitaculo	String	80
Id Poligono	Integer	10

En codificación UTF-8 y en el SCR ED50/UTM Zona 30 N, EPSG: 23030.

Una vez cargados los [ECW](#) en el sistema SCR ED50/UTM Zona 30 N realizaremos un cambio de Datum a ETRS 89/ UTM Zona 30 N, para ello utilizaremos la [Rejilla para cambio de Datum](#) entre ED50 y ETRS89 (en formato NTV2) del Instituto Geográfico Nacional (IGN).

Consta de dos rejillas, una para la península PENR2009.gsb y otra para Baleares denominada BALR2009.gsb. Ambas rejillas están en formato NTV2 y contienen los incrementos en longitud y latitud entre estos dos datums.

```
+proj=utm +zone=30 +ellps=intl +nadgrids=C:\Program Files\QGIS  
3.2\share\proj\PENR2009.gsb +units=m +wktext+no_defs
```

A continuación, para importarlos posteriormente a [Geoserver](#), guardaremos todos los [ECW](#) como [GeoTIFF](#).

2. GeoTIFF

Es un estándar de metadatos de dominio público que permite que información georreferenciada sea encajada en un archivo de imagen de formato TIFF.

Para digitalizar el edificio, crearemos polígonos siguiendo la arquitectura.

Una vez finalizada la digitalización, usaremos el algoritmo [recopilar geometrías](#): Este algoritmo toma una capa vectorial y recopila la información de sus geometrías en nuevas geometrías multiparte.

Para generar la capa denominada Líneas, utilizaremos el algoritmo de [polígonos a líneas](#): Este algoritmo toma una capa de polígono y crea una capa de línea, con líneas que representan los anillos de los polígonos de la capa de entrada.

Inmediatamente, insertaremos puntos siguiendo el esquema de la leyenda en cada una de las zonas de interés del edificio histórico.

Se seguirá el mismo proceso para cada uno de los edificios históricos en formato [GeoTIFF](#).

Después, se calculara el *centroide* de cada edificio para posicionarlo en un mapa a mayor escala, para ello utilizaremos el algoritmo *Media de Coordenadas*: Este algoritmo procesa una capa de puntos con el centro de gravedad de geometrías de una capa de entrada.

Finalmente, una vez digitalizados los edificios, exportaremos cada una de las capas a un formato *GeoJSON*.

3. *GeoJSON*

Es un formato estándar abierto diseñado para representar elementos geográficos sencillos, junto con sus atributos no espaciales, basado en JavaScript Object Notation.

Arquitectura Web del Servidor

En esta fase se implementará todo a través de [Google Cloud Plataform](#), una suite que contiene diversos servicios que funcionan en la misma infraestructura que utiliza Google.

Empleando una cuenta de correo electrónico *gmail* podremos acceder a 300\$ gratuitamente para trabajar con [Google Cloud Platform](#) durante los próximos 12 meses, tiempo suficiente para hacer un producto mínimo viable y proporcionar retroalimentación para el desarrollo futuro.

4. Google App Engine

Crea aplicaciones muy escalables en una plataforma sin servidor totalmente gestionada.

En la que se instalará [Geoserver](#).

5. Geoserver

Es un servidor de código abierto para compartir datos geoespaciales. Diseñado para la interoperabilidad, publica datos de cualquier fuente de datos espacial importante utilizando estándares abiertos.

Para la base de datos utilizaremos [Google Cloud SQL](#) que es un servicio de base de datos totalmente administrado que facilita la configuración, el mantenimiento y la administración de las bases de datos relacionales en [Google Cloud Platform](#), en nuestro caso implementaremos la opción [PostgreSQL](#) (permitir [PostGIS](#)).

6. PostgreSQL

Es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto.

7. PostGIS

Es un extensor de base de datos espacial para la base de datos relacional de objetos PostgreSQL.

Para almacenar los archivos *GeoTIFF* y servirlos a la aplicación web, utilizaremos:

8. Google Cloud Storage

Es un servicio de almacenamiento de archivos en línea RESTful para almacenar y acceder a datos en la infraestructura de Google Cloud Platform.

Arquitectura del Cliente

En este punto, para lograr la visualización interactiva utilizaremos por un lado, el cliente ligero [*Leaflet*](#).

9. Leaflet

Es la biblioteca de JavaScript de código abierto líder para mapas interactivos aptos para dispositivos móviles teniendo en cuenta la simplicidad, el rendimiento y la facilidad de uso.

¿Por qué usar *Leaflet*?

Porque esta *API* ofrece muchas ventajas:

- Sencillo y rápido de aprender.
- Facilidad de uso.
- Características básicas pero que funcionan a la perfección.
- Soporte móvil.
- *HTML 5* y *CSS3*.
- Funciona tanto en los modernos como en los viejos navegadores web.
- Ampliable con *plugins*.
- *API* bien documentada.

