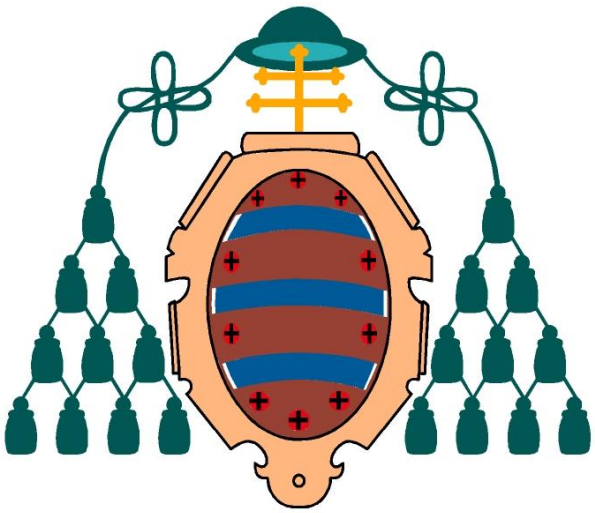


UNIVERSIDAD DE OVIEDO

EPM
ESCUELA
POLITECNICA DE
MIERES

PUBLICACIÓN DE CARTOGRAFÍA EN INTERNET

TRABAJO FIN DE MÁSTER

Máster en Teledetección y Sistemas de Información Geográfica

Autor: Enrique Álvarez Díaz

Tutor académico: Celestino Ordóñez

Índice

Imágenes

1. RESUMEN.

En este proyecto se desarrollará una aplicación web capaz de gestionar y monitorizar una serie de archivos. La finalidad de este documento es obtener un visor cartográfico dedicado al senderismo en el concejo de Pravia, Asturias.

Para la creación de este Sistema de Información Geográfica, se utilizará el software libre *QGIS* y la biblioteca de Javascript *OpenLayers 3* bajo una licencia BSD para mostrar mapas interactivos en navegadores web.

Este visor permitirá conocer la situación del concejo de Pravia, y obtener la ubicación e información acerca de las rutas y los puntos de interés de este municipio asturiano.

2. ABSTRACT.

The purpose of this project is to create a web application able to manage a series of files. The tarjet of this document is to obtain a cartographic display dedicated to hiking in the council of Pravia, Asturias.

For the creation of this GIS, it will be used the free software *QGIS* and the JavaScript library *OpenLayers 3* under a BSD license used to display interactive maps in web browsers.

This viewer will reveal the situation of the council of Pravia, and get the location and information about the routes and points of Interest of this Asturian town.

3. INTRODUCCIÓN.

El principado de Asturias está situado en la parte noroccidente de España. Tiene una extensión de 10.654 kilómetros cuadrados, y tiene algo más de un millón de habitantes. Es uno de los principales destinos turísticos de España, debido a la riqueza de su paisaje.

Sin tener que realizar grandes desplazamientos, se puede disfrutar de la playa y la montaña en el mismo día. Gracias a esto, uno de los pasatiempos más realizados por los turistas en sus visitas a Asturias y también por los autóctonos es el senderismo, actividad sobre la que se desarrolla el proyecto.

Para comprender este proyecto, se han de conocer antes unos conceptos básicos sobre la publicación de cartografía en Internet.

3.1. CONCEPTOS BÁSICOS SOBRE LA PUBLICACIÓN DE CARTOGRAFÍA EN INTERNET.

3.1.1. SISTEMAS DE INFORMACIÓN GEOGRÁFICA (S.I.G.)

Para el Instituto Geográfico Nacional, la acepción principal es la de SIG como proyecto, un Sistema de Información que gestiona Información Geográfica, es decir información que está georreferenciada.

La definición más extendida de Sistema de Información Geográfica o SIG, con pequeñas variaciones, es la establecida por el Departamento de Medio Ambiente (DoE). La cual podemos sintetizar diciendo que un SIG es un:
«Conjunto integrado de medios y métodos informáticos, capaz de recoger, verificar, almacenar, gestionar, actualizar, manipular, recuperar, transformar, analizar, mostrar y transferir datos espacialmente referidos a la Tierra.»

Sin embargo se cree que, tal y como sostienen Burrough y Bouillé, un SIG debe verse también como un modelo del mundo real, por lo que se podría definir como:

«Modelo informatizado del mundo real, en un sistema de referencia ligado a la Tierra para satisfacer unas necesidades de información concretas».

Los SIG almacenan información cartográfica, pero no solamente almacenan la localización de esos datos en el espacio (lo que se conoce como georreferenciación), también la relación de unos elementos con otros.

A diferencia de un mapa tradicional, un SIG puede centrarse sobre una determinada región del mapa,seleccionar distintos elementos, tener varias capas con diferentes informaciones y que se superpongan (usos del suelo, base topográfica, etc.)

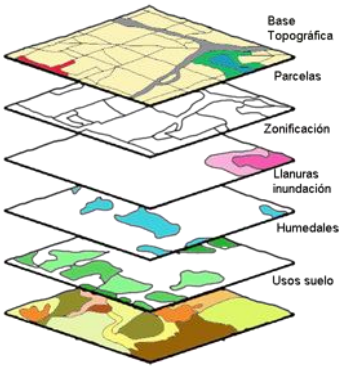


Imagen 1: Representación de las distintas capas de un SIG.

De esta manera, con un SIG se puede catalogar y digitalizar bienes tanto de la administración pública (infraestructuras, edificios, etc.) como entidades privadas (rutas comerciales, comercios, etc.)

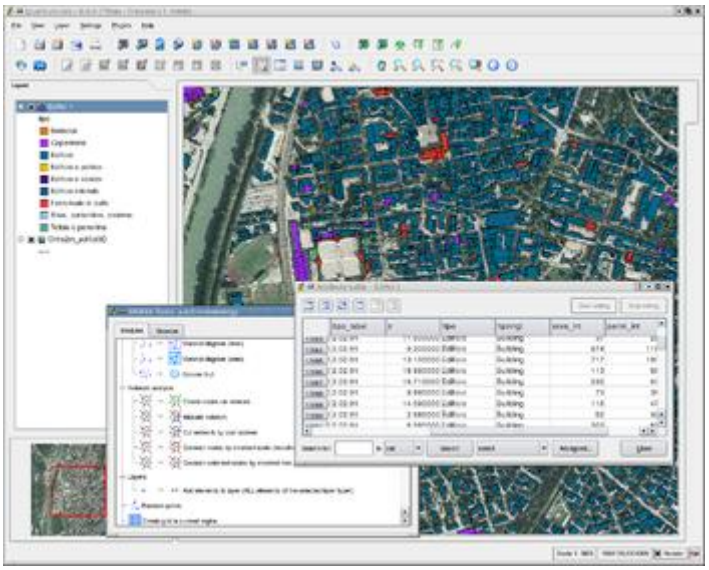


Imagen 2: Representación de un SIG realizado con el software gratuito QGIS.

Realmente cuando se habla de un SIG, no se puede hablar solamente de un programa o una serie de análisis. Los SIG son un conjunto de elementos que permiten realizar consultas y crear nuestra propia información.

Un SIG se compone de los siguientes elementos:

- Software. Incluye tanto los programas SIG, como los de tratamientos de datos.

- Hardware. Las capacidades del equipo informático afectan a la realización de los procesos y las tareas.
- Datos. La disponibilidad y precisión que tengan los datos, afecta al resultado del análisis.
- Procedimientos. El análisis requerirá métodos bien definidos y consistentes.
- Usuario o Recurso Humano. Es el componente más importante y quien debe desarrollar los procedimientos y realizar las tareas.

Imagen 3: Componentes de un SIG.

3.1.2. CARTOGRAFÍA EN INTERNET.

La cartografía se encuentra en una constante evolución gracias a las nuevas tecnologías, en especial Internet.

Según Dodge y Kitchin, hay tres grandes evoluciones basadas en Internet:

"*Primero, los avances tecnológicos que han experimentado los Sistemas de Información Geográfica y los GPS; segundo, los nuevos métodos de la interactividad en la visualización geográfica y, por último, las actitudes de la propia cartografía que se consolida como un instrumento objetivo y científico. Además, hay que tener muy en cuenta que la web rompe con las barreras de producción, distribución y accesibilidad a los mapas, pudiendo elaborar mapas según la propia demanda*" (DODGE M. y KITCHIN,(2000))

Como se observa en el gráfico mostrado a continuación, en el año 2009 la consulta de mapas en Internet ocupaba el segundo lugar de las actividades realizadas en la red, por detrás de la lectura de noticias de actualidad.

En el año 2010 descendió casi un 10%, pero seguía ocupando el tercer lugar tras la búsqueda de noticias y la visualización de vídeos.

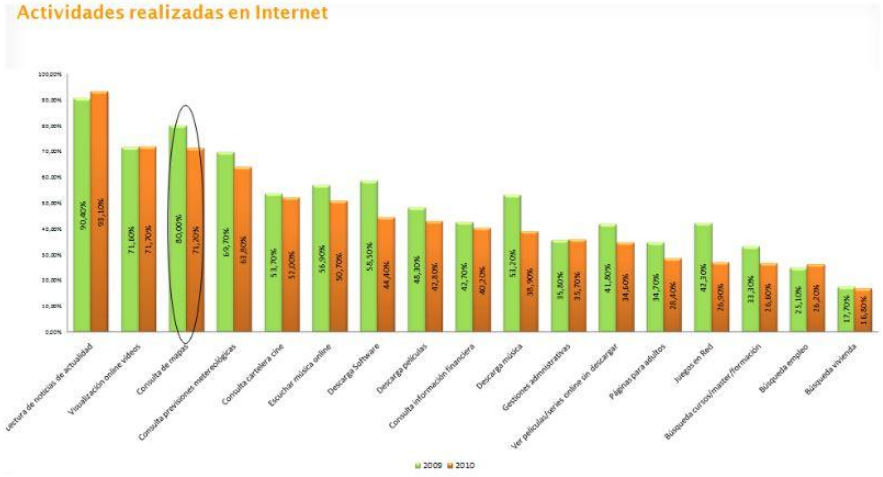


Imagen 4: Actividades realizadas en Internet en los años 2009/2010.

Gracias a este gráfico se puede entender la importancia que tiene la Información Geográfica dentro del mundo virtual, y como está creciendo con los nuevos avances.

Unos de los ejemplos más claros que permiten interactuar con la cartografía en Internet son **Google Earth** y **World Wind**. Son dos productos gratuitos basados en el formato XML y SVG. Estos programas tienen diferentes versiones, que ofrecen más o menos herramientas dependiendo de los archivos que se instalen en el ordenador. Ambos son programas geoespaciales, donde se puede consultar lugares, rutas de transporte, hidrografía, modelos digitales e imágenes aéreas.

Las imágenes están disponibles en diferentes tiempos, espectros y resoluciones, las cuales vienen dadas por el sensor de cada satélite. Ambos programas tienen foros en Internet en los que tratan discusiones y se proporcionan soluciones. Como diferencia, Google Earth incluye el 3D en determinadas ciudades.

