

Universidad de Alicante

# Sindicación de contenidos en mapas

Dpto. de Física, Ingeniería de Sistemas y Teoría de la Señal

Rubén Jarque Torrejón  
José Manuel Pérez Pérez

## Tabla de contenido

Introducción .....	4
Estado actual de las tecnologías web.....	4
Comparando mashups con portales .....	5
Mashups de vídeos, fotos y demás .....	5
Motivación.....	6
Desarrollo .....	7
Herramientas empleadas .....	7
TortoiseCVS .....	7
FileZilla.....	7
Navegadores Web .....	7
Extensión Firebug.....	7
Eclipse.....	8
Tomcat.....	8
EasyPHP .....	8
PHP .....	8
MySQL .....	9
Python .....	9
Notepad++.....	9
MediaWiki .....	9
Arquitectura física del contenido del sitio web.....	10
Mapa Web .....	13
Funcionamiento de la interfaz del sitio web .....	15
Capa de Wikipedia.....	15
Capa de Youtube .....	17
Capa de Commons.....	19
Capa de Noticias .....	22
Capa de Meteorología .....	24
Capa de Anuncios .....	26
Capa de Usuarios.....	27
Extracción de contenido de MediaWiki .....	29
Funcionamiento de los scripts python .....	29
Tandas de extracción.....	31

Recopilación de imágenes de Commons.....	32
Importación de SQL.....	34
Eliminación de paréntesis .....	35
Corrector de coordenadas Wikipedia .....	36
Clasificador de puntos .....	37
Gestión de feeds.....	45
Proveedores de puntos .....	50
Categorizador de textos .....	59
Componentes empleados .....	71
Geoposicionamiento por IP .....	73
Planificación de actividades .....	74
Anexo I: Diagramas de la base de datos.....	76
Anexo II: Diagramas de clases de la aplicación Java.....	79

## Introducción

### Estado actual de las tecnologías web

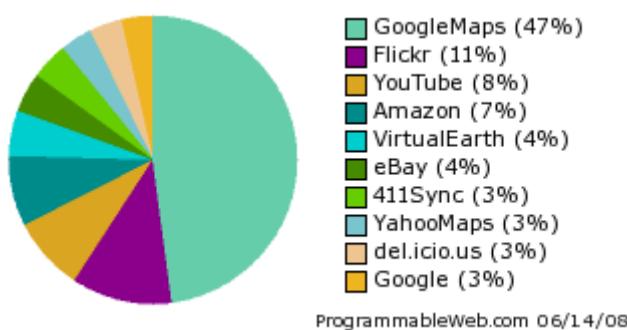
Durante los últimos años se ha puesto de manifiesto un cambio en lo referente a la participación de los usuarios en los sitios web. Mientras al principio el usuario se limitaba a observar el contenido de los mismos, que solía ser introducido por el mismo administrador de la página, ahora los visitantes son parte activa del desarrollo. Esto permite un crecimiento más rápido dado que, de alguna forma, el aumento del contenido es proporcionado por los usuarios. Además, ellos mismos pueden ser capaces de validar las contribuciones del resto de usuarios, convirtiendo los sitios web en comunidades colaborativas. Por otra parte, los mismos usuarios se sienten más partícipes del sitio y mejora su percepción y fidelidad.

Algunos ejemplos de este tipo de sitios son:

- Wikipedia (otros wikis)
- Foros
- Flickr
- Youtube
- Wikia Search

La extensión de las redes sociales y la apertura de los servicios ofrecidos por sitios como Google Maps han propiciado que haya surgido una multitud de sitios web denominados "mashup", que es un sitio web o aplicación web que usa contenido de otras aplicaciones Web para crear un nuevo contenido completo, consumiendo servicios directamente siempre a través de protocolo http. El contenido de un mashup normalmente proviene de sitios web de terceros a través de una interfaz pública o usando un API. Otros métodos que constituyen el origen de sus datos incluyen: sindicadores web (RSS o Atom), Screen scraping, etc.

A fecha de 14 de junio de 2008, el sitio web <http://www.programmableweb.com/mashups>, un referente en directorio de mashups, recogía 3.124 de estos sitios, con un crecimiento de más de 3 nuevos diarios.



El API más usada por los mashups es, con diferencia, la de GoogleMaps. Esta API permite utilizar un mapa de Google y ofrece una serie de funciones para trabajar con los mapas (zoom, desplazamientos, tipos de mapa, buscador de lugares) así como marcadores que abren ventanas de información. A continuación está la de Flickr, que permite obtener las fotos subidas por los usuarios en dicho sitio, buscándolas según autor, tags, descripción o fecha.

Así, cada API pretende que los usuarios puedan hacer uso del contenido de los sitios, sin necesidad de entrar, en este caso, a los sitios web de GoogleMaps o Flickr. Este modelo de negocio, que al principio

puede parecer poco viable al no recibir visitas directamente a dicho sitio web, proporciona en cambio un *feedback* y una mayor presencia en Internet de los sitios que, en lugar de ser meros portales, pasan a ofrecer las funcionalidades a otras webs.

## Comparando mashups con portales

Los mashups y los portales son sitios web basados en la agregación de contenidos. Los portales es una tecnología más antigua diseñada como una extensión a las aplicaciones web dinámicas tradicionales en las que el proceso de convertir contenido en páginas web está dividido en 2 fases: generación de "fragmentos" de contenido y agregación de los fragmentos en páginas. Cada uno de los fragmentos es generado por un portlet y el portal los combina en una única página web. Los portlets pueden ser hospedados de forma local en el servidor del portal o remotamente en otro servidor.

La tecnología de portal define un modelo de evento completo que cubre lecturas y actualizaciones. Una petición hacia una página agregada en un portal es traducida a operaciones de lectura individuales a todos los portlets que forman la página (operaciones de "renderizado" en local, portlets JSR 168 u operaciones "getMarkup" en remoto, portlets WSRP). Si se presiona un botón de Enviar en cualquier portlet de un portal, es traducido a una única operación de actualización sólo en ese portlet ("processAction" en local, portlet JSR 168 o "performBlockingInteraction" en remoto, portlet WSRP). La actualización es seguida inmediatamente por una lectura en *todos* los portlets de la página.

## Mashups de vídeos, fotos y demás

Aunque hay muchos mashups, pocos de ellos proporcionan más de un tipo de contenido. Centrándonos en contenido sobre mapas (en lo que se basa nuestro proyecto) podemos destacar:

### Fotos

- <http://loc.alize.us/>
- <http://www.flickr.com/map/>
- <http://www.panoramio.com>
- <http://www.google.com/maps>

### Mashups de vídeos

- <http://www.virtualvideomap.com/>

### Noticias

- <http://meneame.net/map.php>

### Aplicaciones de escritorio

- Google Earth

Aunque hay miles de mashups, en realidad se suelen centrar en un sólo tipo de contenido (salvo en el caso de Google Earth con su sistema de capas). Así, para obtener información completa sobre una región determinada es necesario visitar cada uno de estos sitios. Google Maps está comenzando a incluir fotos de Panoramio y, desde un par de meses atrás, algunos artículos de Wikipedia. De hecho, se podría decir que

hasta la aparición de nuestro proyecto no había ningún sitio que contemplara el posicionamiento de los artículos de Wikipedia ni las fotos del repositorio que utiliza, Wikimedia Commons.

## Motivación

Este proyecto surge a partir de una proposición por parte del Departamento de Física y Teoría de la Señal de la Universidad de Alicante.

Se trataba de realizar una aplicación avanzada de sindicación de contenidos multiformato en ubicaciones específicas de mapas satelitales, pretendiendo importar a mapas 2D diverso material multimedia de otras webs y redes sociales de Internet. Este material provendría de Webs que permitieran la sindicación de sus contenidos en otras plataformas y/o que presentaran una alta contribución social por parte de los internautas como, por ejemplo, enciclopedias libres (Wikipedia), comunidades de vídeo (YouTube), sistemas de blogs (Blogger, Wordpress), anuncios clasificados (Loquo, Planetanuncios, Infocampus), servidores fotográficos (Flickr), medios de comunicación, etc.

Estudiando las diversas fuentes que podíamos integrar en el proyecto nos pareció interesante introducir contenido que no estuviera posicionado en otros sitios web. Tal es el caso de Wikipedia, para la cual tuvimos que implementar unos scripts que recorriera la enciclopedia y extrajeran información sobre su posición, o las noticias, para las que desarrollamos unas funciones para extraer información sobre localización, basándonos en la información obtenida en Wikipedia. Además, utilizamos esta misma técnica para posicionar vídeos de Youtube y anuncios clasificados.

Así, nuestro proyecto supone un avance en la integración de contenido que no tiene por qué estar posicionado en el momento de la incorporación en sus respectivos sitios, sino que se adapta a la información existente y es capaz de posicionar contenido basándose en texto, ampliando la cantidad de contenido susceptible de ser posicionada.

## Desarrollo

### Herramientas empleadas

#### TortoiseCVS

TortoiseCVS<sup>1</sup> es una herramienta CVS para Microsoft Windows publicada bajo la GNU General Public License. Al contrario que la mayoría de las herramientas CVS, se incluye en el shell propio de Windows añadiendo entradas en el menú contextual del explorador de ficheros, por lo tanto no se ejecuta en su propia ventana. Más aún, esto añade iconos sobre los ficheros y directorios controlados por CVS, dando información adicional al usuario sin tener que ejecutar una aplicación individual.

Aunque en un principio utilizamos como herramienta de control de versiones la incluida con el IDE Eclipse, conforme fuimos desarrollando parte de proyecto en PHP fuimos utilizando este CVS.

El uso de un CVS nos ha permitido tener un control sobre las diversas modificaciones que cada uno de nosotros llevaba a cabo, así como la sincronización del contenido que, de otra forma, habría sido muy difícil de gestionar. Como espacio para el CVS hemos empleado el proporcionado por Assembla<sup>2</sup>, que es un sitio web que incluye seguimiento de tareas, wiki y visualización de los cambios realizados en cada archivo en forma de diff. Nosotros hemos usado esta última funcionalidad, que complementábamos con el envío de correos electrónicos de forma automática al realizar cada una de las modificaciones.

#### FileZilla

FileZilla<sup>3</sup> es un cliente FTP, gratuito, libre (GNU) y de código abierto. Soporta FTP, SFTP y FTP sobre SSL. Inicialmente sólo diseñado para funcionar bajo Windows, desde la versión 3.0.0, gracias al uso de wxWidgets, es multiplataforma, estando disponible además para otros sistemas operativos, entre ellos Linux, FreeBSD y MacOS X. De hecho, lo hemos empleado tanto en Windows como en Ubuntu Linux, con muy buenos resultados.

El cliente FTP ha sido empleado para realizar la gestión de los archivos HTML, CSS, PHP, Javascript e imágenes. Optamos por esta solución porque la que está integrada en Plesk (administrador de ficheros) es poco cómoda, en especial para trabajar con varios archivos.

#### Navegadores Web

Para el desarrollo del proyecto hemos empleado durante todo el ciclo varios navegadores para observar la compatibilidad de las distintas funcionalidades en cada uno de ellos. Cada uno de ellos tiene un motor de renderizado diferente, por lo que probando con ellos equivalía a probar con otros muchos navegadores que también emplean sus motores de renderizado. En concreto, Mozilla Firefox (que usa Gecko), Internet Explorer (que usa Trident) y Safari (que usa WebKit).

#### Extensión Firebug

Firebug es una extensión (add-on) de Firefox creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga,

<sup>1</sup> El sitio web de Tortoise CVS es <http://www.tortoisecvs.org>

<sup>2</sup> Disponible en <http://www.assembla.com>

<sup>3</sup> Filezilla se puede descargar en <http://filezilla-project.org>

estructura DOM), editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea e “inline”.

Firebug no es un simple inspector como DOM Inspector, además edita y permite guardar los cambios, un paso por delante del conocido Web Developer. Su atractiva e intuitiva interfaz, con solapas específicas para el análisis de cada tipo de elemento (consola, HTML, CSS, Script, DOM y red), permite al usuario un manejo fácil y rápido.

Gracias a su uso hemos podido depurar mucho mejor las distintas llamadas asíncronas y la maquetación CSS.

### Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus.

Decidimos emplear Eclipse desde el primer momento para realizar gran parte del proyecto, en especial lo concerniente a la actualización y gestión de feeds RSS. La exportación del proyecto la hemos ido realizando a archivos WAR que eran alojados en el servidor a través del panel de control Plesk.

### Tomcat

Tomcat es un servidor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache. Es el servidor que hemos utilizado tanto en local como luego en el servidor.

### EasyPHP

EasyPHP proporciona un servidor Apache, base de datos MySQL y PHP, ya configurados y preparados para el desarrollo. Nos ha ayudado inmensamente para realizar la aplicación en local y, dado que en el servidor se utilizan las mismas herramientas, no hemos tenido apenas problemas para la adaptación del trabajo desarrollado en nuestros equipos (salvo algunos problemas de codificación de caracteres).

### PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Nos ha sido muy útil para incorporar el contenido dinámico al sitio web y realizar modificaciones rápidamente. Aunque en un principio el frontal estaba siendo realizado utilizando

JSP, al final optamos por PHP porque la constante exportación y actualización del archivo WAR nos llevaba bastante tiempo.

### MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Para el tamaño de nuestra base de datos es muy adecuada, además de ser gratuita y estar ampliamente disponible en los servidores de alojamiento.

### Python

Python es un lenguaje de programación interpretado que hemos utilizado para poder recorrer los distintos proyectos basados en Mediawiki (Wikipedia y Wikimedia Commons) y extraer información de los artículos, como el texto, coordenadas y, en el caso de las poblaciones, número de habitantes.

### Notepad++

Notepad++ es un editor de código fuente libre, que admite varios lenguajes de programación y se ejecuta en Microsoft Windows. Este proyecto, basado en el componente de edición Scintilla, está escrito en C++ utilizando directamente la API de Win32 y STL, lo que asegura una velocidad mayor de ejecución y un tamaño más reducido del programa final. Se distribuye bajo los términos de la Licencia Pública General de GNU.

Los lenguajes de programación admitidos son: C, C++, Java, C#, XML, HTML, PHP, JavaScript, archivos de recursos RC, makefile, Arte ASCII, doxygen, archivos INI, archivos por lotes (BAT), ASP, archivos VB/VBS, Shell script de UNIX, SQL, Objective-C, CSS, Pascal, Perl, Python, Lua, TeX, TCL, lenguaje ensamblador, Ruby, Lisp, Scheme, Smalltalk, PostScript, VHDL, FORTRAN, Ada, Caml, Autolt, KiXtart, Matlab y Verilog. Además, los usuarios pueden definir su propio lenguaje usando User Language Define System incorporado, el cual hace al Notepad++ extensible, para tener resaltado de sintaxis y plegamiento de sintaxis.

Nos ha resultado muy útil por el resaltado de sintaxis y, sobre todo, por poder trabajar con distintas codificaciones de caracteres y poder procesar archivos de texto relativamente grandes mucho más eficientemente que el básico bloc de notas.

### MediaWiki

MediaWiki es un motor para wikis bajo licencia GNU, programado en PHP. A pesar de haber sido creado y desarrollado para Wikipedia y los otros proyectos de la fundación Wikimedia, ha tenido una gran expansión a partir de 2005, existiendo gran número de wikis basados en este software que nada tienen que ver con dicha fundación. La mayoría de ellos se dedican a la documentación de software o a temas especializados.

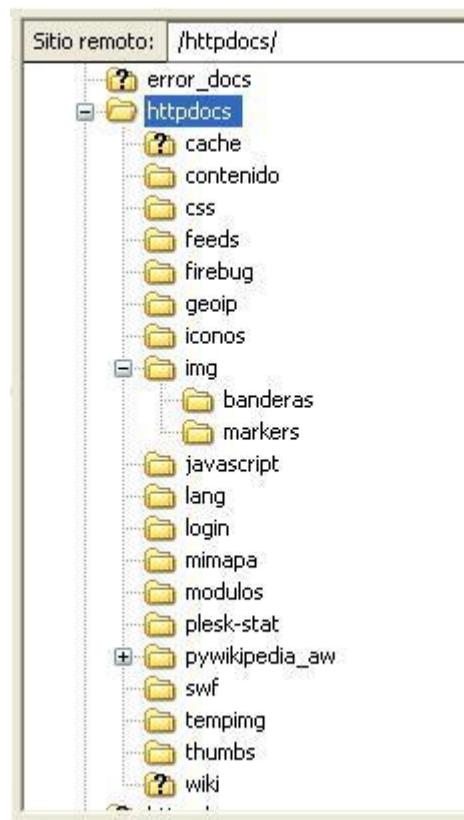
La experiencia que teníamos al usar el proyecto de Wikipedia y las posibilidades que le vimos desde el primer momento han hecho que investigáramos en el uso de este software para encontrar la forma de recuperar el contenido de los artículos. Además, también lo hemos empleado para realizar la documentación del proyecto.

## Arquitectura física del contenido del sitio web

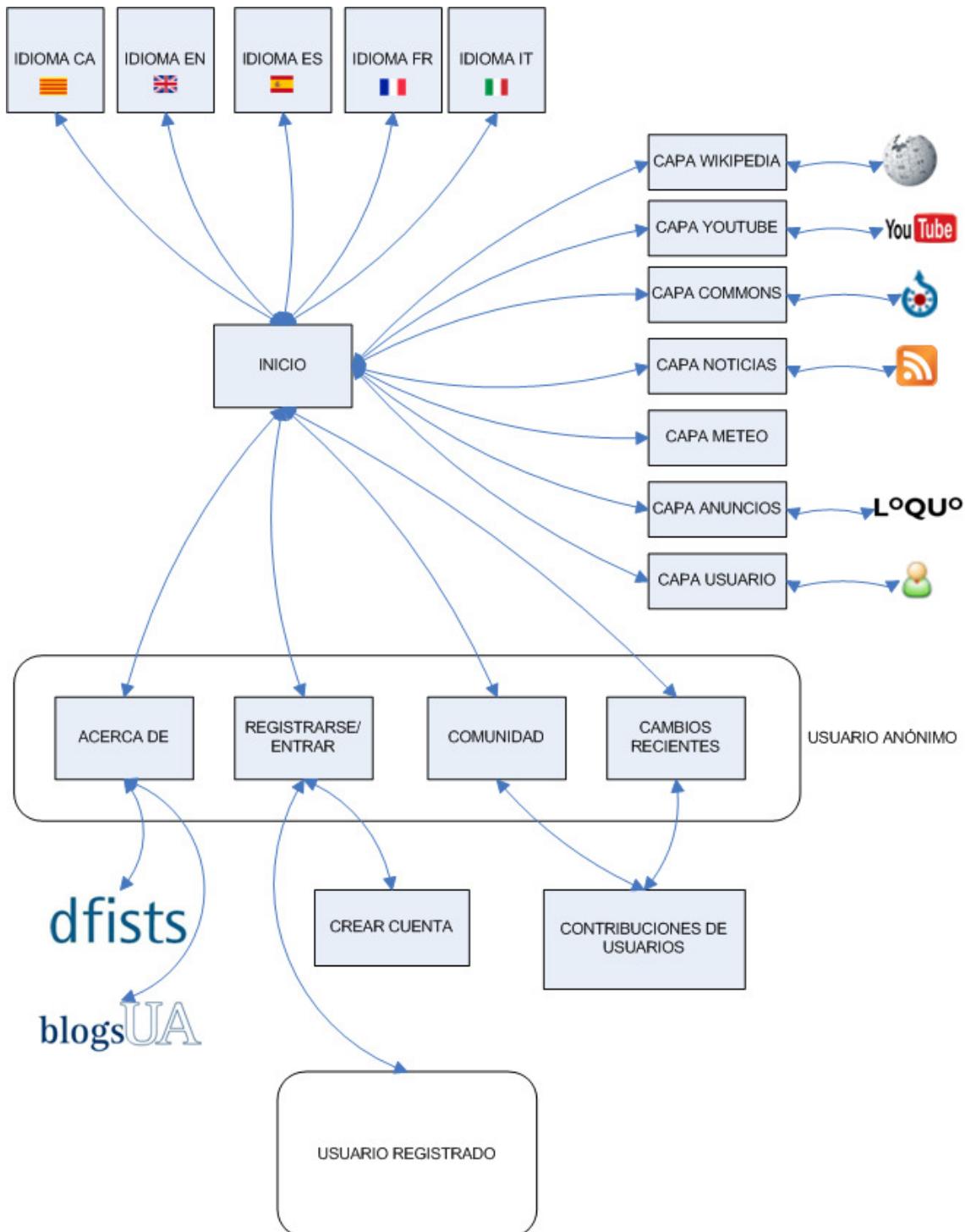
Para facilitar la gestión de los archivos componentes del sistema se ha seguido una estructura organizada de los mismos. A continuación se describe la arquitectura física de los ficheros almacenados en el servidor.

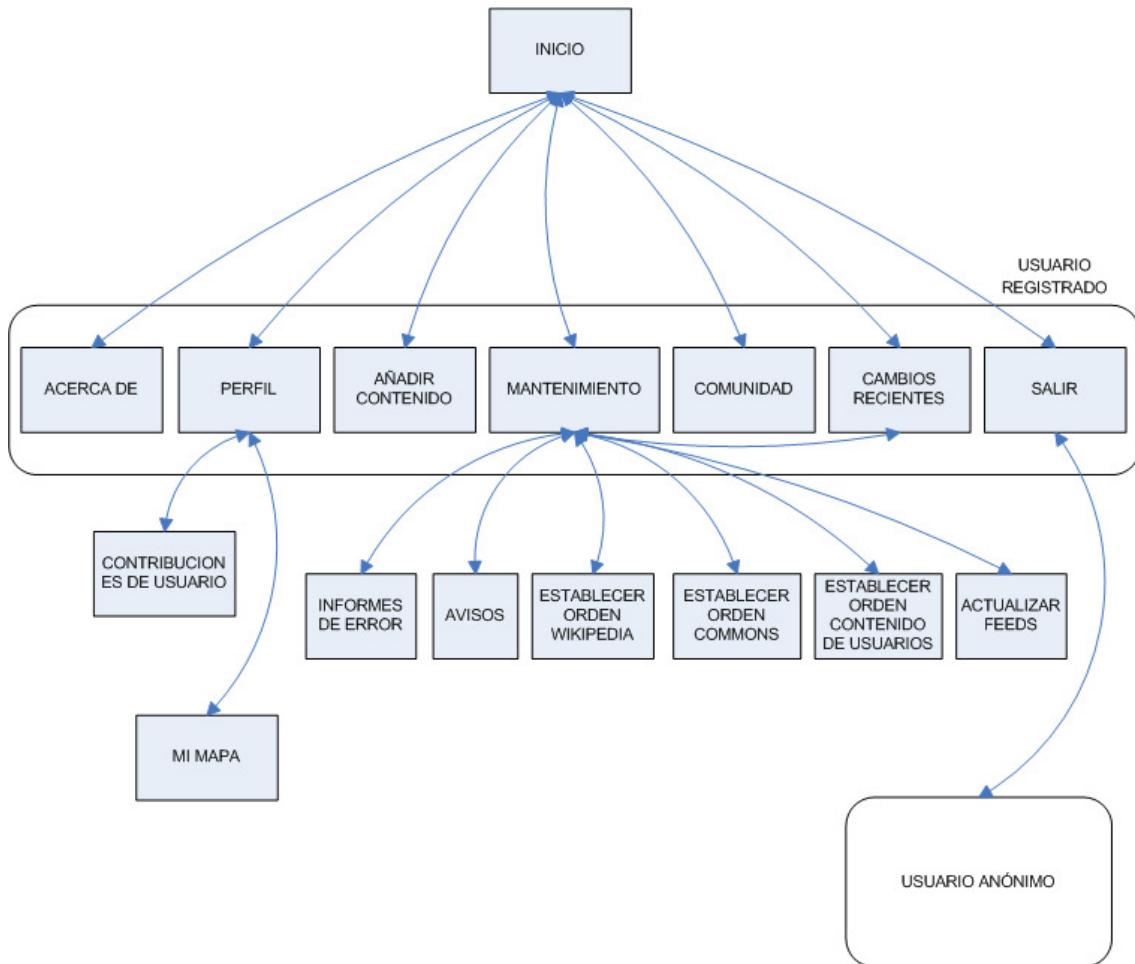
- / Directorio raíz (htdocs en el servidor)
  - **acercade.php** Página donde se describe el proyecto, su propósito y sus autores.
  - **actualizarNumAvisos.php** Establece el número de avisos necesarios para retirar preventivamente un punto publicado por los usuarios.
  - **actualizarWikipedia.php** Página que realiza la llamada al cgi de actualización de un artículo de Wikipedia.
  - **arregladoArticulo.php** Marca un artículo de Wikipedia como arreglado, después de haber recibido avisos de información errónea en su contenido.
  - **cambiarPerfil.php** Formulario para que el usuario cambie su información de perfil.
  - **cambiosRecientes.php** Página de los cambios recientes.
  - **comunidad.php** Página de los usuarios registrados en el sistema.
  - **conexion.php** Abre una conexión con la base de datos.
  - **cookies.php** Establece las variables necesarias tras la lectura de las cookies encontradas en el usuario.
  - **creaMinatura.php** Procesa la petición de un usuario de aportar un contenido, añadiendo el contenido a la base de datos.
  - **email.php** Fichero con clases utilizadas en el envío de correos electrónicos informativos a los usuarios.
  - **enviarAviso.php** Recoge la petición del usuario de enviar un aviso de contenido inapropiado.
  - **error.php** Página que muestra un mensaje de acción errónea al usuario.
  - **funciones\_bd.php** Funciones que abstraen las llamadas a la base de datos.
  - **geopais.php** Comprueba si el país de visita del usuario tiene las coordenadas establecida en la base de datos.
  - **iframe.php** Página que muestra al usuario su mapa de contribuciones y el código necesario para exportar el mapa a otra página web.
  - **index.php** Página principal de inicio.
  - **informarError.php** Recoge la solicitud del usuario de enviar un informe de error de contenido en los artículos de Wikipedia.
  - **mantenimiento.php** Página de mantenimiento del sistema para los administradores.
  - **marcarInapropiado.php** Marca como inapropiado un punto subido por los usuarios.
  - **mensaje.php** Muestra información al usuario de una acción llevada a cabo correctamente.
  - **menu.php** Código de la cabecera usada en todas las páginas.
  - **nuevoUsuario.php** Da de alta una nueva cuenta de usuario.
  - **num\_avisos.php** Lee de la base de datos el número de avisos necesarios para retirar preventivamente contenido inapropiado.
  - **perfil.php** Página que muestra los datos del perfil de usuario.
  - **registrarVisita.php** Registra una visita a un marcador en la base de datos.
  - **status.php** Obtiene información del sistema, como el número de artículos, imágenes, etc.
  - **subida.php** Página para la subida de contenidos para usuarios.
  - **thumbnail.php** Crea una miniatura cuadrada de una imagen dada.
  - **verAvisos.php** Página que muestra los avisos de contenido inapropiado registrados en el sistema.