Para ello solo tenemos que cargar la hoja de estilos de *leaflet* y sus librerías en el siguiente orden:

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css"
  integrity="sha512-
puBpdR0798OZvTTbP4A8Ix/I+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWAKuyeQUytO7+7
N4QKrDh+drA=="
  crossorigin=""/>
```

```
<script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"
  integrity="sha512-
QVftwZFqvtRNi0ZyCtsznlKSWOStnDORoeFr1enyq5mVL4tmKB3S/EnC3rRJcxCPavG10lcrVGSmPh
6Qw5lwrg=="
  crossorigin=""></script>
```

Por otro lado, para la estructura del contenido de la *app responsive* utilizaremos el *front-end Bootstrap 4*.

10. Bootstrap

Es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web.

¿Por qué usar *Bootstrap*?

Porque el *CSS* de *Bootstrap 4* se ajusta a teléfonos, tabletas y computadoras de escritorio de forma responsive.

Y además, el *front-end* de *Bootstrap 4* es compatible con todos los navegadores modernos (*Chrome, Firefox, Internet Explorer, Edge, Safari y Opera*).

Lo incluiremos en la *app* desde un *CDN* (*Content Delivery Network*):

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/
4.3.1/css/bootstrap.min.css">
```

```
<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
.min.js"></script>
```

```
<!-- Popper JS -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/u
md/popper.min.js"></script>
```

```
<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootst
rap.min.js"></script>
```

Simultáneamente, para realizar las llamadas al servidor de los archivos *GeoJSON* lo haremos a través de un *Web Feature Service (WFS)*.

11. Web Feature Service (WFS)

Estándar que define operaciones Web de interface para la consulta y edición de entidades geográficas vectoriales.

Mientras que, para realizar las llamadas al servidor de los archivos *GeoTIFF* emplearemos una llamada *Web Map Service (WMS)*.

12. Web Map Service (WMS)

Este estándar internacional define un "mapa" como una representación de la información geográfica en forma de un archivo de imagen digital.

Y para cohesionarlo todo emplearemos *jQuery* y *Ajax*.

13. Ajax

Es un conjunto de desarrollo web técnicas que usan muchas tecnologías web en el lado del cliente para crear asíncronos aplicaciones web.

14. JQuery

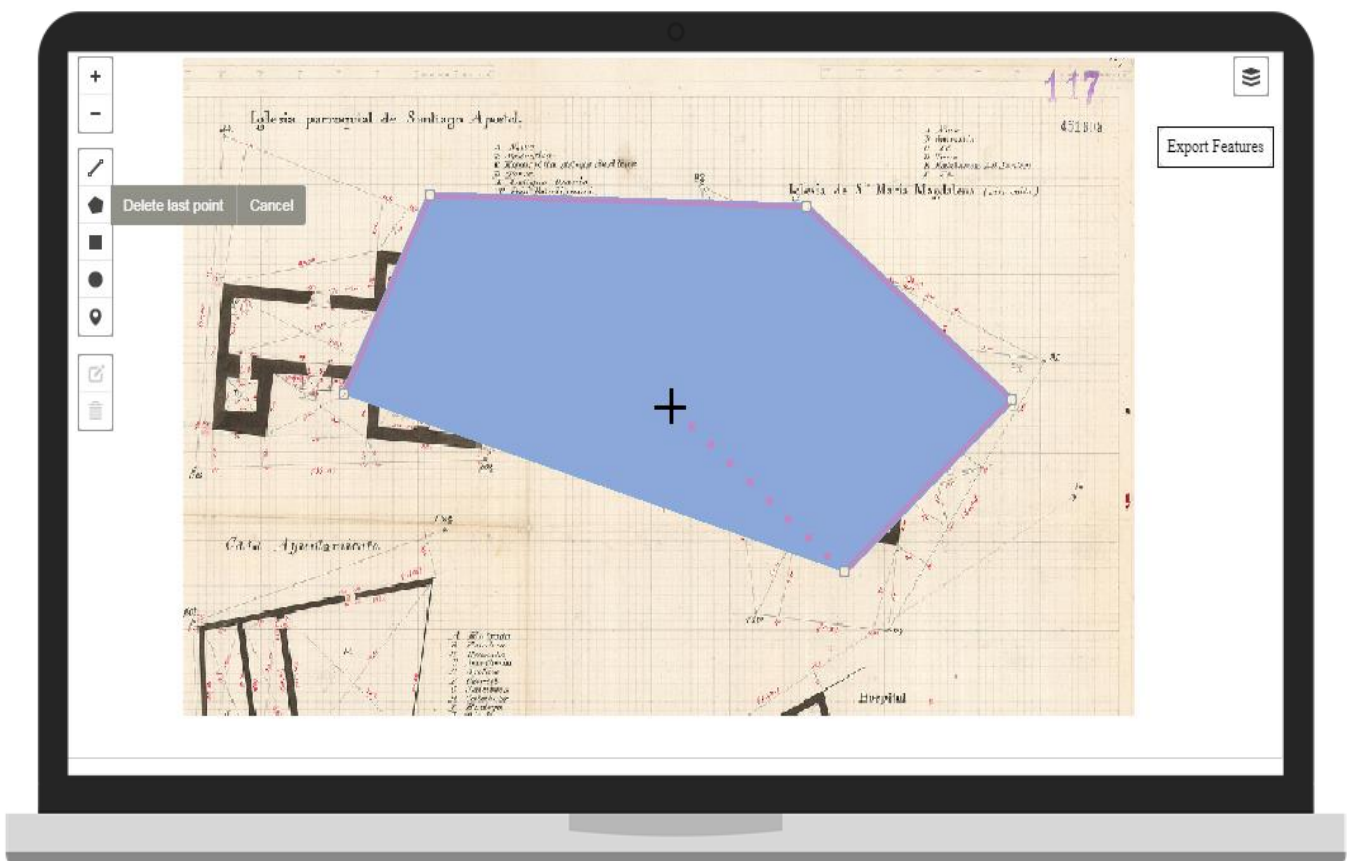
Es una biblioteca multiplataforma de JavaScript.