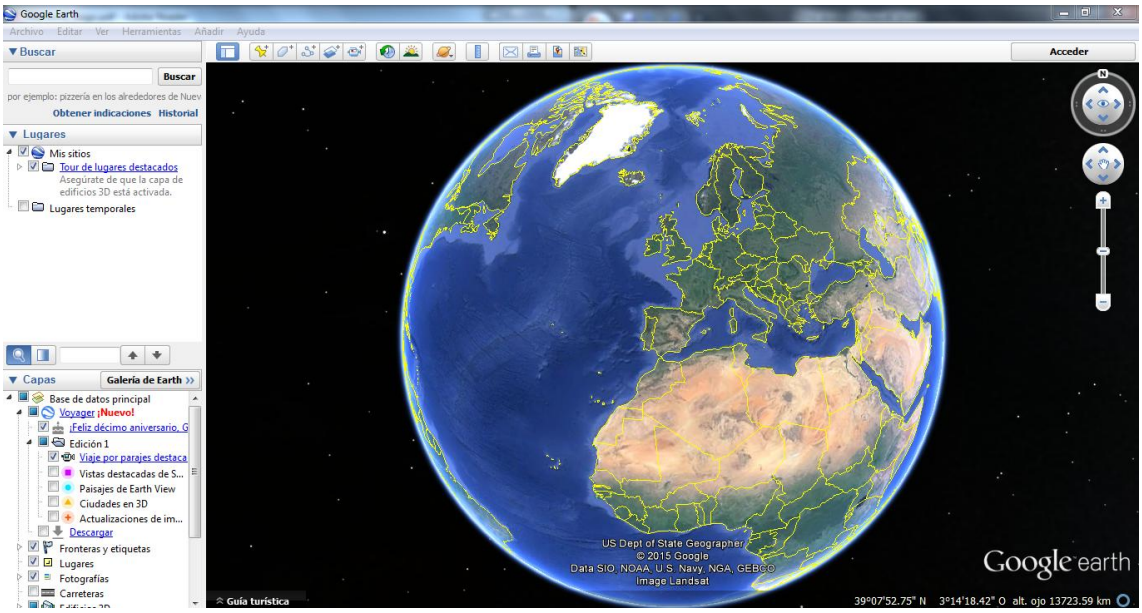


Imagen 5:Interfaz de Google Earth.

3.1.3. SOFTWARE LIBRE.

La definición de software libre viene estipulada por los criterios que se han de cumplir para que un programa se considere libre.

El "software libre" es aquel que respeta la libertad de los usuarios y la comunidad. Para que se entienda, significa que los usuarios tienen la libertad de ejecutar, distribuir, estudiar, modificar y mejorar el software. Para que se entienda, el concepto de "software libre" es una cuestión de libertad, no de precio.

Decimos que un programa es software libre si los usuarios cumplen klas cuatro libertades esenciales:

- La libertad de ejecutar el programa como se desea, con cualquier propósito. (Libertad 0)
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que cada uno quiera. (Libertad 1)
- Libertad de redestribuir copias para demás gente. (Libertad 2)
- Libertad de distribuir copias de sus versiones modificadas. Esto permite ofrecer a la comunidad la oportunidad de disponer de las modificaciones del software. (Libertad 3)

Si un programa no cumple estas cuatro libertades, no se puede considerar que sea un software libre.

Imagen 6: Libertades del Software libre.

3.2. PROGRAMACIÓN.

3.2.1. CONCEPTOS BÁSICOS.

- **API.**

Es un conjunto de protocolos y herramientas usado por desarrolladores. Su abreviatura viene del término en inglés *Application Programming Interfaces* (Interfaces de programación de aplicaciones).
«Una API es una especificación formal sobre cómo un módulo de un software se comunica o interactúa con otro.» (Benjy Weinberger, en su blog "10 Conceptos Técnicos que todo el mundo debería saber").

En otras palabras, las API son un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos. Las API simplifican en gran medida el trabajo de un creador de programas, ya que no tiene que escribir códigos desde cero. Es decir, las API permiten al informático usar funciones predefinidas para interactuar con el sistema operativo o con otro programa.

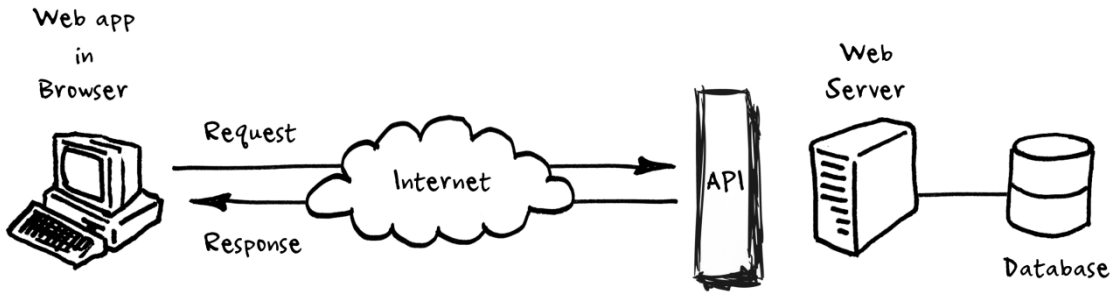


Imagen 7: Esquema del funcionamiento de una API.

- **OpenLayers.**

OpenLayers es una biblioteca de de JavaScript de código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores web.

Las Licencias BSD son llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sitemas operativos BSD (*Berkeley Software Distribution*). El autor, bajo esas licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU/GPL con quienes son compatibles. Puede argumentarse que esta licencia asegura "verdadero" software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre.

OpenLayers ofrece una API para acceder a diferentes fuentes de información cartográfica en la red: Web Map Services, Mapas comerciales (Google Maps, Bing, Yahoo), Web Feature Services, distintos formatos vectoriales, mapas de OpenStreetMap, etc.
Los mapas se pueden dotar de diversos controles con capacidades de zoom, panning, medida de distancias y muchas otras herramientas.

Este proyecto se ha realizado con OpenLayers3, algunas de las mejoras que trae esta versión son las siguientes:

- Mejora del tratamiento de los datos vectoriales, los elementos vectoriales se redibujan ahora cada vez que el usuario interactúa con el mapa, permitiendo una mayor calidad y rapidez en la visualización.
- Mejora la interoperabilidad con formatos de intercambio de datos geográficos como KML, GeoJSON, TopoJSON, GPX…
- Inclusión de nuevos formatos como WebGL que posibilitan la capacidad de mostrar datos en formato 3D y toda la potencia de html5.
- Mayor sencillez a la hora de personalizar nuestros mapas mediante CSS.
- Está adaptado para mostrarse en pantallas de alta densidad de píxeles o pantallas retina.
- Nuevas herramientas como la posibilidad de renderizar una capa con precomposición o postcomposición.

3.2.2. LENGUAJES DE PROGRAMACIÓN.

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como los ordenadores. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

- **HTML.**

HTML significa *HyperText Markup Language*, es el lenguaje que se emplea para el desarrollo de páginas de internet. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc.
Podríamos decir que HTML sirve para crear páginas web, darles estructura y contenido.

Los documentos HTML son ficheros de texto plano (también conocidos como ASCII) que pueden ser editados con cualquier editor de texto (como el "Bloc de notas" de Windows). También se podría utilizar cualquier programa procesador de textos (como StarWriter o Word), guardando el documento como "solo texto". El nombre de los ficheros escritos en lenguaje HTML suelen tener la extensión `html` o `htm`.

El lenguaje HTML se basa en la sintaxis SGML (*Standard Generalized Markup Language*). Esto quiere decir que los diferentes elementos (párrafos, encabezamientos, tablas, listas, ...) de un documento para la WWW se señalan intercalando etiquetas que indican al navegador cómo debe mostrarlo.
Una etiqueta HTML consiste en un signo menor "<", un nombre de una directiva (orden o comando para el navegador), seguido de los parámetros o atributos y un signo mayor ">". Para cualquier etiqueta que indica un el inicio de un elemento hay otra de cierre que indica que esa orden ya no debe actuar sobre el texto que sigue (en algunas ocasiones no es necesario poner, o no existe, la etiqueta de cierre correspondiente).

Hay disponibles varios editores WYSIWYG (Composer, FrontPage, ...). WYSIWYG es acrónimo de *what yuo see is what you get* (lo que ves es lo que consigues), que significa que a la vez que se diseña el documento HTML estamos viendo su aspecto final.

```
<html>
<head>
   <title>Primera página</title>
</head>
<body>

</body>
</html>
```

Imagen 8:Estructura de una página web.

En la imagen superior se puede apreciar un ejemplo de estructura de una página web. En la primera línea aparece la etiqueta "<html>", que indica que comienza un documento HTML.
A continuación aparece "<head>" que es la cabecera, donde se puede poner información que no se ve en la pantalla del navegador. Aquí podría ir el título del documento,"<title>", que es lo que se verá como título de la ventana en el navegador que lo permita y cómo se conocerá la página en algunos buscadores.
Tras la cabecera viene el cuerpo "<body>", que es donde se coloca toda la información que se quiere mostrar en la pantalla principal del navegador.

- **JavaScript.**

Es un lenguaje que se puede utilizar tanto por profesionales como por quienes se inician en el desarrollo y diseño de páginas web. No requere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

Muchas personas confunden el JavaScript con el Java, pero ambos lenguajes son distintos y tienen sus características singulares. JavaScript tiene la ventaja de ser incorporado en cualquier página web, y puede ser ejecutado sin la necesidad de tener que instalar otro programa para visualizarlo.

El JavaScript es un lenguaje con infinidad de posibilidades, utilizado para crear pequeños programas que luego se insertan en una página web y en programas más grandes, orientados a objetos mucho más complejos.

Su sintaxis es similar a la usada en Java y C, al ser un lenguaje del lado del cliente este es interpretado por el navegador, no se necesita tener instalado ningún Framework. Es soportado por la mayoría de los navegadores como internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros.

- **CSS.**

El CSS es un lenguaje utilizado en la presentación de documentos HTML. Se podría decir que el lenguaje CSS sirve para organizar la presentación y aspecto de una página web. Este lenguaje es utilizado por parte de los navegadores web de internet y por los programadores web informáticos para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, etc.

La filosofía del lenguaje CSS se basa en intentar separar la estructura del documento HTML, de su presentación. Explicado de otra manera, la página web sería lo que hay debajo (el contenido) y CSS sería un cristal de color que hace que el contenido se vea de una manera o de otra.

Se basa en una serie de reglas que rigen el estilo de los elementos en los documentos estructurados, y que forman la sintaxis de las hojas de estilo. Cada regla consiste en un selector y una declaración, esta última va entre corchetes y consiste en una propiedad o atributo, y un valor separados por dos puntos.
El selector especifica que elementos HTML van a estar afectados por esa declaración, de manera que hace de enlace entre la estructura del documento y la regla estilístics en la hoja de estilo.

La declaración que va entre corchetes, es la información de estilo que indica cómo se va a ver el selector. En caso de que haya más de una declaración se usa punto y coma para separarlas.
Dentro de la declaración, la propiedad o atributo define la interpretación del elemento asignándose un cierto valor, que puede ser color, alineación, tipo de fuente, tamaño… es decir, se especifica qué aspecto del selector se va a cambiar.