- **verContribuciones.php** Muestra un resumen de las contribuciones realizadas por un usuario.
- **verInformes.php** Página que muestra los informes de error registrados en el sistema.
- **contenido/** Directorio con las miniaturas de las imágenes subidas por los usuarios.
- **css/** Directorio con las hojas de estilo usadas.
- **feeds/** Directorio con los proveedores de puntos php que devuelven marcadores en formato JSON a la página principal:
  - **conexion.php** Abre una conexión con la base de datos en este nivel del árbol de directorios.
  - **contenidoPropio.php** Devuelve los marcadores en la región solicitada de contenido de usuario.
  - **cookies.php** Establece las variables leídas de las cookies en este nivel del árbol de directorios.
  - **noticias.php** Devuelve las noticias correspondiente a la región actual.
  - **wiki.php** Devuelve los marcadores de artículos de Wikipedia.
  - **youtube.php** Devuelve los videos encontrados para la región actual.
- **geoip/** Directorio con los ficheros necesarios para obtener la IP y el lugar de origen de las visitas.
- **iconos/** Iconos usados en los marcadores meteorológicos y en la clasificación de las noticias.
- **img/** Resto de imágenes e iconos usadas en el proyecto.
  - **banderas/** Iconos de banderas usados para representar los idiomas disponibles.
  - **markers/** Iconos de marcadores.
- **javascript/** Directorio con las funciones javascript empleadas para la gestión del mapa (destaca **map\_functions.js**)
- **lang** Directorio con las traducciones de los mensajes del sistema en los distintos idiomas, tanto los mensajes php como los mensajes javascript.
- **login/** Páginas de gestión de las cuentas de usuario:
  - **conexion.php** Abre una conexión a la base de datos para la gestión de usuarios.
  - **cookies.php** Establece las variables de usuario leídas en las cookies del usuario.
  - **identificar.php** Comprueba si el usuario tiene una cookie con el nombre de usuario y contraseña correctos.
  - **ingresar.php** Lee el nombre de usuario y la contraseña introducidos por el usuario y procede a su comprobación.
  - **login.php** Página que muestra el formulario de identificación al usuario.
  - **logout.php** Cierra la sesión del usuario.
  - **menuLogin.php** Cabecera común en las páginas de gestión de cuentas del usuario.
  - **nuevaClave.php** Genera una nueva contraseña para el usuario y se la envía por correo.
  - **nuevoUsuario.php** Crea una nueva cuenta de usuario.
  - **registro.php** Formulario de datos para crear una nueva cuenta de usuario.
- **mimap/** Directorio con los ficheros de las páginas necesarias para la visualización del mapa personalizado de usuario exportable.
- **modulos/** Directorio con páginas de implementación de módulos adicionales para el proyecto.
- **thumbs/** Directorio donde se almacenan las miniaturas de las imágenes extraídas de Wikimedia Commons.
- **cgi-bin/** Directorio que almacena los cgi en python que extraen información de Wikipedia y de Commons.



## Mapa Web





## Funcionamiento de la interfaz del sitio web

### Capa de Wikipedia

La capa de Wikipedia es la que se abre por defecto al usuario nuevo cuando entra en el sitio. Esta capa contiene una introducción de los artículos de la Wikipedia en el idioma seleccionado geoposicionados en el mapa actual del usuario. El aspecto de la pantalla será un mapa con una serie de puntos azules y violetas, con una W. Al seleccionar uno de los puntos tendremos el siguiente aspecto:



En la parte derecha tenemos una pequeña descripción de cada uno de los puntos en el mapa. Al pinchar sobre Punto categorizado cualquiera de ellos, abrirá su correspondiente globo. Ciertos puntos pueden ser categorizados a partir de su título. Así artículos de aeropuertos, museos o universidades, por ejemplo, aparecerán con un fondo personalizado a su contenido. En la captura de la derecha se puede ver un ejemplo del fondo resultado en el punto correspondiente a la Universidad de Alicante.

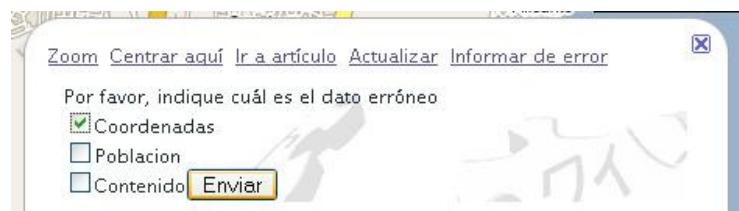
Dentro del globo tenemos la introducción completa. Esta introducción es el comienzo, en texto plano, del artículo de Wikipedia. En la parte superior del globo tenemos un enlace directo a la página del artículo correspondiente en Wikipedia, un enlace para hacer zoom hasta ese punto, y otro para centrar el mapa en ese punto. La herramienta de centrar el mapa puede ser útil para consultar la información meteorológica cercana a ese punto, pues en la parte superior izquierda, debajo de los enlaces de los idiomas, se muestra la información del tiempo en el punto central del mapa, o el de la localidad más cercana a ese punto.

Los marcadores azules, que son un poco más grandes que los violetas, corresponden a localidades de las que se ha extraído su población automáticamente de Wikipedia.



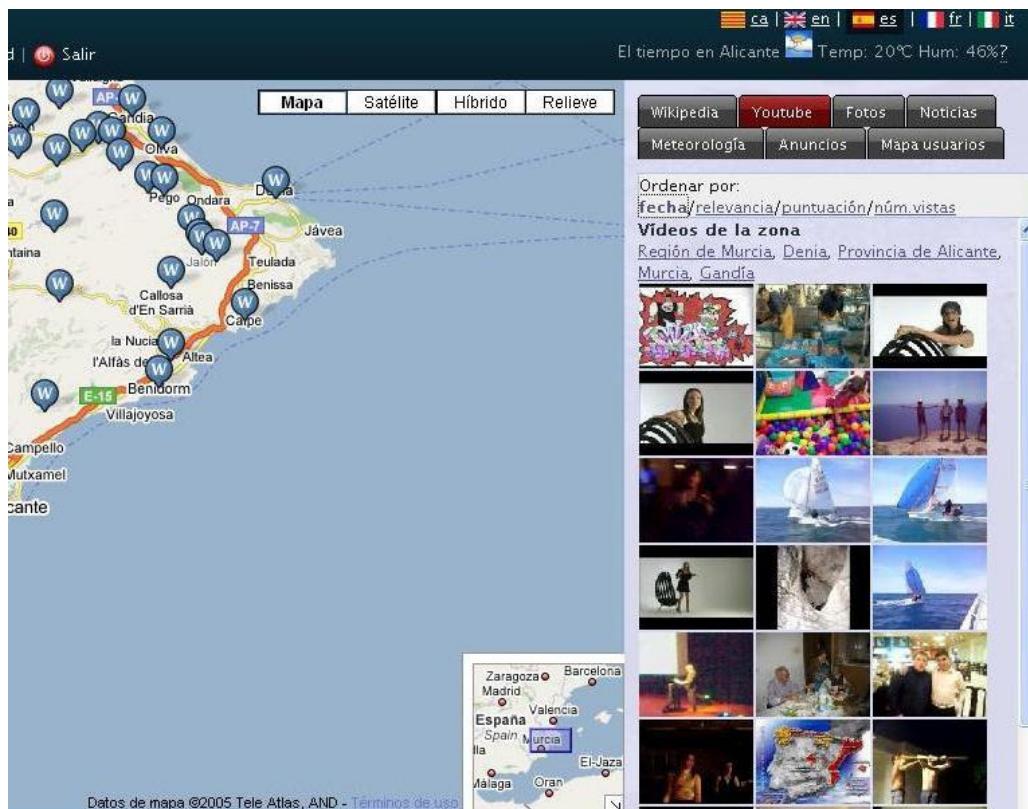
Los puntos violetas son artículos sin población. En la captura anterior se puede ver como la Universidad de Alicante es un punto violeta.

Una vez nos identificamos como usuario registrado, tenemos un par de enlaces adicionales disponibles en cada globo de artículos de Wikipedia. El primero de ellos dice “Actualizar”, y lo que hace es lanzar un proceso que sincroniza el contenido actual del artículo de Wikipedia con el guardado en el proyecto. De esta manera se pueden rectificar fallos en el contenido, ocasionados por haberse obtenido de una versión inadecuada del artículo en Wikipedia. Hay que recordar que cualquiera puede editar el contenido de Wikipedia, por lo que ante un fallo de contenido sería tan fácil como arreglarlo en Wikipedia y proceder a actualizar el punto. Pero en el caso de no saber cómo editar correctamente en Wikipedia está la opción de acudir al enlace “InforFormulario de informe de errormar de error”, donde se podrá dejar un informe de contenido erróneo, bien sea del contenido general del punto, de sus coordenadas geográficas o de su población, de manera que los administradores podamos corregir la información ofrecida por el punto.



## Capa de Youtube

La capa de Youtube muestra una selección de vídeos obtenidos del sitio Youtube, relacionados con el mapa actual del usuario. Para acceder a ella, como es lógico, pincharemos en la segunda pestaña empezando por la izquierda de la fila superior, la que tiene la etiqueta “Youtube”. A continuación una muestra del aspecto de la pantalla cuando seleccionamos la pestaña.



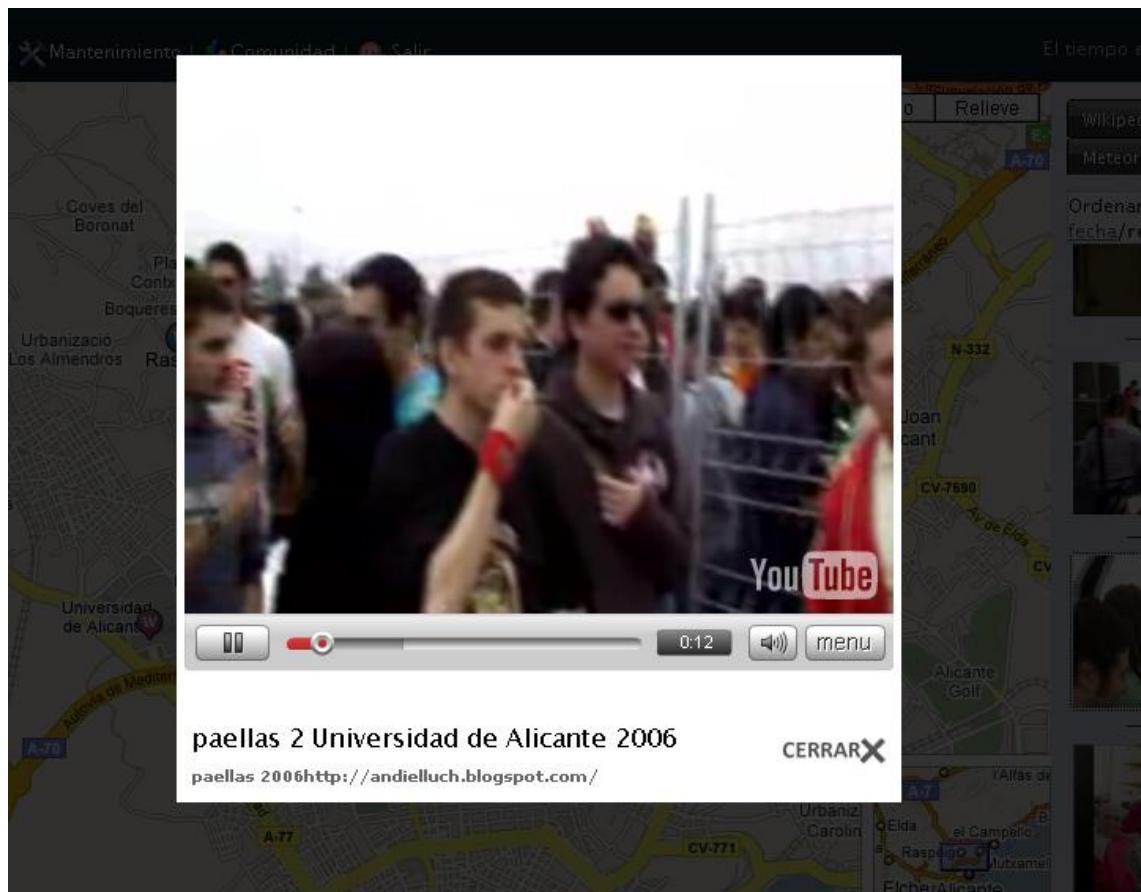
Como se puede ver, a la derecha aparecen una serie de miniaturas, cada una de las cuales enlaza a un vídeo obtenido de Youtube y relacionado con el mapa actualmente activo. Arriba de los vídeos hay dos filtros distintos.