Conjuntamente, para interactuar con el Público y agilizar el proceso de digitalización se propone utilizar [Leaflet](#) como herramienta y el mapeo colaborativo como motor.

Por lo que serviremos a la aplicación Web los ficheros [GeoTIFF](#) como *WMS*.

Posteriormente, agregaremos la barra de herramientas de dibujo insertándolas en el mapa a través del *plugin* de [Leaflet L.Draw](#).

Una vez el usuario ha finalizado la digitalización puede exportarlo a [GeoJSON](#) y enviarlo para que se cargue en el [Geoserver](#) y visualizarlo en la Aplicación Web.



Para visualizar el total de los centroides de los polígonos calculados con anterioridad visiblemente de una forma más armoniosa cargaremos el *plugin* de [Leaflet L.MarkerClusterGroup](#).

Utilidad y su carácter innovador

La utilidad práctica de este proyecto yace en su carácter renovador pues lo que se pretende es actualizar los planos obsoletos de edificaciones históricas a través del uso de nuevas tecnologías de manera que se pueda acceder a ellos de manera más fácil desde cualquier dispositivo y en cualquier momento y lugar empleando una aplicación web *responsive* y totalmente gratuita.

En consecuencia, gracias a esta innovación el público en general podrá involucrarse en la digitalización de los planos históricos y acceder a ellos con mayor facilidad.

Además, muchos de estos planos de los que se dispone desde hace mucho tiempo, no están digitalizados, por lo que resultan realmente atractivos para los grandes productores de mapas como *google maps* o *openstreetmaps* que no disponen de estos datos, por lo que seríamos pioneros en ofrecer una visualización interactiva de ellos.

Conclusiones y líneas de futuro

Gracias a la versatilidad de las aplicaciones web *responsive*, es posible llegar a un mayor número de público y que sea compartido por el máximo número de personas.

Debido a este alcance, junto al gran auge del mapeo colaborativo en la actualidad, véase proyectos como *openstreetmaps* es posible agilizar mucho el proceso de digitalización, con unos pocos clics y un fichero estándar como *GeoJSON*.

Se espera iniciar el proyecto con los datos proporcionados por el IGN, de las provincias de Badajoz, Ciudad Real, Guadalajara, Madrid, Segovia y Toledo, pero es posible escalarlo a nivel nacional, permitiendo a otras administraciones subir sus planos digitalizados o incluso a los propios usuarios, lo que aumentaría de forma significativa la base de datos existente y la riqueza de esta en cuanto información arquitectónica histórica.

Presupuesto

Se pretende inicializar el proyecto con el Crédito de 300 \$ gratuito que ofrece *Google Cloud Platform* durante los próximos 12 meses.

Google App Engine:

Recurso	Unidad	Coste de la unidad
vCPU	Por hora de núcleo	\$0.0579
Memoria	Por GB por hora	\$0.0078
Disco persistente	Por GB al mes	\$0.0400

Google Cloud Storage:

Multi-Regional Storage (por GB al mes)	Regional Storage (por GB al mes)	Nearline Storage (por GB al mes)	Coldline Storage (por GB al mes)
\$0.026	\$0.02	\$0.01	\$0.007

Gastos en Publicidad:

Google Ads:

Google subvenciona 75€ para tu campaña de *Google Ads* al invertir 25€.

Invertiremos 25 € en *Facebook ads*.

Invertiremos 25 € en anuncios por *LinkedIn*.

Costes de personal:

Frontend developer: 16 euros b/h.

Backend developer: 16 euros b/h.

Gis Developer: 16 euros b/h.

Personal Administrativo: 12 euros b/h.

Bibliografía

Se dispone de numerosa documentación para poder llevar a cabo el proyecto:

<http://centrodedescargas.cnig.es/CentroDescargas/index.jsp>

<https://www.w3schools.com/js/default.asp>

<https://www.w3schools.com/bootstrap4/default.asp>

<https://www.w3schools.com/jquery/default.asp>

https://www.w3schools.com/js/js_ajax_intro.asp

<https://leafletjs.com/reference-1.4.0.html>

<https://cloud.google.com/>