- Estilos CSS.

La información CSS se puede proporcionar por varias fuentes, ya sea adjunto como un documento por separado o incorporado en el documento HTML, y dentro de estas posibilidades destacan tres formas de dar estilo a n documento web:

a) Hoja de Estilo Externa.
   Esta hoja se almacena en un archivo diferente al del archivo con el código HTML al estar vinculado a través del elemento "*link*", que debe ir situado en la sección "*head*". Es la manera de programar más eficiente, ya que separa completamente las reglas de formato para la página HTML de la estructura básica de la página.

b) Hoja de Estilo Interna.
La Hoja de Estilo Interna está incorporada a un documento HTML, a través del elemento "*style*" dentro de la sección "*head*", consiguiendo de esta manera deparar la información del estilo del código HTML.

c) Estilo en Línea.
El Estilo en Línea sirve para insertar el lenguaje de estilo directamente debtro de la sección "*body*" con el elemento "*style*". Sin embargo, este tipo de estilo no se recomienda pues se debe intentar siempre separar el contenido de la presentación.

Existen una serie de ventajas del lenguaje CSS:

- La principal ventaja de CSS sobre el lenguaje HTML o similar, es que el estilo se puede guardar completamente por separado del contenido siendo posible, por ejemplo, almacenar todos los estilos de presentación para una web de 10.000 páginas en un sólo archivo de CSS.
- CSS permite un mejor control en la presentación de un sitio web que los elementos de HTML, agilizando su actualización.
- Aumento de la accesibilidad de los usuarios gracias a que pueden especificar su propia hoja de estilo, permitiéndoles modificar el formato de un sitio web según sus necesidades, de manera que por ejemplo, personas con deficiencias visuales puedan configurar su propia hoja de estilo para aumentar el tamaño del texto.
- El ahorro global en el ancho de banda es notable, ya que la hoja de estilo se almacena en cache después de la primera solicitud y se puede volver a usar para cada página del sitio, no se tiene que descargar con cada página web. Por otro lado, quitando todo lenguaje de marcado en la presentación en favor del uso de CSS reduce su tamaño y ancho de banda hasta más del 50%.
- Una página puede tener diferentes hojas de estilo para mostrarse en diferentes dispositivos, como pueden ser impresoras, lectores de voz, o móviles.

3.3. CONCEJO DE PRAVIA.

Pravia es un concejo y una parroquia de la comunidad autónoma del Principado de Asturias y cuenta con una población que ronda los diez mil habitantes. Limita al norte con los concejos de Cudillero y Muro del Nalón, al este con Candamo y Soto del Barco, al oeste con Cudillero y Salas, y al sur con Candamo y Salas otra vez.

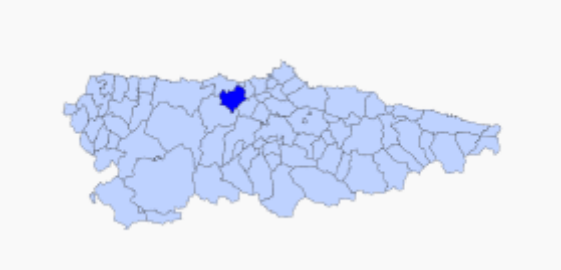


Imagen 9 : Situación del concejo de Pravia

Dentro del concejo de Pravia, se decidió incluir las siguientes rutas, áreas recreativas y formaciones montañosas o picos:

- Rutas.
  - Senda Verde.
  - Pravia - Santoseso.
  - Pravia - Pico Aguión – Pravia.
  - Ruta de los Marineros.
  - Ablanedo - Sierra del Pumar.
  - Pravia - Sierra del Pumar.
  - Ruta de Lin de Cubel.

- Áreas Recreativas.
  - Área recreativa del Mozabán.
  - Área recreativa de Quinzanas.
  - Área recreativa de Monteagudo.

- Picos.
  - Pico Cuetu.
  - Alto de la Cuesta.
  - Pico de Birabeche.
  - Santa Catalina.
  - Monteagudo.
  - Alto de la Peñona.
  - Pico Llanteiro.
  - Alto de la Casafría.
  - La Peralina.

- Alto de la Cachuela.
- Pico de la Cruz.
- Picultaos.
- Alto del Miñande.
- Pico de Andolinas.
- Pico de Comals.
- Pico Lin de Cubel.
- Pico Carceda.
- Peña de la Cabra.

4. OBJETIVOS.

Este proyecto se basa en la creación de un visor cartográfico donde se representarán las diferentes rutas, tanto a pie como en bicicleta, del concejo de Pravia. Para ello se trabajó principalmente con dos software libres, *QGIS* y la biblioteca de JavaScript *OpenLayers3*.

También aparecen representadas los principales montes o picos y algunas de las áreas recreativas de la zona. Todos los elementos vienen descritos con sus características. Por ejemplo, en las rutas se describe su dificultad, longitud, modo de realización, etc. En los montes, su altura; en las áreas recreativas su localización, etc.

El resultado final es un visor por el que se podrá navegar libremente, alternando los diferentes mapas que se tienen como mapas base (satélite, físico, callejero e híbrido) para obtener el que mejor se adecúe a nuestras exigencias (si lo que se quiere ver es el callejero, o el físico para tener en cuenta el relieve por ejemplo).

A la derecha, se encuentra una leyenda con todos los elementos representados donde se podrá activar o desactivar su visualización, para observar solo la ruta o rutas que queremos conocer.

5. PROCEDIMIENTO.

En este apartado se procederá a explicar, paso por paso, la creación de este visor cartográfico.

5.1. QGIS.

- Características.

Antes de redactar el trabajo realizado con QGIS, analizaremos las principales características de este software.

También conocido como *Quantum GIS* es Sistema de Información Geográfica de código libre. Permite manejar formatos ráster (GRASS GIS, GeoTIFF, TIFF, JPG, etc.) y vectoriales (Shapefile, MapInfo, etc.), así como bases de datos.

QGIS está desarrollado en C++, y permite la integración de plugins desarrollados tanto en C++ como en Python.

QGIS es capaz de trabajar en cualquiera de los **sistemas operativos**: GNU/Linux, BSD, Unix, Mac OSX, Windows y Android, funcionando de una manera similar en todos ellos.
Opera bajo la licencia **GNU GPL**. El software de QGIS puede ser modificado libremente de tal manera que se pueda realizar diferentes y más especializadas funcionalidades.

Una de las grandes versatilidades de este software es su facilidad de interconexión con muchas bases de datos geoespaciales.

- Trabajo con QGIS.

La primera parte de este proyecto, se desarrolló con QGIS. El primer paso fue digitalizar los picos más representativos del concejo, y algunas de las áreas recreativas.

Para ello, se instaló un complemento propio de QGIS llamado *OpenLayers Plugin*, el cual nos permite utilizar como base distintos mapas. Se utilizó como mapa base para la digitalización uno de los que nos ofrece *OpenStreet Map*.

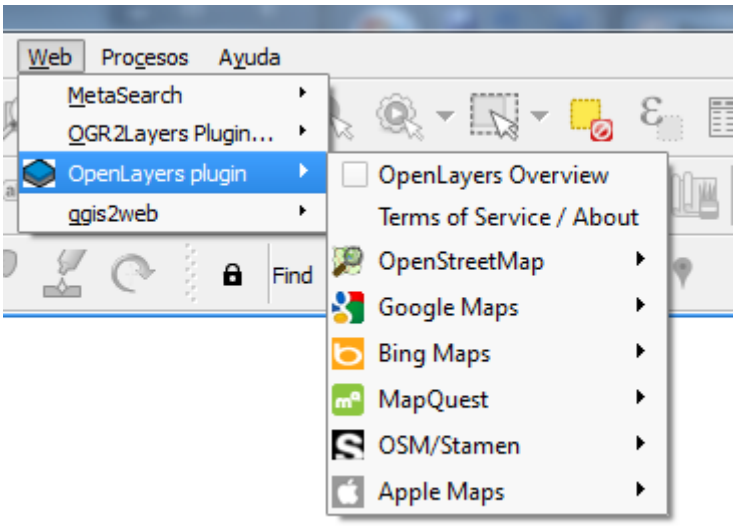


Imagen 10: Complemento de OpenLayers y variedad de mapas base que ofrece.

El mapa base, tiene coordenadas WGS84/Pseudo Mercator (EPSG:3857) por lo que los Shape Files que se creen para el proyecto, como las formaciones montañosas y las áreas recreativas, tendrán también ese sistema de coordenadas.

Se escogió este SRC (Sistema de Coordenadas de Referencia), ya que aplicaciones web de cartografia como Google Maps, OpenStreetMap o Bing Maps utilizan internamente este sistema de proyección.

Concretamente emplean una variante que supone que la superficie del planeta es esférica en vez de la forma exacta, elipsoidal, para simplificar los cálculos.

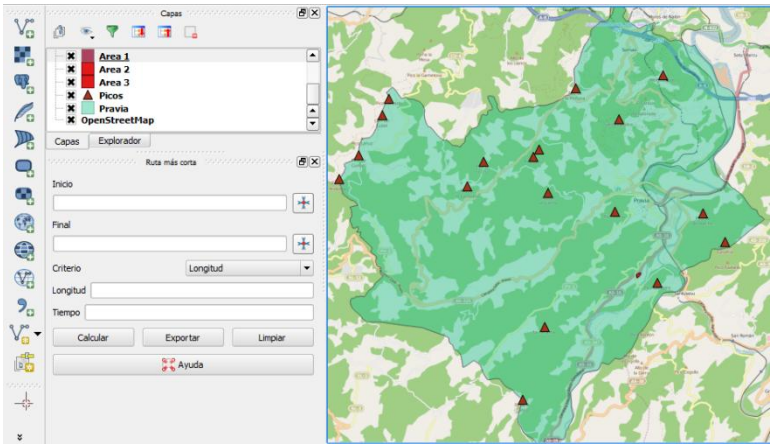


Imagen 11: Digitalización de las áreas recreativas y las montañas más representativas del concejo, con el OpenStreetMap como mapa base.

Como se observa en la imagen, también se decidió superponer al mapa base un ShapeFile del concejo de Pravia para observar cuales de las rutas tienen parte de ellas fuera del concejo. Para obtener solo el concejo se descargó de la página del IGN el callejero de CartoCiudad, y

desde QGIS se seleccionó por atributo el municipio de Pravia y se guardó como Shape File con coordenadas WGS84/Pseudo Mercator.

A los shapes de los picos y las áreas recreativas, se les añadieron una serie de atributos para que se nos muestren al visualizarlos en el visor cartográfico. En los picos, por ejemplo, se agrupan todos en una sola capa y se crearon tres campos que son: Nombre, Altura (en metros), características y el Id. Este último lo crea automáticamente QGIS y es autonumérico.