El superior, que contiene los enlaces “fecha/relevancia/puntuación/núm.vistas”, se refiere a los filtros de búsqueda disponibles en Youtube. Por defecto los vídeos aparecen ordenados por fecha, de manera que aparecen primero los más recientes, dando a la capa un mayor dinamismo de contenidos. A medida que cambiamos de filtro, los vídeos se reordenarán, o aparecerán nuevos, y el filtro activo pasará a estar en negrita.

El segundo filtro se refiere a los puntos del mapa por los que se ha buscado los vídeos. En total aparecerán cinco puntos, generalmente los más característicos del mapa actual, predominando aquellos con mayor población, pero no apareciendo dos puntos demasiado cercanos entre ellos. Si se desea que se busque por un punto que no aparece entre estos

cinco, podemos hacer más zoom en el mapa, hasta que aparezca nuestro punto. Al pinchar en el nombre de un punto cambiará ligeramente la vista de los vídeos. La búsqueda se restringirá a ese punto concreto, y aparecerá una miniatura de vídeo por fila, acompañada de una breve descripción de su contenido.

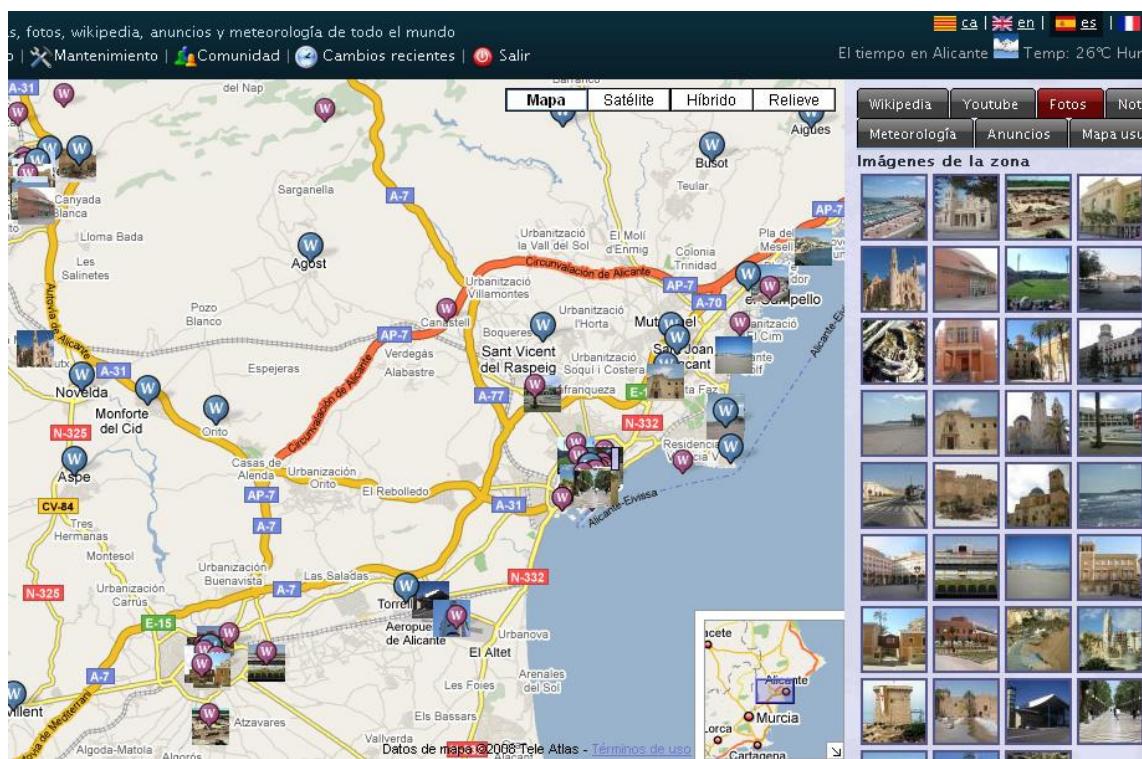
Y bueno, una vez buscado los vídeos que queremos, ya sólo queda visualizarlos. Para ello es tan fácil como pinchar la miniatura deseada, y el vídeo aparecerá en el centro de la imagen, listo para pulsar el "Play", según estamos acostumbrados en los vídeos de Youtube. Para cerrar la ventana del vídeo y volver al mapa basta con seguir el enlace "Cerrar X".



## Capa de Commons

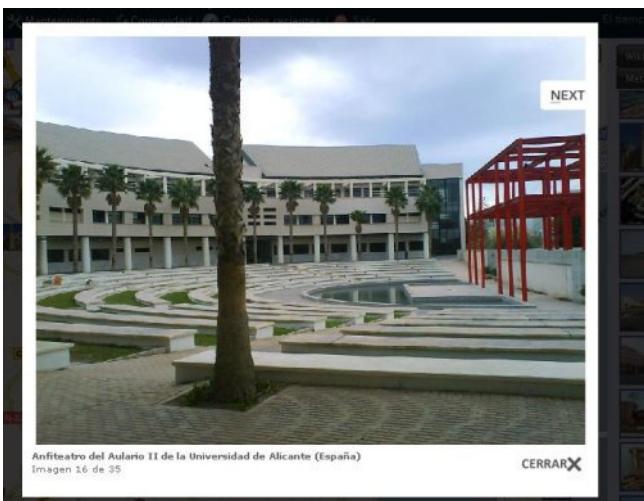
En la capa llamada “Fotos” encontraremos las imágenes procedentes de Wikimedia Commons, el proyecto almacén de ficheros multimedia para los proyectos de la Fundación Wikimedia. Las imágenes almacenadas en Commons están publicadas bajo licencias copy-left, o bien están en el dominio público. Eso significa que se puede hacer cualquier uso de ellas, siempre que se cumplan pequeños requisitos de las licencias copy-left, como no cambiar la licencia al republicarlas y citar siempre la fuente original.

Las imágenes contenidas en el mapa aparecen en miniatura en el lugar en que se han geoposicionado. En el espacio para contenido de las capas de la derecha también se tiene una galería con todas las imágenes mostradas en el mapa. Este es el aspecto que nos encontramos:



Desde la galería de la derecha podemos abrir la imagen que deseemos. Al pinchar una imagen, ésta se abrirá en un lightbox sobre el mapa, con un aspecto muy parecido a la visualización de los vídeos de Youtube. Al pasar el ratón por la parte derecha de la imagen aparecerá un botón “NEXT”, desde el que podremos ir recorriendo toda la galería de las imágenes contenidas en el mapa. Para recorrer la galería en sentido contrario buscaremos el botón “PREV”, en la parte izquierda de la imagen.

En ciertos puntos del mapa puede que coincidan exactamente varias imágenes, como puede ser un recorrido de 360º girando sobre el punto desde que se realizan las fotografías. En este caso todas las imágenes se cargarán en la galería de la derecha, pero como



es evidente, sólo se podrá crear un marcador para una misma coordenada. Sin embargo, todas las imágenes contenidas en las mismas coordenadas se agruparán en ese marcador. Al abrir un marcador nos encontraremos con un globo, que puede tener una imagen, un resumen de hasta 4 imágenes, o una galería para el caso de tener más de 4 imágenes. Al pinchar la miniatura de una imagen en el globo abriremos la imagen con el Slimbo, exactamente igual que se ha mostrado antes. Pero además de la imagen encontraremos la descripción extraída de Commons (menos en los globos con galerías en los que las descripciones ocuparían demasiado), y un enlace a la página de Commons, donde podremos encontrar la licencia de la imagen y e información de su autor.

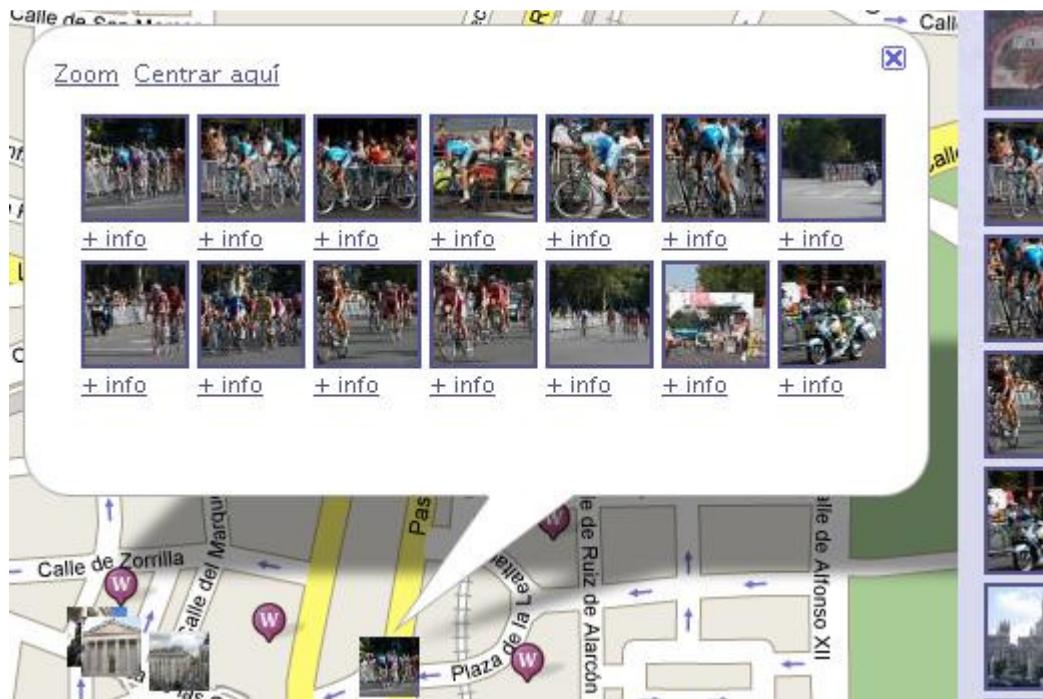
Aspecto de un globo que contiene una única imagen:



Aquí un globo con 3 imágenes en las mismas coordenadas:

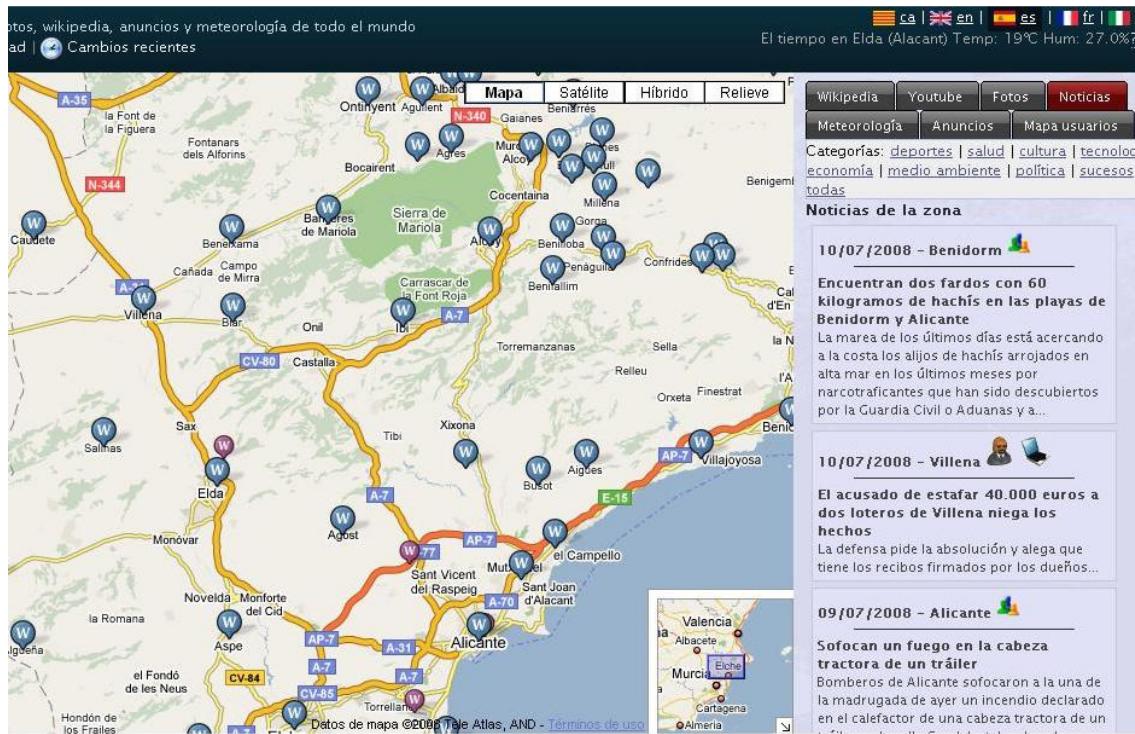


Y aquí un punto con una larga serie de imágenes, mostradas como galería:



## Capa de Noticias

La capa de noticias muestra una selección de noticias, ordenadas por fecha y clasificadas según contenido, relacionadas con el mapa actual del usuario. Las pestaña de la capa de noticias es la cuarta empezando por la izquierda en la fila de arriba de pestañas.