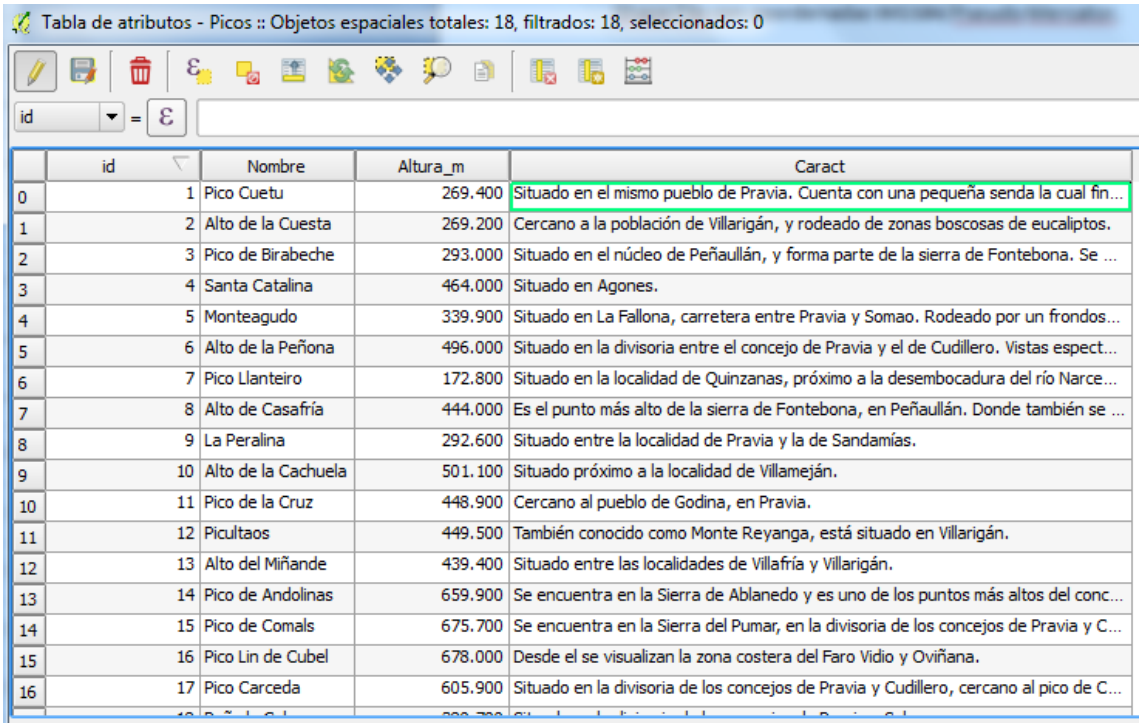


Tabla de atributos - Picos :: Objetos espaciales totales: 18, filtrados: 18, seleccionados: 0

id = ε

	id	Nombre	Altura_m	Caract
0	1	Pico Cuetu	269.400	Situado en el mismo pueblo de Pravia. Cuenta con una pequeña senda la cual fin...
1	2	Alto de la Cuesta	269.200	Cercano a la población de Villarigán, y rodeado de zonas boscosas de eucaliptos.
2	3	Pico de Birabeche	293.000	Situado en el núcleo de Peñaullán, y forma parte de la sierra de Fontebona. Se ...
3	4	Santa Catalina	464.000	Situado en Agones.
4	5	Monteagudo	339.900	Situado en La Fallona, carretera entre Pravia y Somao. Rodeado por un frondos...
5	6	Alto de la Peñona	496.000	Situado en la divisoria entre el concejo de Pravia y el de Cudillero. Vistas espect...
6	7	Pico Llanteiro	172.800	Situado en la localidad de Quinzanas, próximo a la desembocadura del río Narce...
7	8	Alto de Casafría	444.000	Es el punto más alto de la sierra de Fontebona, en Peñaullán. Donde también se ...
8	9	La Peralina	292.600	Situado entre la localidad de Pravia y la de Sandamías.
9	10	Alto de la Cachuela	501.100	Situado próximo a la localidad de Villameján.
10	11	Pico de la Cruz	448.900	Cercano al pueblo de Godina, en Pravia.
11	12	Picultaos	449.500	También conocido como Monte Reyanga, está situado en Villarigán.
12	13	Alto del Miñande	439.400	Situado entre las localidades de Villafría y Villarigán.
13	14	Pico de Andolinas	659.900	Se encuentra en la Sierra de Ablanedo y es uno de los puntos más altos del conc...
14	15	Pico de Comals	675.700	Se encuentra en la Sierra del Pumar, en la divisoria de los concejos de Pravia y C...
15	16	Pico Lin de Cubel	678.000	Desde el se visualizan la zona costera del Faro Vidio y Oviñana.
16	17	Pico Carceda	605.900	Situado en la divisoria de los concejos de Pravia y Cudillero, cercano al pico de C...

Imagen 12: Tabla de atributos de la capa "Picos".

Para las áreas recreativas, se creó una capa para cada una de ellas, y los atributos que poseen son el de Nombre y Características.

Una vez creadas estas dos capas se pasó a las que realmente son las más importantes, las rutas de senderismo.

Puesto que algunas son rutas muy extensas y era muy costosa su digitalización, se descargaron desde una página de Internet llamada "Wikiloc", para después modificar sus atributos en relación a lo que queremos mostrar.

Imagen 13: Ejemplo de una de las rutas descargadas de la página "Wikiloc"

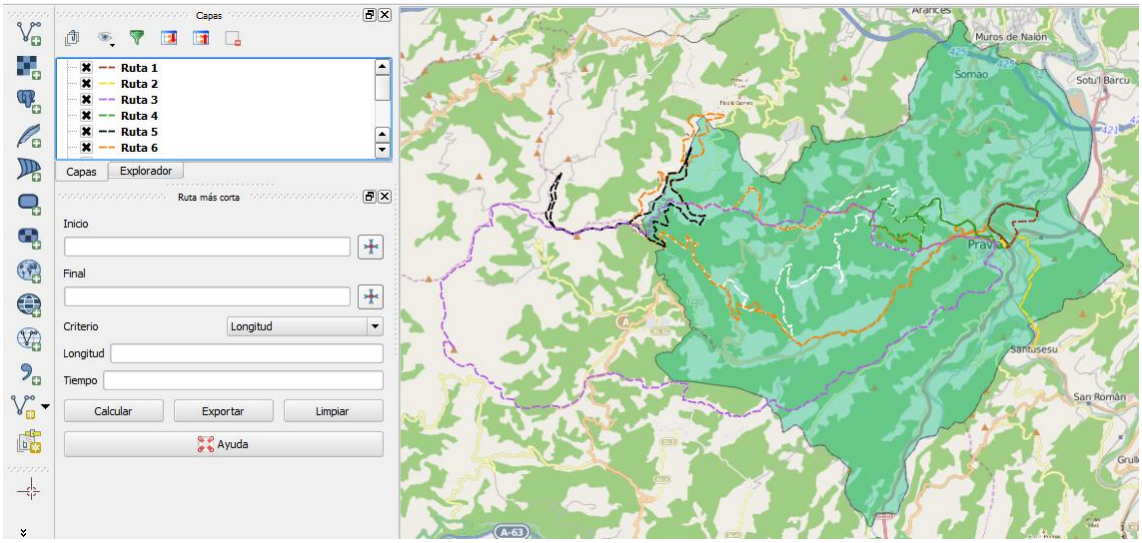


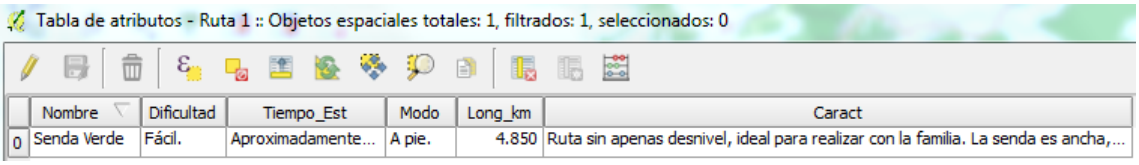
Imagen 14: Representación de las rutas descargadas e importadas a QGIS.

Una vez representadas las rutas en QGIS se eliminaron los atributos que traía por defecto, ya que no se ajustaban a lo que se busca en este proyecto, y se crearon unos de mayor utilidad para obtener información acerca de las rutas.

Los nuevos atributos creados son los siguientes:

- Nombre. Donde figura el nombre por el que se conoce a la ruta.
- Dificultad. Comprende tres niveles: fácil, moderada y difícil.
- Tiempo_Est. Tiempo de realización de la ruta, viene expresado en horas.

- Modo. Es la manera en la que se puede realizar la ruta, por lo general a pie o en bicicleta.
- Longitud. Longitud de la ruta expresada en kilómetros.
- Características. Breve descripción de la ruta, con sus rasgos más significativos.



	Nombre	Dificultad	Tiempo_Est	Modo	Long_km	Caract
0	Senda Verde	Fácil.	Aproximadamente...	A pie.	4.850	Ruta sin apenas desnivel, ideal para realizar con la familia. La senda es ancha,...

Imagen 15: Tabla de atributos de una de las rutas.

Una vez importadas las rutas en QGIS y editadas sus tablas de atributos, guardamos cada una de las capas en formato "*GeoJSON*" que es más adecuado para las publicaciones en Internet puesto que son menos pesadas y la página se carga más rápido, y con SRC WGS84 / Pseudo Mercator.

GeoJSON es un formato de intercambio geoespacial basado en JSON (Javascript Object Notation).

Permite codificar varias estructuras geográficas, y soporta los siguientes tipos de geometría:

- Point.
- LineString.
- Polygon.
- MultiPoint.
- MultiLineString.
- MultiPolygon.
- GeometryCollection.

Además de geometrías soporta *Features* que es una geometría más propiedades adicionales, y colecciones de Features.

Una vez transformadas todas las capas al formato GeoJSON, se procede a la fase de programación, ayudándonos de la biblioteca que ofrece OpenLayers3.

5.2. OPENLAYERS.

El objetivo de este proyecto es crear un visor cartográfico en Internet, por lo que el primer paso será el de crear una página HTML donde se guardarán todos los elementos de los que disponemos.

El siguiente paso es el de añadir un elemento " " donde se mostrará el mapa. Dicho elemento, deberá tener su atributo "Id".

Al abrir el código fuente del mapa, se observa que es muy sencillo. Lo que aquí ocurre es que para que el archivo HTML quedase más limpio, se creó otro archivo "pravia.js" donde viene desarrollado el código original del mapa creado. Este archivo se vincula al mapa.

- MAPAS Y CAPAS.

Una vez creada la página HTML, se añade una pequeña rutina javascript que utilice las clases de OpenLayers para mostrar el mapa. Para utilizar las clases de OpenLayers se debe cargar la librería en memoria. Esto se hace añadiendo un link al fichero "OpenLayers.js" en la cabecera de la página HTML.

```
<script type="text/javascript" src="http://www.openlayers.org/api/2.12/OpenLayers.js"></script>
```

Imagen 16: Referencia al script de la librería de OpenLayers en la página HTML.

Una vez que ya está cargada la librería en memoria, se puede acceder a las clases de OpenLayers. Para mostrar una capa, se utiliza la clase *OpenLayers.Map*, a la que se le añaden una o más capas. Se mantiene la proyección en la que se está trabajando, el centro del mapa y el nivel del zoom en cada momento. El objeto *Map* también posee una serie de controles como hacer zoom, o panning.

```
var map = new OpenLayers.Map({
    projection: new OpenLayers.Projection("EPSG:3857"),
    controls: []
});
```

Imagen 17: Función OpenLayers.Map y SRC escogido.

La primera tarea, es crear un objeto *OpenLayers.Map*. La parte visible del mapa son las capas que se van añadiendo al mismo. OpenLayers dispone de capas para acceder a los Web Map Services.

Para este proyecto se escogieron las capas de Google, ya que son las que mejor se adaptan a nuestro objetivo y proporcionan bastante información. Las capas de Google tienen su propia clase, "*OpenLayers.Layer.Google"*.

Para cargar las cuatro capas de Google (Física, Streets, Satélite e Híbrida) debemos importar un fichero JavaScript de maps.google.com.
Después se añaden las capas con el método addLayers() de la clase *OpenLayers.Map*, y se añade el selector de capas.

Las cuatro capas de Google que se han cargado son: Relieve, Híbrido, Satélite y Callejero.

Una de las razones de escoger Google Maps como mapa base es que este servicio de Google permite acceder a mapas en diferentes formatos tanto desde la web de Google Maps como desde páginas de usuarios.

Google Maps utiliza como SRC el Mercator-Esférico, que utiliza coordenadas en metros y también es usado por OpenStreets Maps, Bing Maps o Yahoo Map. De todas maneras utilizando el API de Google directamente, se pueden utilizar coordenadas de latitud y longitud de acuerdo al sistema de proyección WGS84/Pseudo Mercator, el cual se había utilizado en QGIS.

- CONTROLES.

Los controles que nos ofrece Open Layers sirven para interactuar con el mapa. Permiten mover el mapa, hacer zoom, zoom por selección, impresión de pantalla, etc.

Todos estos controles provienen de la clase *OpenLayers.Control*, y se añaden al objeto *OpenLayers.Map* el cual tiene una propiedad que guarda la lista de controles del mapa. Para añadir un control al mapa se puede hacer mediante el método OpenLayers.Map.addControl() o bien directamente en el constructor de OpenLayers.Map.

```
var barraHerramientas = new OpenLayers.Control.Panel();
barraHerramientas.addControls([
    zin,
    zou,
    pan,
    zoomVentana,
    print
]);
map.addControl(barraHerramientas);
```

Imagen 18: Código utilizado para añadir los diferentes controles al mapa.

- Visualizador de capas.

El visualizador de capas nos permite seleccionar lo que se quiere ver. Por ejemplo, en el apartado de Mapa Base se puede alternar el mapa de Google que se adecue más a lo que se está buscando. En este apartado, siempre habrá una de las capas de Google seleccionada.

Los otros apartados son los de rutas, áreas recreativas y picos. Las rutas, áreas o picos se pueden activar o desactivar a gusto del usuario, en el caso por ejemplo de que solo se quiera consultar una ruta y las demás nos dificulten ver su recorrido.

Imagen 19: Visualizador de capas o leyenda.

```
var layerTree = new Ext.tree.TreePanel({
    title: '',
    renderTo: "layers",
    root: layerList,
    border: false,
    bodyStyle: {"background-color": "#CEE3F6","opacity":".6","z-index": "2"}
});
```

Imagen 20: Código del visualizador de capas.

- Zoom.

Disponemos de dos botones, con los que se podrá acercarse o alejarse del mapa libremente.

Imagen 21: Botones de Zoom + y Zoom –

```
var zin = new OpenLayers.Control.ZoomIn ({
    title: 'Zoom M\u00e1s'
    });

var zou = new OpenLayers.Control.ZoomOut ({
    title: 'Zoom Menos'
    });
```

Imagen 22: Códigos de las funciones Zoom+ y Zoom-

- Zoom Ventana.

Esta herramienta sirve para realizar el zoom en una zona determinada, o que se vaya quiera consultar algo en concreto.

Imagen 23: Zoom ventana.

```
var zoomVentana = new OpenLayers.Control.ZoomBox({
    title: 'Hacer Zoom ventana',
    zoomOnClick: false,
    displayClass:'ZoomBox'
    });
```

Imagen 24 : Código de la función Zoom ventana.

- Arrastrar el mapa.

Esta herramienta sirve para poder navegar libremente por el mapa, pudiendo alternarla por ejemplo con los botones del zoom.

Imagen 25: Herramienta de movimiento.

```
var pan =  new OpenLayers.Control.Navigation({
    title: 'Arrastrar el mapa',
    mouseWheelOptions: {interval: 100}
    });
```

Imagen 26: Código de la función "Arrastrar el mapa"

- Impresión de pantalla.

En cada una de las capas, mediante la herramienta "*popup*", se pueden visualizar todas las características que se le añadieron en QGIS como se comentó anteriormente. Por lo que con este comando, se podría imprimir las características de una de las rutas que se vaya a realizar por ejemplo.

Imagen 27: Impresión de pantalla.

```
var print = new OpenLayers.Control.Button({
    title:'Imprimir pantalla', displayClass: "print", trigger: Imp
    });
```

Imagen 28: Código de la función de Impresión de pantalla.

- Etiquetas de Información.

Son una parte muy importante de todo SIG. Nos permite obtener la posición de las rutas, picos, o áreas recreativas así como información sobre cada una de ellas.

Mediante la orden "*popup*", al hacer click sobre cualquiera de los elementos aparecerá una ventana con información acerca de la ruta, pico o área recreativa en cuestión.

Cabe destacar, que en cada una de las rutas se ha insertado una imagen propia de la ruta para que el usuario se haga una idea del aspecto de la ruta.

En la ruta uno, por ejemplo, en vez de insertar una imagen se decidió insertar un vídeo explicativo acerca del concejo de Pravia.

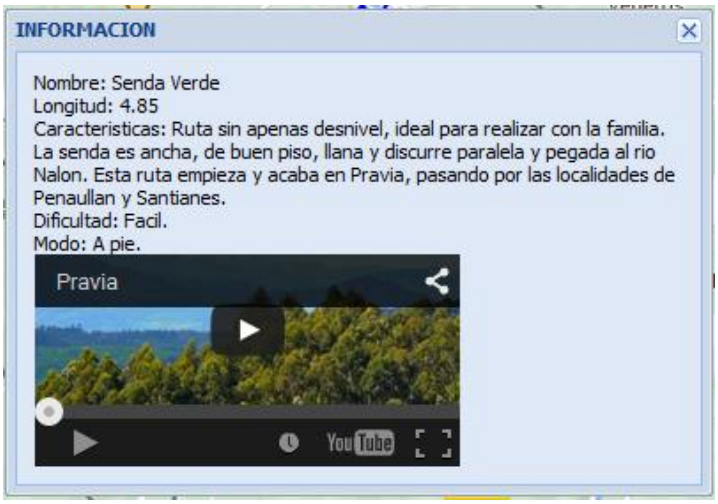


Imagen 29: Ejemplo de ventana de información de la Ruta 1.

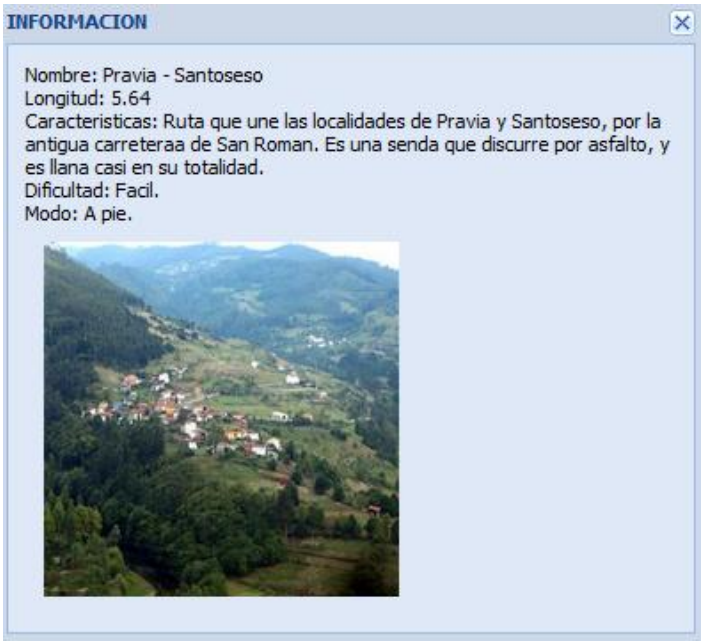


Imagen 30: Ventana de información de la Ruta 2.

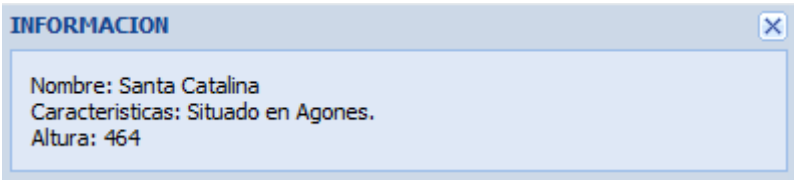


Imagen 31: Ventana de información de uno de los picos.

```
function oninfo7Select(feature) {
    popup7 = new GeoExt.Popup({
    title: 'INFORMACION',
    location: feature,
    feature: feature,
    width:400,
    height:'auto',
    unpinnable:false,
    html:
        "<div style='text-align: left; margin: 10px 10px 10px 10px'>Nombre: " + feature.attributes.Nombre +
        "<div>Longitud: " + feature.attributes.Long_km +
        "<div >Caracteristicas: " + feature.attributes.Caract +
        "<div >Dificultad: " + feature.attributes.Dificultad +
        "<div >Modo: " + feature.attributes.Modo+
        "<div><img src='css/img/7.jpg' title='' height='200' width='200' style='margin:10px;'></div>"
    });
    popup7.on({
        close: function() {
            if(OpenLayers.Util.indexOf(r7.selectedFeatures,
                                       this.feature) > -1) {
                info.unselect(this.feature);
            }
        }
    });
    popup7.show();
}
r7.events.on({
    featureselected: function(e) {
        oninfo7Select(e.feature);
    }
});
```

Imagen 32: Código de la pestaña de información de la Ruta 7.