Las noticias se muestran en el área informativa de la derecha, una encima de la otra. Estas noticias corresponden a los marcadores de localidades y regiones visualizados actualmente por el usuario. De cada noticia se ofrece la categoría en que se ha clasificado automáticamente la noticia a través de su contenido, la fecha de publicación, el lugar al que corresponde, el titular, y una breve introducción de la misma. Al pasar el ratón por la noticia el fondo se hará más claro, indicándonos que existe un hipervínculo. Al hacer click con el ratón sobre la noticia nos redirigirá a la web donde se encuentra la noticia original completa.

Al igual que con los vídeos de Youtube, existe un filtro, además de la zona geográfica correspondiente al mapa, para seleccionar noticias. En la parte superior del área informativa tenemos una serie de enlaces a cada una de las categorías en que se puede clasificar una noticia. Estas categorías son: deportes, salud, cultura, tecnología, economía, medio ambiente, política, sucesos. Al pinchar cada uno de estos enlaces se filtrarán las noticias, mostrando sólo las clasificadas en esa categoría. Se puede deshacer el filtro pinchando el enlace a "todas". Una misma noticia puede clasificarse en hasta dos categorías, por lo que es posible que dicha noticia aparezca en dos filtros.

Categorías: deportes | **salud** | cultura | tecnología | economía | medio ambiente | política | sucesos | todas

**Noticias de la zona**

**07/07/2008 - Elche**

**Leones marinos, una ayuda frente al autismo**

Curro y Aragón, dos leones marinos, son parte de una terapia experimental que se está realizando en Elche para mejorar algunas facultades de niños con autismo. ....

**03/07/2008 - Murcia**

**La consejera murciana de Sanidad trata hoy con Soria aspectos en materia de financiación y asistencia sanitaria**

La consejera de Sanidad de Murcia, María Ángeles Palacios, se reunirá hoy, a las 10.00 horas, en el Ministerio de Sanidad y Consumo, con el ministro de Sanidad y Consumo, Bernat Soria, al que expondrá...

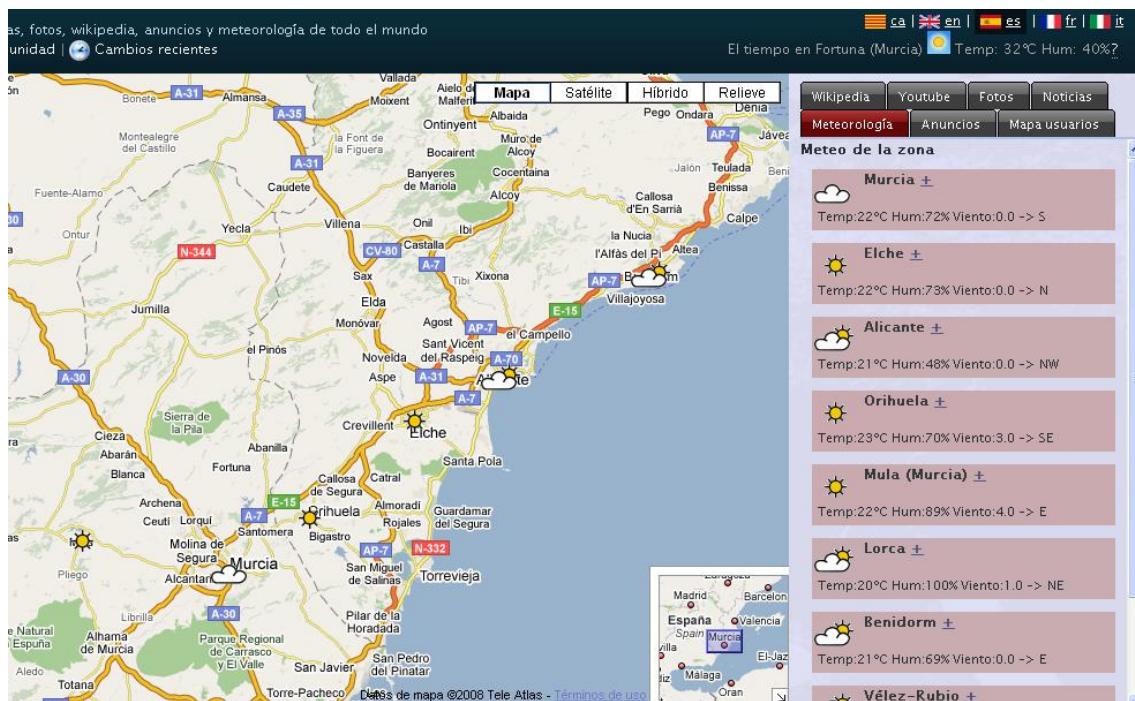
**27/05/2008 - Elche**

**Denuncian ante la síndica la falta de personal en la UCI Neonatal del hospital**

La extracción de noticias nuevas se realiza diariamente, durante la noche. Dicha extracción también puede ordenarse de forma manual desde la parte de mantenimiento para administradores del sitio.

## Capa de Meteorología

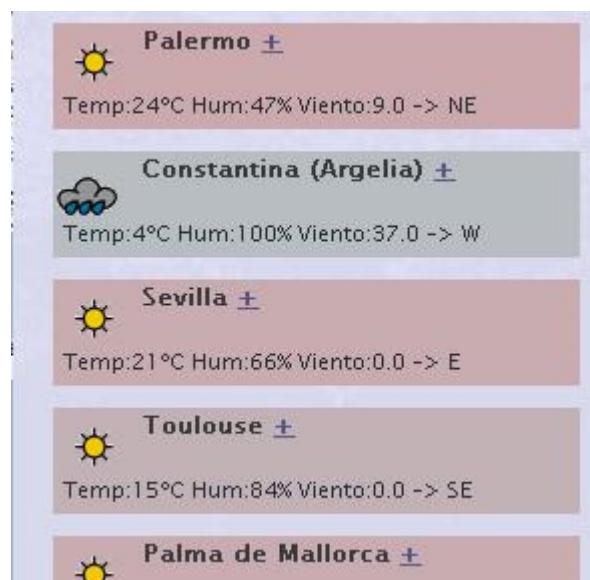
La capa de meteorología muestra información climática de algunos puntos alrededor de todo el mundo. Su aspecto puede resultar familiar a los clásicos mapas meteorológicos, con dibujos de nubes y soles. Para acceder a la capa se llega desde la primera pestaña empezando por la izquierda de la parte inferior de pestañas.



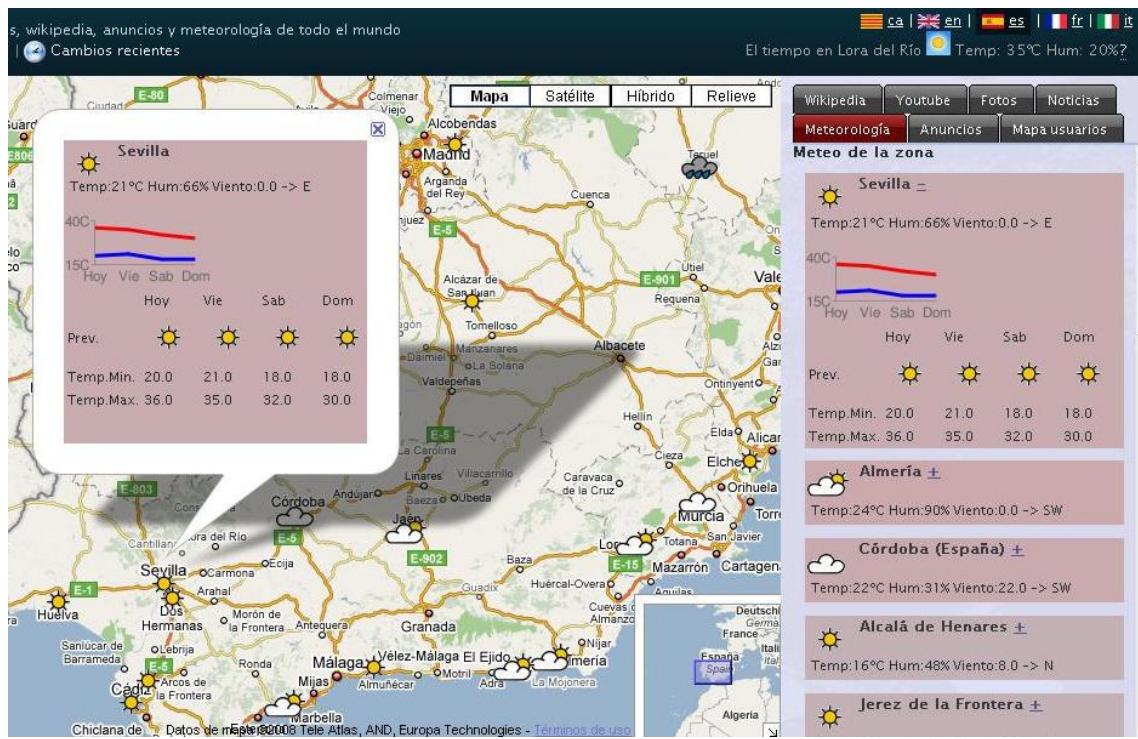
Los puntos con información meteorológica son los disponibles en los servicios externos proveedores de la información: Google y antiguamente MeteoClimatic. La información meteorológica de estos puntos también se actualiza diariamente, aunque igualmente a las noticias puede ejecutarse manualmente por los administradores.

Por cada punto en el mapa con información climática, se dibuja un marcador correspondiente al ícono que representa las condiciones meteorológicas de la localidad. Igualmente, se representan los marcadores con el ícono y la información más básica de temperatura, humedad y viento en el área informativa de la derecha. En esta área el fondo del marcador representa una medida de la temperatura en la localidad. Si el fondo es de un color rojo intenso, significará que la zona tiene una alta temperatura; si por el contrario tiene un tono morado, o incluso azulado, representará un valor bajo de temperatura.

Pero también es posible desplegar los marcadores del área informativa, si se quiere hacer una consulta de una previsión meteorológica de hasta cuatro días de la localidad. Esta previsión se muestra en base a temperaturas máximas y mínimas, representada por una gráfica de estos valores, seguida por una tabla con los valores tomados en la previsión. Esta misma



previsión puede consultarse abriendo el marcador sobre el mapa. Al pinchar el marcador se abrirá un globo informativo con la gráfica y la tabla de valores de la previsión a cuatro días.



## Capa de Anuncios

La capa de anuncios muestra sobre el mapa una serie de anuncios clasificados geoposicionados. A la capa de anuncios se accede desde la segunda pestaña empezando por la izquierda de la capa inferior de pestañas.



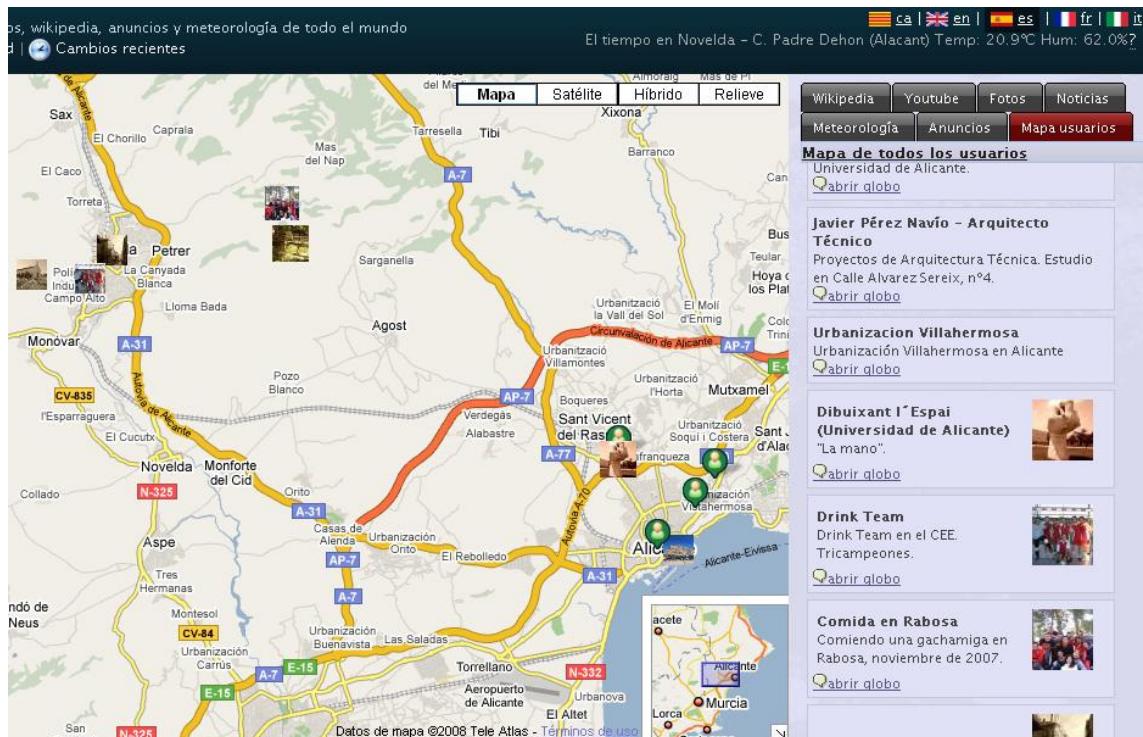
Por cada anuncio se crea un marcador con un icono rojo sobre el mapa. Y también por cada ícono se añade una pequeña descripción del mismo al área informativa de la derecha. El aspecto de esta área informativa es similar al de las noticias. Se ofrece el título del anuncio y una pequeña introducción del mismo. El título es un hipervínculo, que lleva hasta la web donde se encuentra el anuncio original.