6. RESULTADOS.

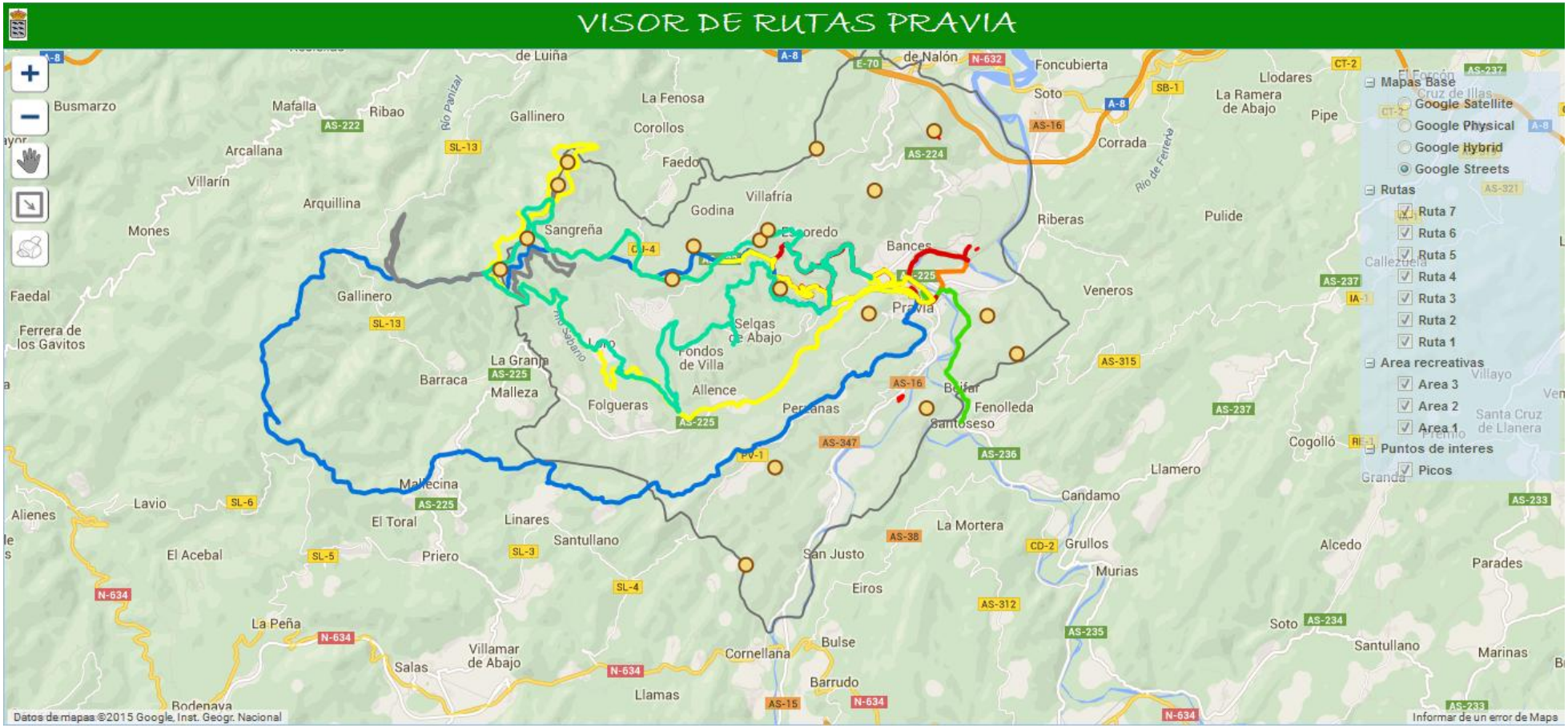


Imagen 33: Resultado final del visor cartográfico.

En la imagen anterior se observa el resultado final del visor creado.

En la parte de la derecha se observa la leyenda, donde todas las capas aparecen divididas por grupos:

- Mapas base. En este apartado se encuentran las cuatro capas cargadas desde Google Maps: Google Satellite, Google Physical, Google Hybrid y Google Streets. Pudiéndose alternar cada una de ellas según lo que se quiera ver.
- Rutas. Aquí se agrupan todas las rutas importadas para este proyecto. Se representan con una línea y cada una de un color distinto a la anterior, para así distinguirlas.
  Se pueden activar o desactivar a gusto del usuario, para solo consultar las que se quieran realizar por ejemplo.
  Si se quiere conocer información sobre la ruta, bastaría con pinchar encima de ella y aparecerá una ventana con información donde vienen el nombre de la ruta, la dificultad, el tiempo estimado en realizarla, características sobre ella, etc.
  En cada ventana de información también se han insertado una imagen para cada ruta, excepto en la *Ruta 1* que aparece un vídeo explicativo sobre el concejo de Pravia.
- Picos. En esta capa aparecen las formaciones montañosas más representativas del concejo. Se representan mediante puntos de color marrón.
  Al igual que las rutas, pinchando encima de cada pico nos dará información acerca de él, como por ejemplo el nombre, la situación, algunas características ,etc.
- Áreas recreativas. En esta capa aparecen las tres áreas recreativas más representativas del concejo. Vienen representadas como polígonos de color rojo, y al igual que las dos capas anteriores posee una ventana de información donde aparece su nombre, su situación y sus características.

En la parte de la izquierda del visor, aparecen las herramientas que se han creado.

- Zoom + y zoom -. Para acercarse o alejarse, según se quieran apreciar más o menos detalles del relieve.
- Arrastrar el mapa. Para poder navegar libremente por el visor.
- Zoom por ventana. Para acercarse a una parte concreta del mapa.
- Impresión de pantalla. Para imprimir información sobre las rutas, picos, áreas recreativas o el recorrido de las rutas, por ejemplo.

En el centro del visor, se observa el concejo de Pravia totalmente transparente para que no interfiera en la visualización del mapa. Esto se hizo para observar cuales de las rutas pasaban también por otros concejos.

7. CONCLUSIONES.

Llegados a este punto se puede decir que se ha cumplido el objetivo del proyecto, que era el de crear un visor cartográfico sobre rutas de senderismo y poder obtener información sobre ellas.

La realización de este Trabajo Fin de Máster en el área de los Sistemas de Información Geográfica, ha sido muy interesante como introducción a estas técnicas de programación, tanto por el aprendizaje que ha supuesto, ya que mi nivel era más bien bajo en estos temas, y por el descubrimiento de la gran cantidad de aplicaciones prácticas que ofrece.

Con respecto a *OpenLayers 3* se podría decir que es una biblioteca de JavaScript muy amplia, y ofrece una gran cantidad de opciones para configurar un SIG.

El visor se desarrolló principalmente en HTML, JavaScript y OpenLayers.

8. BIBLIOGRAFÍA.

- Dodge M., Kitchin R. (2000). *Mapping Cyberspace.*
- INSTITUTO GEOGRÁFICO NACIONAL. *Sistemas de Información Geográfica*. <http://www.ign.es/ign/layoutIn/actividadesSistemaInfoGeografica.do>
- FREE SOFTWARE FOUNDATION. *El sistema operático GNU*. <http://www.gnu.org/philosophy/free-sw.es.html>
- ABC TECNOLOGÍA. (2015). *¿Qué es una API y para que sirve?* <http://www.abc.es/tecnologia/consultorio/20150216/abci--201502132105.html>
- APRENDE A PROGRAMAR. *¿Qué es y para qué sirve HTML?* <http://www.aprenderaprogramar.es/index.php?option=com_content&view=article&id=435:ique-es-y-para-que-sirve-html-el-lenguaje-mas-importante-para-crear-paginas-webs-html-tags-cu00704b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192>
- JUNTA DE ANDALUCÍA. *El lenguaje HTML.* <http://www.juntadeandalucia.es/averroes/~04000134/informatica/html.html>
- MAESTROS DEL WEB. *¿Qué es JavaScript?* <http://www.maestrosdelweb.com/que-es-javascript/>
- APRENDE A PROGRAMAR. *¿Qué es y para qué sirve el lenguaje CSS?* <http://aprenderaprogramar.es/index.php?option=com_content&view=article&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo&catid=46:lenguajes-y-entornos&Itemid=163>
- MÁS ADELANTE. *Definición de CSS. ¿Qué son las hojas de estilo o cascading style sheets?* <https://www.masadelante.com/faqs/css>

9. ANEXOS.

9.1. CÓDIGOS DEL MAPA

9.1.1. CÓDIGO HTML.

```
<html>
	<head>
		<title>VISOR DE RUTAS PRAVIA</title>
		<meta http-equiv="Content-Style-Type" content="text/css">
		<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
		<link rel="stylesheet" type="text/css" href="http://extjs.cachefly.net/ext-3.2.1/resources/css/ext-all.css" />
		<link rel="stylesheet" type="text/css" href="http://extjs.cachefly.net/ext-3.2.1/examples/shared/examples.css" />
		<link href="css/style.css" type="text/css" rel="stylesheet">
		<script type="text/javascript" src="http://www.openlayers.org/api/2.12/OpenLayers.js"></script>
		<script type="text/javascript" src="http://extjs.cachefly.net/ext-3.2.1/adapter/ext/ext-base.js"></script>
		<script type="text/javascript" src="http://extjs.cachefly.net/ext-3.2.1/ext-all.js"></script>
		<script src="http://maps.google.com/maps/api/js?v=3&amp;sensor=false"></script>
		<script type="text/javascript" src="GeoExt/script/GeoExt.js"></script>
		<script type="text/javascript" src="js/pravia.js"></script>
	</head>
<body>
	<div id="header">
		<div class="content">
			<a><img src="css/escudo.jpg" style="float:left; margin:5px 0 0 5px"></a>VISOR DE RUTAS PRAVIA
		</div>
	</div>
  <div id="layers"></div>
</body>
</html>
```

9.1.2. CÓDIGO PRINCIPAL.

```
var
mapPanel,popup1,popup2,popup3,popup4popup5,popup6,popup7,popupa1,popupa2,popupa
3,popup9;

Ext.onReady(function() {

  // if true a google layer is used, if false

        var map = new OpenLayers.Map({
                projection: new OpenLayers.Projection("EPSG:3857"),
                controls: []
        });

        var goo = new OpenLayers.Layer.Google(
                'Google Streets',
                {
                        numZoomLevels: 20
                }
  );
        map.addLayer(goo);

        var gooh = new OpenLayers.Layer.Google(
                'Google Hybrid',
                {
                        type: google.maps.MapTypeId.HYBRID,
                        numZoomLevels: 20
                }
  );
        map.addLayer(gooh);

        var goofis = new OpenLayers.Layer.Google(
                'Google Physical',
                {
                        type: google.maps.MapTypeId.TERRAIN
                }
  );
        map.addLayer(goofis);

        var goosat = new OpenLayers.Layer.Google(
                'Google Satellite',
                {
                        type: google.maps.MapTypeId.SATELLITE,
                        numZoomLevels: 20
                }
  );
        map.addLayer(goosat);

  var extent = new OpenLayers.Bounds(-698127.639105,5375151.925536,-
671123.758199,5396124.282677);
```

```
var estilo_ruta = new OpenLayers.StyleMap({
                                    'default' : new OpenLayers.Style({
                                    fillColor: "#FF8000",
                                    strokeColor: "#FF8000" ,
                                    strokeWidth: 4
                                    })
});

var r1 = new OpenLayers.Layer.Vector(
                                " Ruta 1 ", {
                                    isBaseLayer: false,
                                    strategies: [
                                            new OpenLayers.Strategy.Fixed()
                                            ],
                                            protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/r1.geojson",
        format: new OpenLayers.Format.GeoJSON({
           ignoreExtraDims: true
        })
                                                    }),
                                            styleMap: estilo_ruta
                                    });
                                r1.setVisibility(true);
                                map.addLayer(r1);
var estilo_ruta2 = new OpenLayers.StyleMap({
                                    'default' : new OpenLayers.Style({
                                    fillColor: "#3ADF00",
                                    strokeColor: "#3ADF00" ,
                                    strokeWidth: 4
                                    })
});
var r2 = new OpenLayers.Layer.Vector(
                                " Ruta 2 ", {
                                    isBaseLayer: false,
                                    strategies: [
                                            new OpenLayers.Strategy.Fixed()
                                            ],
                                            protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/r2.geojson",
        format: new OpenLayers.Format.GeoJSON({
           ignoreExtraDims: true
        })
                                                    }),
                                            styleMap: estilo_ruta2
                                    });
                                r2.setVisibility(true);
                                map.addLayer(r2);

var estilo_ruta3 = new OpenLayers.StyleMap({
                                'default' : new OpenLayers.Style({
                                    fillColor: "#0174DF",
                                    strokeColor: "#0174DF" ,
```

```
                                    strokeWidth: 4
                                    })
});

var r3 = new OpenLayers.Layer.Vector(
                                " Ruta 3 ", {
                                    isBaseLayer: false,
                                    strategies: [
                                        new OpenLayers.Strategy.Fixed()
                                        ],
                                        protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/r3.geojson",
        format: new OpenLayers.Format.GeoJSON({
           ignoreExtraDims: true
        })
                                                    }),
                                        styleMap: estilo_ruta3
                                    });
                                r3.setVisibility(true);
                                map.addLayer(r3);
var estilo_ruta4 = new OpenLayers.StyleMap({
                                    'default' : new OpenLayers.Style({
                                        fillColor: "#DF0101",
                                        strokeColor: "#DF0101" ,
                                        strokeWidth: 4
                                        })
                                });

var r4 = new OpenLayers.Layer.Vector(
                                " Ruta 4 ", {
                                    isBaseLayer: false,
                                    strategies: [
                                        new OpenLayers.Strategy.Fixed()
                                        ],
                                        protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/r4.geojson",
        format: new OpenLayers.Format.GeoJSON({
           ignoreExtraDims: true
        })
                                                    }),
                                        styleMap: estilo_ruta4
                                    });
                                r4.setVisibility(true);
                                map.addLayer(r4);

var estilo_ruta5 = new OpenLayers.StyleMap({
                                'default' : new OpenLayers.Style({
                                    fillColor: "#848484",
                                    strokeColor: "#848484" ,
                                    strokeWidth: 4
                                    })
});
```

```
var r5 = new OpenLayers.Layer.Vector(
                        " Ruta 5 ", {
                            isBaseLayer: false,
                            strategies: [
                                new OpenLayers.Strategy.Fixed()
                                ],
                                protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/r5.geojson",
        format: new OpenLayers.Format.GeoJSON({
          ignoreExtraDims: true
        })
                                        }),
                                styleMap: estilo_ruta5
                            });
                        r5.setVisibility(true);
                        map.addLayer(r5);

var estilo_ruta6 = new OpenLayers.StyleMap({
                        'default' : new OpenLayers.Style({
                            fillColor: "#FFFF00",
                            strokeColor: "#FFFF00" ,
                            strokeWidth: 4
                            })
});

var r6 = new OpenLayers.Layer.Vector(
                        " Ruta 6 ", {
                            isBaseLayer: false,
                            strategies: [
                                new OpenLayers.Strategy.Fixed()
                                ],
                                protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/r6.geojson",
        format: new OpenLayers.Format.GeoJSON({
          ignoreExtraDims: true
        })
                                        }),
                                styleMap: estilo_ruta6
                            });
                        r6.setVisibility(true);
                        map.addLayer(r6);

var estilo_ruta7 = new OpenLayers.StyleMap({
                        'default' : new OpenLayers.Style({
                            fillColor: "#01DFA5",
                            strokeColor: "#01DFA5" ,
                            strokeWidth: 4
                            })
});
```

```
var r7 = new OpenLayers.Layer.Vector(
                        " Ruta 7 ", {
                            isBaseLayer: false,
                            strategies: [
                                new OpenLayers.Strategy.Fixed()
                                ],
                                protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/r7.geojson",
        format: new OpenLayers.Format.GeoJSON({
           ignoreExtraDims: true
        })
                                        }),
                                styleMap: estilo_ruta7
                            });
r7.setVisibility(true);
map.addLayer(r7);

var estilo_picos = new OpenLayers.StyleMap({
                        'default' : new OpenLayers.Style({
                            fillColor: "#F5DA81",
                            strokeColor: "#8A4B08" ,
                            pointRadius: 6
                                })
                        });

var picos = new OpenLayers.Layer.Vector(
                        " Picos ", {
                            isBaseLayer: false,
                            strategies: [
                                new OpenLayers.Strategy.Fixed()
                                ],
                                protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/picos.geojson",
        format: new OpenLayers.Format.GeoJSON({
           ignoreExtraDims: true
        })
                                        }),
                                styleMap: estilo_picos
                            });
                        picos.setVisibility(true);
                        map.addLayer(picos);

var estilo_concejo = new OpenLayers.StyleMap({
                        'default' : new OpenLayers.Style({
                            fillColor: "#D8D8D8",
                            strokeColor: "#6E6E6E" ,
                            fillOpacity: 0.1
                            })
});
```

```
var concejo                 = new OpenLayers.Layer.Vector(
                                    "", {
                                    isBaseLayer: false,
                                    strategies: [
                                            new OpenLayers.Strategy.Fixed()
                                            ],
                                            protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/concejo.geojson",
        format: new OpenLayers.Format.GeoJSON({
          ignoreExtraDims: true
        })
                                                    }),
                                            styleMap: estilo_concejo
                                    });
                            concejo.setVisibility(true);
                            map.addLayer(concejo);

var estilo_area = new OpenLayers.StyleMap({
                            'default' : new OpenLayers.Style({
                                    fillColor: "#FF0000",
                                    strokeColor: "#FF0000" ,
                                    pointRadius: 15
                                    })
});

var a1                      = new OpenLayers.Layer.Vector(
                            " Area 1 ", {
                                    isBaseLayer: false,
                                    strategies: [
                                            new OpenLayers.Strategy.Fixed()
                                            ],
                                            protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/a1.geojson",
        format: new OpenLayers.Format.GeoJSON({
          ignoreExtraDims: true
        })
                                                    }),
                                            styleMap: estilo_area
                                    });
                            a1.setVisibility(true);
                            map.addLayer(a1);

var a2                      = new OpenLayers.Layer.Vector(
                            " Area 2 ", {
                                    isBaseLayer: false,
                                    strategies: [
                                            new OpenLayers.Strategy.Fixed()
                                            ],
                                            protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/a2.geojson",
        format: new OpenLayers.Format.GeoJSON({
          ignoreExtraDims: true
```

```
        })
                                                    }),
                                            styleMap: estilo_area
                                    });
                            a2.setVisibility(true);
                            map.addLayer(a2);

var a3                      = new OpenLayers.Layer.Vector(
                            " Area 3 ", {
                                    isBaseLayer: false,
                                    strategies: [
                                            new OpenLayers.Strategy.Fixed()
                                            ],
                                            protocol: new OpenLayers.Protocol.HTTP({
        url: "datos/a3.geojson",
        format: new OpenLayers.Format.GeoJSON({
           ignoreExtraDims: true
        })
                                                    }),
                                            styleMap: estilo_area
                                    });
                            a3.setVisibility(true);
                            map.addLayer(a3);

var info = new OpenLayers.Control.SelectFeature([
                            a1,a2,a3,r1,r2,r3,r4,r5,r6,r7,picos
                            ],{
                                    title: "",
                                    clickout: true
                            });
  map.addControl(info);
  info.activate();

                            function oninfo1Select(feature) {
                                    popup1 = new GeoExt.Popup({
                                    title: 'INFORMACION',
    location: feature,
                                    feature: feature,
                                    width:400,
                                    height:'auto',
                                    unpinnable:false,
    html:
                                            "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                                            "<div>Longitud: " + feature.attributes.Long_km +
                                            "<div >Caracteristicas: " + feature.attributes.Caract +
                                            "<div >Dificultad: " + feature.attributes.Dificultad +
                                            "<div >Modo: " + feature.attributes.Modo+
                                            "<div><iframe width='240' height='120'
src='https://www.youtube.com/embed/B8endrEG6_8' frameborder='0'
allowfullscreen></iframe></div>"
    });
```

```
    popup1.on({
      close: function() {
        if(OpenLayers.Util.indexOf(r1.selectedFeatures,
                   this.feature) > -1) {
          info.unselect(this.feature);
        }
      }
    });
    popup1.show();
  }
  r1.events.on({
    featureselected: function(e) {
      oninfo1Select(e.feature);
    }
  });

                   function oninfo2Select(feature) {
                        popup2 = new GeoExt.Popup({
                        title: 'INFORMACION',
    location: feature,
                        feature: feature,
                        width:400,
                        height:'auto',
                        unpinnable:false,
    html:
                              "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                              "<div>Longitud: " + feature.attributes.Long_km +
                              "<div >Caracteristicas: " + feature.attributes.Caract +
                              "<div >Dificultad: " + feature.attributes.Dificultad +
                              "<div >Modo: " + feature.attributes.Modo+
                              "<div><img src='css/img/2.jpg' title='' height='200'
width='200' style='margin:10px;'></div>"
    });
    popup2.on({
      close: function() {
        if(OpenLayers.Util.indexOf(r2.selectedFeatures,
                   this.feature) > -1) {
          info.unselect(this.feature);
        }
      }
    });
    popup2.show();
  }
  r2.events.on({
    featureselected: function(e) {
      oninfo2Select(e.feature);
    }
  });

                   function oninfo3Select(feature) {
                        popup3 = new GeoExt.Popup({
```

```
                                    title: 'INFORMACION',
    location: feature,
                                    feature: feature,
                                    width:400,
                                    height:'auto',
                                    unpinnable:false,
    html:
                                            "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                                            "<div>Longitud: " + feature.attributes.Long_km +
                                            "<div >Caracteristicas: " + feature.attributes.Caract +
                                            "<div >Dificultad: " + feature.attributes.Dificultad +
                                            "<div >Modo: " + feature.attributes.Modo+
                                            "<div><img src='css/img/3.jpg' title='' height='200'
width='200' style='margin:10px;'></div>"
    });
    popup3.on({
      close: function() {
        if(OpenLayers.Util.indexOf(r3.selectedFeatures,