El geoposicionamiento de los anuncios se realiza a través de su contenido. Se procesa automáticamente el texto del anuncio, buscando una dirección o localidad donde colocar el marcador. En caso de encontrar una dirección, se busca esta en el servicio de Google Maps, y se usan las coordenadas devueltas para crear el marcador. Si no se dispone de una dirección, pero sí de una localidad de las que tenemos en nuestra base de datos de artículos de Wikipedia, se usan las coordenadas obtenidas de Wikipedia para localizar el anuncio.

Los anuncios se obtienen a través de RSS. En nuestro caso hemos usado RSS procedentes de Loquo, un portal de anuncios clasificados, pero es fácilmente sustituible esta fuente de anuncios por cualquier otra.

## Capa de Usuarios

La capa de mapa de usuarios presenta los contenidos aportados por los usuarios registrados al sitio. Estos contenidos pueden ser puntos informativos sobre el mapa, o bien fotografías acompañadas de una descripción. A la capa de usuarios se accede desde la tercera pestaña empezando por la izquierda de la parte inferior de pestañas.



Los

puntos que no disponen de una fotografía se muestran con un ícono verde. Los que sí que tienen una imagen, muestran la misma fotografía en miniatura como marcador del punto. Los puntos con más visitas (se entiende por visita abrir el marcador mapa) son los que se muestran en primer necesidad de realizar zoom para visualizarlos. También aparecen con la marcador un poco más grande aquellas con más visitas registradas.

Al abrir un marcador, bien pulsando marcador en el mapa, bien accediendo a contenido en el área informativa de la se abrirá un globo con el contenido aportado por el usuario. Se mostrará la imagen, en caso de tenerla, y el texto asociado. En la parte superior hay una serie de enlaces que llevan a la dirección de la imagen original, si la hay, a una página web de referencia aportada optionalmente por el usuario, y un enlace para indicar que se trata de un contenido inapropiado para el proyecto, y que por lo tanto no debería mostrarse.

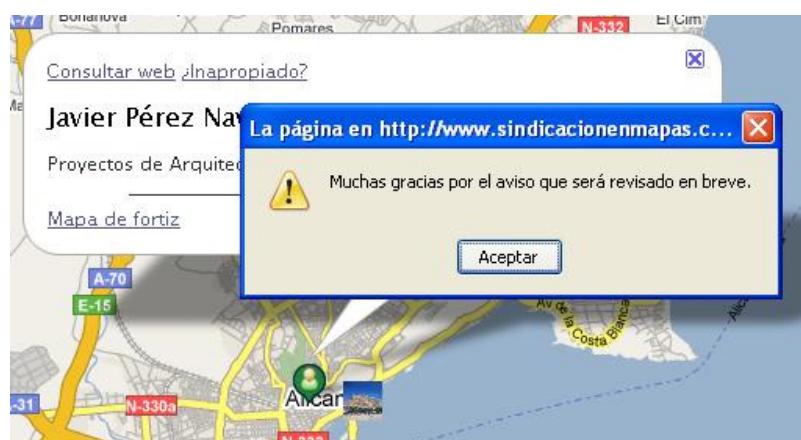


en el lugar, sin

imagen del imágenes

sobre el su derecho, aportado

Los administradores pueden consultar en el apartado de mantenimiento los avisos de contenido inapropiado recibidos, descartando manualmente los contenidos. Pero si se recibe un número determinado de avisos, desde direcciones IP distintas, el punto se retira preventivamente, dejando de mostrarse a los usuarios. Posteriormente un administrador podrá confirmar el retiro, o bien reponer el punto. El número de avisos necesarios para el retiro preventivo es también configurable desde el área de mantenimiento.



En el globo abierto, en su parte inferior, también aparecerá un enlace que dirá "Mapa de usuario", donde usuario es el nombre del usuario que aportó el contenido. Al pulsar el enlace se mostrarán únicamente sobre el mapa los contenidos subidos por el usuario en cuestión. Si el usuario que está navegando se ha identificado en el sistema, también podrá ver en la parte superior del área informativa de la derecha un enlace a "Mi mapa", donde se realizará un filtro de los contenidos aportados únicamente por el usuario que está navegando. Cuando un usuario está viendo su propio mapa podrá ver todos los puntos subidos por él, tanto los públicos como los privados. Cuando se consulta otro mapa sólo aparecen los puntos marcados como públicos. Para eliminar el filtro basta con seguir el enlace "Mapa de todos los usuarios" que se encuentra en la parte superior del área informativa de la derecha.

A screenshot of a map application for the region of Murcia and Alicante, Spain. The main map shows major roads like A-30, AP-7, and E-15. A callout box highlights a specific location: "Consultar web Imagen original | Inapropiado" (Check website Original image | Inappropriate). Below this, there's a thumbnail image and the text "Ermita de las Cañas (Elda)". To the right, there's a sidebar with various links: "Wikipedia", "Youtube", "Fotos", "Noticias", "Meteorología", "Anuncios", and "Mapa usuarios" (which is highlighted in red). Below these are several cards showing user-submitted content: "Dibujant l' Espai (Universidad de Alicante) 'La mano.'", "Calle Nueva (Elda)", "Ermita de las Cañas (Elda)", "Ojos del Diablo (Tobarra)", and "Santuario de la Encarnación (Tobarra)". Each card includes a thumbnail image and a "Abrir globo" (Open globe) link.

## Extracción de contenido de MediaWiki

El elemento central de la arquitectura lógica de la información ofrecida por el proyecto es la base de datos recopilada de puntos, correspondientes a artículos enciclopédicos de Wikipedia, relacionados con sus coordenadas geográficas. Estos puntos son usados a la hora de agrupar noticias relacionadas con ellos, la información meteorológica disponible, búsqueda de vídeos relacionados de Youtube, y sirven para localizar los anuncios clasificados de los que no se dispone de una localización más concreta.

Actualmente hay disponibles en Internet grandes bases de datos de localizaciones asociadas a coordenadas geográficas. Sin embargo, en el proyecto se ha optado por limitar los puntos incluidos a los que se les ha podido asociar a un artículo enciclopédico de Wikipedia. Además, pensamos que era la mejor forma de mantener nomenclaturas y textos descriptivos en castellano, dado que las bases de datos existentes suelen estar en inglés. En total hay disponibles más de 37.000 puntos, pudiendo añadir además cualquier usuario registrado nuevos puntos que tengan artículo en Wikipedia.

Para la extracción de la información se ha usado el API de Wikipedia<sup>4</sup>, una funcionalidad del software MediaWiki que ofrece una interfaz con el proyecto en lenguaje XML. Concretamente se ha usado la implementación en Python de la librería pywikipedia<sup>5</sup>, que ofrece una interfaz sobre el API de Wikipedia. Esta librería es ampliamente usada por usuarios de Wikipedia para implementar scripts automáticos de mantenimiento sobre el proyecto, llamados bots.

De esta manera se han implementado una serie de scripts python, que han recorrido los artículos de Wikipedia, buscando en el texto de los mismos coordenadas geográficas que puedan situar el artículo en un mapa. Así se ha poblado la base de datos con una pequeña introducción del artículo, y de su traducción en otros idiomas si hay otras versiones traducidas del artículo en las distintas ediciones lingüísticas de Wikipedia. En total se han guardado traducciones en siete idiomas: español, inglés, francés, italiano, alemán, portugués y catalán. También se ha guardado la población de los artículos correspondientes a localidades y divisiones administrativas, extrayendo el dato del texto del artículo.

### Funcionamiento de los scripts python

Para el recorrido inicial de la base de datos de artículos de Wikipedia se usó la clase de la librería pywikipedia *pagegenerators*. Esta clase ofrece métodos para poder designar una serie de filtros y reglas que extraerán y procesarán un subconjunto de la cantidad total de artículos. De esta manera se lanza un proceso que va obteniendo artículo por artículo, en consultas al API de Wikipedia de unos 50 artículos mediante la apertura de sockets, con una eficiente implementación de threads y balanceo de carga, evitando la sobrecarga de llamadas al servidor de Wikipedia y también la carga de procesamiento del proceso local, pudiendo dejarlo ejecutándose en background sin problemas.

En nuestro caso el generador de páginas debía recorrer todos los artículos disponibles, lo cual fue una tarea bastante costosa en términos de tiempo. Un recorrido y procesamiento de la Wikipedia en idioma español tardaba alrededor de una semana. Para ello lo que se hacía era realizar un recorrido por orden alfabético, de manera que si el proceso tenía que ser detenido, poder volver a lanzarlo desde el último artículo procesado.

<sup>4</sup> El API de Wikipedia se puede consultar en <http://es.wikipedia.org/w/api.php>

<sup>5</sup> La librería pywikipedia está disponible en [http://meta.wikimedia.org/wiki/Using\\_the\\_python\\_wikipediabot](http://meta.wikimedia.org/wiki/Using_the_python_wikipediabot)

De este modo, la clase `pagegenerator` puede ser recorrida con un bucle que recorre todos los objetos `page` que contiene. La clase `page` es la representación de un artículo en cualquier sistema wiki de mediawiki, para el que basta definir su título y el sitio donde se hospeda. Esta clase dispone de una buena colección de métodos para interactuar con el artículo, basadas en las llamadas disponibles al API de Wikipedia.

Por cada artículo obtenido se comprueba que esté incluido en el espacio de nombres (*namespace*) principal de artículos, es decir, que no se trate una página especial de las usadas en los sistemas Mediawiki, como categorías, páginas de discusión, imágenes, páginas de usuario, etc. De esta manera sabemos que las páginas procesadas corresponden únicamente a artículo válidos de la enciclopedia. También se comprueba que el artículo exista realmente (es posible que haya sido borrado antes de proceder a su procesamiento), y que no se trate de una página de redirección a otra. Una vez efectuadas estas comprobaciones se obtiene el texto almacenado para el artículo.

Este texto se recupera directamente de la base de datos de Wikipedia, por lo que corresponde al texto escrito en código wiki, y no en html. Este aspecto es muy importante a la hora de diseñar las expresiones regulares que recuperarán la información buscada de la página.

Las expresiones regulares que buscan las coordenadas de ubicación de los artículos son capaces de reconocerlas aunque estén escritas en una gran variedad de formatos, fruto del uso de plantillas generadoras de código wiki, ampliamente usadas en Wikipedia. Así se puede comparar con las coordenadas que es capaz de reconocer el sistema de posicionamiento de Wikipedia en Google Maps, que únicamente incluye los artículos con una determinada marca en el principio del artículo:  `{{coord|40|26|N|3|41|O|type:city|display=inline,title}}`. Sin embargo el uso de estas marcas en la Wikipedia en español no está demasiado extensivo, por lo que consideramos que valía la pena generar expresiones regulares más complejas que reconociesen la mayor cantidad posible de formatos. Así los formatos reconocibles por los scripts usados en Sindicación en mapas son:

- `{{coord|40|26|N|3|41|O|}}`,  `{{coor dm|xxx|}}`,  `{{coor dms|xxx}}`. Plantillas de especificación de coordenadas genéricas.
- `|longitude = 2.351074|latitude = 48.86223`. Usado en la plantilla de localidades francesas.
- `|latitudineGradi = 41|latitudineMinuti = 54|latitudineSecondi = 0|longitudineGradi = 12|longitudineMinuti = 30|longitudineSecondi = 0`. Usado en la plantilla de localidades italianas.
- `|latd = 40|latm = 43 |lats = |latNS = N |longd = 74 |longm = 00 |longs = |longEW = W`. Usado en la plantilla de localidades importada desde la Wikipedia en inglés.
- `|latitud = 53.075878|longitud = 8.807311`. Usado en la plantilla de localidades alemanas.
- `north_coord = 53.3472 | west_coord = 6.2689 |`. Usado en la plantilla de localidades irlandesas.
- `|nis=31005`. Usado en la plantilla de localidades belgas. El NIS es el código de la ciudad, el cual hay que buscar en un fichero de texto que contiene todos los NIS de Bélgica, asociados a sus coordenadas y su población.
- `4° 39' 0" N 74° 3' 0" O`. Codificación literal, con amplia gama de reconocimiento de caracteres distintos para `,`, `'`.
- `http://maps.google.com/maps?ll=41.890170,12.491992`. Enlaces a vistas del artículo en Google Maps.