                        this.feature) > -1) {
          info.unselect(this.feature);
        }
      }
    });
    popup3.show();
  }
  r3.events.on({
    featureselected: function(e) {
      oninfo3Select(e.feature);
    }
  });

                                    function oninfo4Select(feature) {
                                    popup4 = new GeoExt.Popup({
                                    title: 'INFORMACION',
    location: feature,
                                    feature: feature,
                                    width:400,
                                    height:'auto',
                                    unpinnable:false,
    html:
                                            "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                                            "<div>Longitud: " + feature.attributes.Long_km +
                                            "<div >Caracteristicas: " + feature.attributes.Caract +
                                            "<div >Dificultad: " + feature.attributes.Dificultad +
                                            "<div >Modo: " + feature.attributes.Modo+
                                            "<div><img src='css/img/4.jpg' title='' height='200'
width='200' style='margin:10px;'></div>"
    });
    popup4.on({
      close: function() {
```

```
            if(OpenLayers.Util.indexOf(r4.selectedFeatures,
                                this.feature) > -1) {
               info.unselect(this.feature);
            }
         }
      });
      popup4.show();
   }
   r4.events.on({
      featureselected: function(e) {
         oninfo4Select(e.feature);
      }
   });

                                function oninfo5Select(feature) {
                                      popup5 = new GeoExt.Popup({
                                      title: 'INFORMACION',
      location: feature,
                                      feature: feature,
                                      width:400,
                                      height:'auto',
                                      unpinnable:false,
      html:
                                                "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                                                "<div>Longitud: " + feature.attributes.Long_km +
                                                "<div >Caracteristicas: " + feature.attributes.Caract +
                                                "<div >Dificultad: " + feature.attributes.Dificultad +
                                                "<div >Modo: " + feature.attributes.Modo+
                                                "<div><img src='css/img/5.jpg' title='' height='200'
width='200' style='margin:10px;'></div>"
      });
      popup5.on({
         close: function() {
            if(OpenLayers.Util.indexOf(r5.selectedFeatures,
                                this.feature) > -1) {
               info.unselect(this.feature);
            }
         }
      });
      popup5.show();
   }
   r5.events.on({
      featureselected: function(e) {
         oninfo5Select(e.feature);
      }
   });

                                function oninfo6Select(feature) {
                                      popup6 = new GeoExt.Popup({
                                      title: 'INFORMACION',
      location: feature,
```

```
                                    feature: feature,
                                    width:400,
                                    height:'auto',
                                    unpinnable:false,
    html:
                                        "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                                        "<div>Longitud: " + feature.attributes.Long_km +
                                        "<div >Caracteristicas: " + feature.attributes.Caract +
                                        "<div >Dificultad: " + feature.attributes.Dificultad +
                                        "<div >Modo: " + feature.attributes.Modo+
                                        "<div><img src='css/img/6.jpg' title='' height='200'
width='200' style='margin:10px;'></div>"
    });
    popup6.on({
      close: function() {
        if(OpenLayers.Util.indexOf(r6.selectedFeatures,
                    this.feature) > -1) {
          info.unselect(this.feature);
        }
      }
    });
    popup6.show();
  }
  r6.events.on({
    featureselected: function(e) {
      oninfo6Select(e.feature);
    }
  });

                        function oninfo7Select(feature) {
                            popup7 = new GeoExt.Popup({
                            title: 'INFORMACION',
    location: feature,
                            feature: feature,
                            width:400,
                            height:'auto',
                            unpinnable:false,
    html:
                                        "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                                        "<div>Longitud: " + feature.attributes.Long_km +
                                        "<div >Caracteristicas: " + feature.attributes.Caract +
                                        "<div >Dificultad: " + feature.attributes.Dificultad +
                                        "<div >Modo: " + feature.attributes.Modo+
                                        "<div><img src='css/img/7.jpg' title='' height='200'
width='200' style='margin:10px;'></div>"
    });
    popup7.on({
      close: function() {
        if(OpenLayers.Util.indexOf(r7.selectedFeatures,
                    this.feature) > -1) {
```

```
            info.unselect(this.feature);
          }
        }
      });
      popup7.show();
    }
    r7.events.on({
      featureselected: function(e) {
        oninfo7Select(e.feature);
      }
    });

                              function oninfoa1Select(feature) {
                                    popupa1 = new GeoExt.Popup({
                                    title: 'INFORMACION',
      location: feature,
                                    feature: feature,
                                    width:400,
                                    height:'auto',
                                    unpinnable:false,
      html:
                                          "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                                          "<div>Caracteristicas: " + feature.attributes.Caract
      });
      popupa1.on({
        close: function() {
          if(OpenLayers.Util.indexOf(a1.selectedFeatures,
                              this.feature) > -1) {
            info.unselect(this.feature);
          }
        }
      });
      popupa1.show();
    }
    a1.events.on({
      featureselected: function(e) {
        oninfoa1Select(e.feature);
      }
    });

                              function oninfoa2Select(feature) {
                                    popupa2 = new GeoExt.Popup({
                                    title: 'INFORMACION',
      location: feature,
                                    feature: feature,
                                    width:400,
                                    height:'auto',
                                    unpinnable:false,
      html:
                                          "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
```

```
                                        "<div>Caracteristicas: " + feature.attributes.Caract
    });
    popupa2.on({
      close: function() {
        if(OpenLayers.Util.indexOf(a2.selectedFeatures,
                        this.feature) > -1) {
          info.unselect(this.feature);
        }
      }
    });
    popupa2.show();
  }
  a2.events.on({
    featureselected: function(e) {
      oninfoa2Select(e.feature);
    }
  });

                        function oninfoa3Select(feature) {
                            popupa3 = new GeoExt.Popup({
                            title: 'INFORMACION',
    location: feature,
                            feature: feature,
                            width:400,
                            height:'auto',
                            unpinnable:false,
    html:
                                        "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                                        "<div>Caracteristicas: " + feature.attributes.Caract
    });
    popupa3.on({
      close: function() {
        if(OpenLayers.Util.indexOf(a3.selectedFeatures,
                        this.feature) > -1) {
          info.unselect(this.feature);
        }
      }
    });
    popupa3.show();
  }
  a3.events.on({
    featureselected: function(e) {
      oninfoa3Select(e.feature);
    }
  });

                        function oninfop1Select(feature) {
                            popup9 = new GeoExt.Popup({
                            title: 'INFORMACION',
    location: feature,
                            feature: feature,
```

```
                                    width:400,
                                    height:'auto',
                                    unpinnable:false,
    html:
                                        "<div style='text-align: left; margin: 10px 10px 10px
10px'>Nombre: " + feature.attributes.Nombre +
                                        "<div>Caracteristicas: " + feature.attributes.Caract +
                                        "<div>Altura: " + feature.attributes.Altura_m
    });
    popup9.on({
      close: function() {
        if(OpenLayers.Util.indexOf(picos.selectedFeatures,
                        this.feature) > -1) {
          info.unselect(this.feature);
        }
      }
    });
    popup9.show();
  }
  picos.events.on({
    featureselected: function(e) {
      oninfop1Select(e.feature);
    }
  });

                            var pan =  new OpenLayers.Control.Navigation({
                                title: 'Arrastrar el mapa',
                                mouseWheelOptions: {interval: 100}
                                });

                            var zoomVentana = new OpenLayers.Control.ZoomBox({
                                title: 'Hacer Zoom ventana',
                                zoomOnClick: false,
                                displayClass:'ZoomBox'
                                });

                            var zin = new OpenLayers.Control.ZoomIn ({
                                title: 'Zoom M\u00e1s'
                                });

                            var zou = new OpenLayers.Control.ZoomOut ({
                                title: 'Zoom Menos'
                                });

                            function Imp() {
                                window.print ()
                                }
                            var print = new OpenLayers.Control.Button({
                                title:'Imprimir pantalla', displayClass: "print", trigger: Imp
                                });

                            var barraHerramientas = new OpenLayers.Control.Panel();
```

```
                        barraHerramientas.addControls([
                                zin,
                                zou,
                                pan,
                                zoomVentana,
                                print
                        ]);
                        map.addControl(barraHerramientas);

  new Ext.Viewport({
    layout: "border",
    items: [{
      region: "north",
      contentEl: "header",
      height: 40,
                                        border: false,
                                        bodyStyle: {"background-color": "#088A08"}
    }, {
      region: "center",
      id: "mappanel",
      title: "",
      xtype: "gx_mappanel",
      map: map,
      extent: extent,
      split: true,
                                        bodyStyle: {"z-index": "1"}
    }]
  });
  mapPanel = Ext.getCmp("mappanel");

                        var layerListgoo = new GeoExt.tree.BaseLayerContainer({
                                text: 'Mapas Base',
    layerStore: mapPanel.layers,
    expanded: true,
    autoLoad: true
                        });

                        var layerTreegoo = new Ext.tree.TreePanel({
                                title: '',
                                renderTo: "layers",
                                root: layerListgoo,
                                border: false,
                                bodyStyle:  {"background-color":  "#CEE3F6","opacity":".6","z-
index": "2"}
                        });

                        var layerList = new GeoExt.tree.LayerContainer({
                                text: 'Rutas',
                                layerStore: mapPanel.layers,
                                leaf: false,
                                expanded: true,
                                loader: {
```

```
                                        filter: function(record) {
                                                var myarr = new Array();
                                                myarr[0] = record.get("layer").name.indexOf("
Ruta 1");
                                                myarr[1] = record.get("layer").name.indexOf("
Ruta 2 ");
                                                myarr[2] = record.get("layer").name.indexOf("
Ruta 3 ");
                                                myarr[3] = record.get("layer").name.indexOf("
Ruta 4 ");
                                                myarr[4] = record.get("layer").name.indexOf("
Ruta 5 ");
                                                myarr[5] = record.get("layer").name.indexOf("
Ruta 6 ");
                                                myarr[6] = record.get("layer").name.indexOf("
Ruta 7 ");
                                                if(myarr[0]==-1 && myarr[1]==-1 &&
myarr[2]==-1 && myarr[3]==-1 && myarr[4]==-1 && myarr[5]==-1 && myarr[6]==-1)
                                                {
                                                        return false;
                                                }
                                        else
                                                {
                                                        return true;
                                                }
                                        }
                                }
                        });

                        var layerTree = new Ext.tree.TreePanel({
                                title: '',
                                renderTo: "layers",
                                root: layerList,
                                border: false,
                                bodyStyle:  {"background-color":  "#CEE3F6","opacity":".6","z-
index": "2"}
                        });

                        var layerList2 = new GeoExt.tree.LayerContainer({
                                text: 'Area recreativas',
                                layerStore: mapPanel.layers,
                                leaf: false,
                                expanded: true,
                                loader: {
                                        filter: function(record) {
                                                var myarr = new Array();
                                                myarr[0] = record.get("layer").name.indexOf("
Area 1 ");
                                                myarr[1] = record.get("layer").name.indexOf("
Area 2 ");
                                                myarr[2] = record.get("layer").name.indexOf("
Area 3 ");
```

```
                                        if(myarr[0]==-1 && myarr[1]==-1 && myarr[2]==-1)
                                        {
                                                return false;
                                        }
                                else
                                        {
                                                return true;
                                        }
                                }
                        }
                });

                var layerTree2 = new Ext.tree.TreePanel({
                        title: '',
                        renderTo: "layers",
                        root: layerList2,
                        border: false,
                        bodyStyle:  {"background-color":  "#CEE3F6","opacity":".6","z-index": "2"}
                });

                var layerList3 = new GeoExt.tree.LayerContainer({
                        text: 'Puntos de interes',
                        layerStore: mapPanel.layers,
                        leaf: false,
                        expanded: true,
                        loader: {
                                filter: function(record) {
                                        var myarr = new Array();
                                        myarr[0] = record.get("layer").name.indexOf("Picos ");
                                        if(myarr[0]==-1)
                                        {
                                                return false;
                                        }
                                else
                                        {
                                                return true;
                                        }
                                }
                        }
                });

                var layerTree3 = new Ext.tree.TreePanel({
                        title: '',
                        renderTo: "layers",
                        root: layerList3,
                        border: false,
                        bodyStyle:  {"background-color":  "#CEE3F6","opacity":".6","z-index": "2"}
```

```
});

});
```