- <http://xxxxx?lat=41.890170&lon=12.491992>. Enlaces a vistas del artículo en otros servicios de mapas.

Del mismo modo que se detecta y extrae la posición geográfica de los artículos, se realiza una búsqueda de la posible población, en el caso de tratarse de una localidad o una región geográfica. Así, los patrones buscados y detectados en los artículos son:

- 1.322.343 habitantes. Patrón comúnmente usado dentro del texto de los artículos.
- \habitantes (población, abitanti, sans, population, censo, etc) = 434.234. Usado en las distintas plantillas de localidades en diversos idiomas.
- \nis=31005. Usado en la plantilla de localidades belgas. El NIS es el código de la ciudad, el cual hay que buscar en un fichero de texto que contiene todos los NIS de Bélgica, asociados a sus coordenadas y su población.</nowiki></code>

En caso de no encontrar ninguno de estos patrones se supone que el artículo no corresponde a una localidad, por lo que se asigna su población como cero. Al extraer la información de la base de datos, se consideran localidades o regiones geográficas los marcadores correspondientes a artículos que tienen guardada una población mayor que cero.

Una vez obtenidas las coordenadas y la población, se procesa el texto del artículo, que se encuentra en formato wiki, para extraer una introducción en texto plano. Esto lo realiza la función Código:limpiarWikitexto(texto, latitud=0, longitud=0), implementada por nosotros. Esta función se encarga de eliminar comentarios, imágenes, referencias, etiquetas html, títulos de secciones, secciones finales de "Enlaces externos" o "Véase también"; se sustituyen plantillas de coordenadas por el valor numérico de las coordenadas ya procesadas; se eliminan plantillas, infoboxes, códigos de marca de enlaces internos, enlaces externos, categorías, interwikis, negritas, cursivas, códigos de listas, etc. La introducción obtenida, en caso de ser necesario, es recortada para que no supere los 500 caracteres.

A continuación se comprueba si el artículo tiene interwikis a los idiomas disponibles en el proyecto. Es decir, que si existe la versión del artículo en las Wikipedias en otros idiomas. Esto se hace buscando el patrón [[en:Título]], lo cual significa que el artículo "Título" tiene una versión en la Wikipedia con código "en" (en inglés). De esta manera se crea una nueva instancia de la clase Page, con el título y el código de wikipedia obtenidos. Este objeto Page se procesa del mismo modo que se ha hecho con el de la versión española, salvo en la obtención de las coordenadas y la población, que ya se han extraído. Así se obtiene la introducción del artículo en el idioma correspondiente, junto a su título traducido. Si no se encuentra el interwiki buscado significa que no está disponible la versión traducida a ese idioma, por lo que el título y la traducción de ese idioma se guardan en base de datos como cadena vacía. En la base de datos hemos almacenada versiones de los artículos en inglés, francés, italiano, portugués, alemán y catalán.

## Tandas de extracción

Con la base del script explicado hasta ahora se recorrió la Wikipedia en idioma español, en su versión de marzo/abril de 2008, extrayendo todos los artículos con coordenadas reconocibles. La base de datos se pobló con unos 22.000 artículos con introducciones y títulos en seis idiomas.

Posteriormente se modificó el script para recorrer la Wikipedia en inglés. La idea era encontrar los artículos geoposicionados en la Wikipedia en inglés y que a su vez tuviesen una traducción al menos en español. Así para cada artículo que cumplía estos requisitos, se comprobaba si ya existía en la base de

datos. Si no se encontraba, se procesaba la traducción al español, con las coordenadas encontradas en la versión inglesa. Recorriendo la versión de Wikipedia en inglés del mes de abril de 2008 se incrementó el número de artículos disponibles en la base de datos en un 70%, llegando a una cifra superior a los 37.000 artículos.

### Recopilación de imágenes de Commons

El proceso de extracción de las imágenes de Wikimedia Commons ha sido análogo al de los artículos de Wikipedia. El proyecto Commons es un almacen de ficheros multimedia (actualmente casi 3 millones), la mayoría de ellos imágenes, usados en el resto de proyectos de la Fundación Wikimedia. Como el resto de los proyectos Wikimedia está implementado bajo el software MediaWiki, por lo que la misma clase de scripts python usados en los artículos de Wikipedia podía usarse para procesar las imágenes geoposicionadas de Commons.

Cada archivo almacenado en Commons tiene una página de MediaWiki (similar a un artículo), donde se especifica toda la información referente al archivo. Muchas imágenes incluyen en este texto su posición geográfica y una descripción de la imagen, a veces en más de un idioma (Commons, a diferencia de otros proyectos, no tiene versiones en distintos idiomas). Por lo tanto el script se encargó de procesar todas estas páginas, buscando las imágenes con información de su geoposicionamiento, de la misma manera que se hizo con los artículos de Wikipedia.

Una vez encontrada una página con coordenadas geográficas, obtenemos de su objeto Page la url de su página, en lugar de su texto wiki. De esta manera se abre una conexión con dicha url, para obtener el contenido html de la página de descripción de la imagen, con el fin de poder encontrar el enlace a la imagen guardada en el servidor de Commons. Pero en este paso nos encontramos con un problema, ya que al intentar abrir la conexión con la url desde python, el servidor Commons nos denegaba la petición, de manera que no podíamos obtener el código html. Finalmente lo solucionamos añadiendo una cabecera a la petición http, en la que decíamos que éramos un navegador web estándar (al acceder con un navegador no hay problemas): `request.add_header('User-Agent', 'Mozilla/4.0')`. De esta manera, el servidor Commons confiaba en nosotros, y nos devolvía el código html de la página.

Una vez obtenido el código html se buscaba la etiqueta en la que se enlazaba a la imagen. En un principio almacenábamos la url de la imagen original almacenada en Commons, y mediante la herramienta thumb<sup>6</sup> de MediaWiki tratábamos de generar las miniaturas necesarias en el proyecto en tiempo de ejecución. Pero la herramienta Lightbox usada no era compatible con las url generadas con el thumb.php de MediaWiki, y además teníamos problemas a la hora de controlar las proporciones de alto y ancho. Cargar en el servidor las imágenes originales tampoco era una medida muy eficiente, ya que normalmente las imágenes de Commons son fotos de muy alta calidad, y por lo tanto muy pesadas, obligando al usuario a descargárselas completas para finalmente sólo ver una versión reducida de los originales. Por todo ello, la solución final fue cambiar la url de las imágenes almacenadas, guardando las url de las versiones reducidas de 800 píxeles de anchura<sup>7</sup> e indicar al usuario un enlace donde puede acceder a la imagen original sin reducir sólo si él lo desea. A partir de esta versión reducida también generábamos, mediante el módulo de miniaturas de imágenes, una miniatura de cada imagen de unos 100x100 píxeles, siendo la miniatura

<sup>6</sup> Un ejemplo de miniatura puede verse en

<http://commons.wikimedia.org/w/thumb.php?f=FirePhotography.jpg&width=100>

<sup>7</sup> La imagen con 800 píxeles de ancho

<http://upload.wikimedia.org/wikipedia/commons/thumb/9/95/FirePhotography.jpg/800px-FirePhotography.jpg>

mostrada en los marcadores de las imágenes, y en el contenido del área de información de la derecha de la página principal.

Con la imagen ya guardada sólo quedaba por obtener el texto informativo relativo a la imagen, obtenido del texto en código wiki del artículo de la imagen, y procesado para obtener una versión en texto plano.

El script se lanzó con la versión de Commons de marzo/abril de 2008, encontrando unas 33.000 imágenes geoposicionadas.

## Importación de SQL

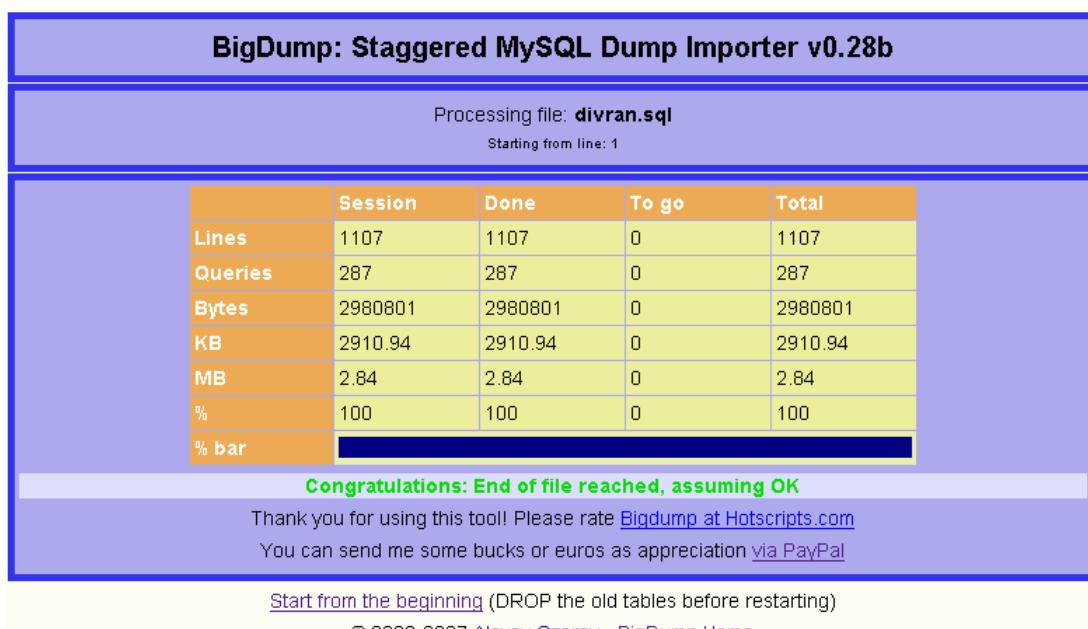
La gran cantidad de datos con los que trabajábamos en local nos ha supuesto varios problemas para ser exportados a la base de datos del servidor. Hemos utilizado varias técnicas:

**Importación a través de archivo de texto y comando de consola:** Este tipo de importación presentaba problemas con la codificación de caracteres, así que la descartamos.

**Importación copiando y pegando instrucciones SQL en PHPMyAdmin:** Hemos empleado este método tanto para pocas instrucciones (con las que funcionaba bien) como particionando miles de instrucciones SQL en grupos más pequeños (de unas 500 instrucciones) para importar bases de datos grandes o encontrar algunos errores de sintaxis que PHPMyAdmin no era capaz de localizar correctamente.

**Importación a través de aplicación Java:** También realizamos un módulo en la aplicación de Java para importar archivos de texto grandes con instrucciones SQL. Este método resultaba rápido pero debido al número de instrucciones a realizar en ocasiones teníamos problemas con la conexión con la base de datos y/o memoria.

**Importación a través de BigDump:** BigDump<sup>8</sup> es un sencillo script PHP que ejecuta las instrucciones SQL contenidas en un archivo de texto (sin comprimir o comprimido con targz). Nos ha sido verdaderamente útil y ha sido el método por el que finalmente hemos optado para subir las bases de datos locales (wikipedia, noticias, anuncios...)



[Start from the beginning](#) (DROP the old tables before restarting)

© 2003-2007 [Alexey Ozerov](#) - [BigDump Home](#)

<sup>8</sup> BigDump puede descargarse en [http://www.озеров.de/bigdump.php](http://www.ozеров.de/bigdump.php)

## Eliminación de paréntesis

En Wikipedia son habituales las páginas de desambiguación. Estas páginas se asemejan a las entradas de los diccionarios e incluyen diversos enlaces a artículos que difieren normalmente en el contexto pero que se llaman igual. Así, Alicante puede ser población o provincia, Salinas se puede referir a una población de España o a una de otros 6 países, o el nombre de una población puede ser también un objeto. Para establecer el contexto se suelen añadir al nombre del artículo un contexto entre paréntesis para diferenciarlo de los otros contextos (desambiguación).

Inicialmente obteníamos los artículos de Wikipedia y no procesábamos posteriormente el nombre, de forma que en ocasiones teníamos en la base de datos poblaciones con estos "contextos" añadidos, pero que en nuestro caso no desambiguaban (algunos distinguen poblaciones de objetos pero en nuestro caso no existe ambigüedad, pues sólo necesitamos las poblaciones). Este contenido era molesto porque no podíamos aplicar el resto de funcionalidades (búsqueda de noticias, vídeos de Wikipedia, etcétera) por lo que implementamos un módulo que para cada artículo de Wikipedia de nuestra base de datos, si tiene ese contexto añadido, comprobamos si existe algún otro artículo con el mismo nombre. Si no existe ninguno más, entonces no hay ambigüedad, por lo que eliminamos la información de contexto.

### Ejemplo

Por ejemplo, **Alerta** es un sustantivo y además también es una población de Perú. En la página de desambiguación obtenemos:

#### Alerta (desambiguación)

**Alerta** puede referirse a:

- [Alerta](#), período anterior a la ocurrencia de un desastre;
- [Alerta](#), localidad peruana;
- [Alerta](#), programa de TV de la RCTV venezolana;
- [Alerta](#), diario español;
- etc.

donde, a pesar de haber varios sentidos para *Alerta*, sólo uno indica es de una población. El artículo para la población se llama *Alerta (Perú)*.

#### Alerta (Perú)

*Para otros usos de este término véase [Alerta \(desambiguación\)](#).*

La localidad [peruana](#) de **Alerta** es un pueblo ubicado en el [distrito de Purús](#), provi  
[Madre de Dios](#) a 696 km. al sur este de la ciudad capital del departamento, [Pucallpa](#).

Sus coordenadas geográficas son  11°39'18"S 69°14'6"O.

En esta localidad se encuentra el [Aeropuerto de Alerta](#).

por lo que en nuestra base de datos sólo existe un registro para *Alerta*. Podemos proceder a eliminar "*(Perú*)".

En realidad, para no perder la referencia al artículo en Wikipedia, utilizamos otro campo en la base de datos para almacenar esta información (en concreto "nombre" para el nombre no ambiguo y "nombreES" para el nombre en Wikipedia en español).

## Corrector de coordenadas Wikipedia

Las coordenadas de los puntos de Wikipedia son introducidas por los usuarios. En ocasiones estas coordenadas son incorrectas (se ha puesto latitud o longitud negativas y eran positivas o viceversa, no tienen una precisión suficiente, etc). Para poder solventar este problema hemos implementado un módulo de mantenimiento de corrección de coordenadas que nos sirve para:

1. Corregir la coordenada de la población sin necesidad de modificar el artículo de Wikipedia manualmente y luego actualizar nuestro contenido.
2. Ser más precisos con el buscador de ciudades incluido dentro del mapa, al usar el mismo API de Google.

## Funcionamiento

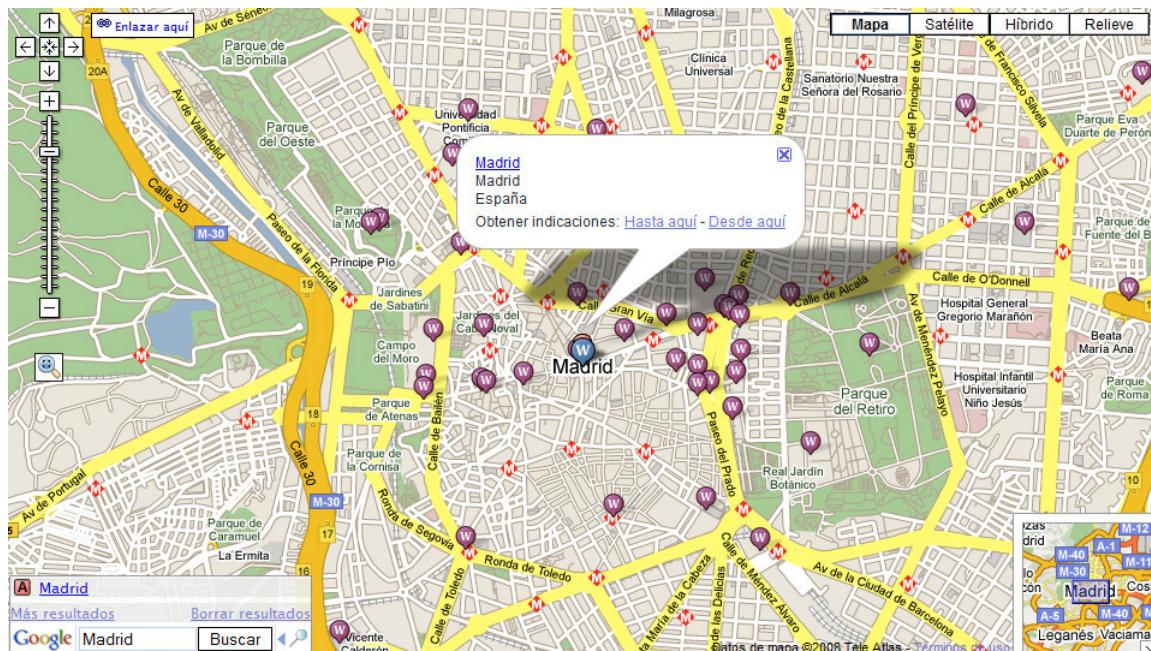
Hemos utilizado el geoposicionador proporcionado por Google que recibe una dirección y devuelve como salida distintos puntos que se corresponden con la dirección o población buscadas, y que incluyen las coordenadas. Estas coordenadas son comparadas con las que tenemos almacenadas y si la distancia entre ambos puntos es inferior a un umbral, asignamos las coordenadas proporcionadas por Google. Además probamos a cambiar el signo de la latitud y/o de la longitud del punto por si el error está en el signo (es bastante habitual este tipo de incorrecciones) y comprobamos la distancia al punto de la misma forma.

Estableciendo un umbral conseguimos que no se traslade en exceso un punto, dado que es posible que pase a señalar un punto de población distinto del que señalaba inicialmente (por ejemplo si varias poblaciones se llaman de la misma forma).

Esta corrección del punto también es comprobada cuando se sube un nuevo artículo de Wikipedia.

## Ejemplo

Si buscamos *Madrid* en el buscador de Google integrado en el mapa vemos como coincide exactamente con nuestro punto de Wikipedia localizado en Madrid:



## Clasificador de puntos

En la aplicación de Java hemos implementado clasificadores de puntos para los puntos de contenido de información de Wikipedia, Commons y puntos propios. Estos clasificadores tratan de organizar los puntos de tal forma que quieren repartidos en el mapa y no se aglomeren en determinadas zonas. La implementación de estos módulos surge a raíz de que debíamos limitar el número de puntos mostrados en el mapa y creímos conveniente ordenarlos de alguna forma. Inicialmente está ordenación se hacía según el número de habitantes de manera que se mostraban los 50 puntos con mayor población dentro de esa región. Aunque para algunas zonas funcionaba bien, para otras los puntos organizaban juntándose en una zona pequeña. Tal es el caso de España, donde se unían muchos puntos en la Comunidad de Madrid.

El método que hemos seguido para realizar la ordenación se basa en la división sucesiva del mapa en regiones, asignando un valor a cada punto por el que luego se ordenan. Hemos utilizado un campo orden que, inicialmente, se establece a 0. Inicialmente se establece el número de niveles de orden (por ejemplo 15) y se va decrementando con cada iteración. El valor del número de niveles es el valor orden establecido a los puntos en la primera iteración.

En la primera interacción el mapa se divide en dos regiones, una con longitud desde -180º a 0º y la otra con longitud de 0º a 180º (ambas con latitud de -90º a 90º). En el caso de Wikipedia se asigna una prioridad máxima al punto de mayor población. En el caso de Commons y puntos propios se hace por longitud de descripción descendente, para dar prioridad a los puntos con una descripción más extensa.

En las sucesivas iteraciones cada región se divide en 2 regiones del mismo tamaño y al punto con más población (o mayor descripción) de cada región se asigna un valor orden igual al máximo entre el valor que contenían y el valor a asignar en esa iteración.

Hemos realizado una poda de forma que no se continúe el algoritmo en aquellas regiones que no contengan un número mínimo de puntos. De esta forma, se reduce el número de cálculos y no es necesario hacer  $2^n$  regiones.

El método funciona bastante bien, pero al ser calculado sobre los puntos existentes, es recomendable realizarlo periódicamente si se han añadido nuevos puntos (una vez al mes es más que suficiente).

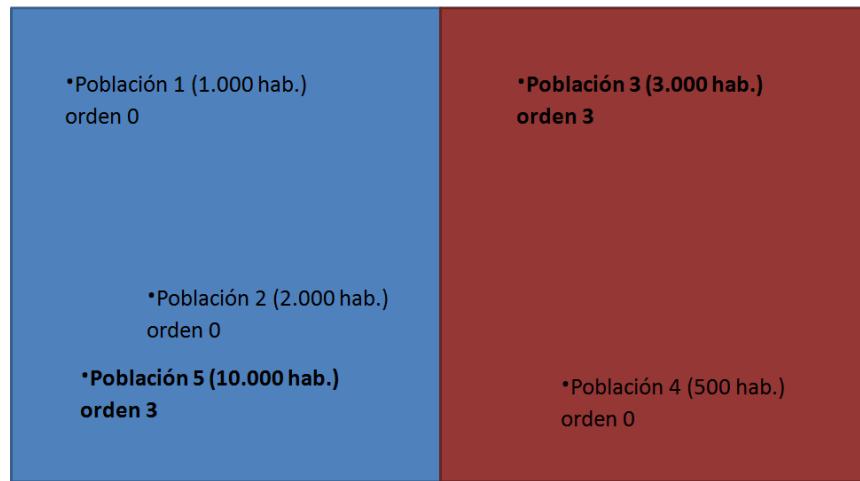
Con este método, los puntos con poca población situados en zonas poco habitadas tienen mayor peso, y los puntos con mucha población, pero cercanos a puntos con mayor población, pierden peso. Así se distribuyen los puntos mostrados por todo el mapa.

Algoritmo

## Iteración 1 – División



## Iteración 1 – Asignación de orden 3



## Iteración 2 -División



## Iteración 2 –Asignación de orden 2



## Iteración 3 -División



## Iteración 3 –Asignación de orden 1



## Ejemplo de funcionamiento

Vista del mapa global

Antes de aplicar el algoritmo, con ordenación por población descendente, vemos cómo los puntos mostrados se corresponden con puntos de India y oriente en su mayor parte, con algunos también por Europa y América, pero no están distribuidos.



Ordenando con el campo calculado:

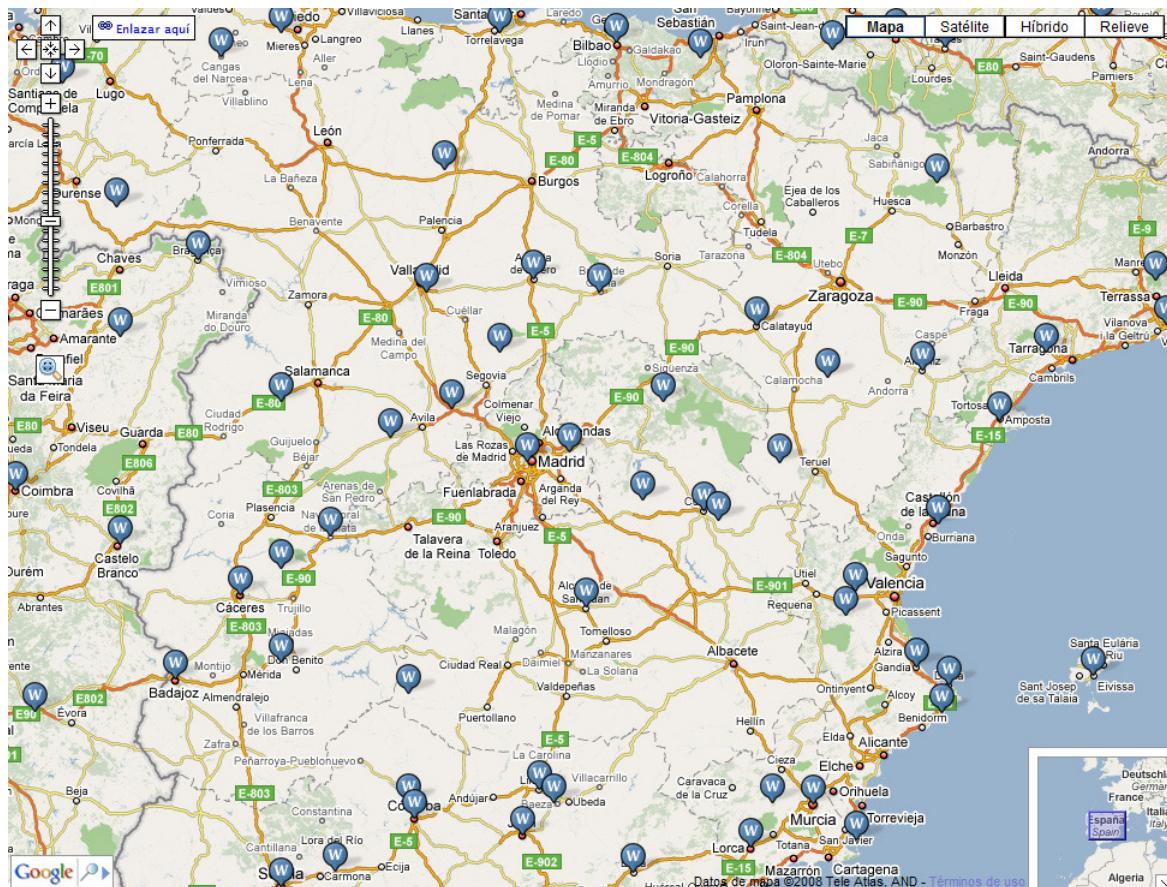


## Vista del mapa de España

Antes de aplicar el algoritmo, con ordenación por población descendente, hace que en Madrid se unan muchos puntos y también en la provincia de Alicante.



Ordenando con el campo calculado:



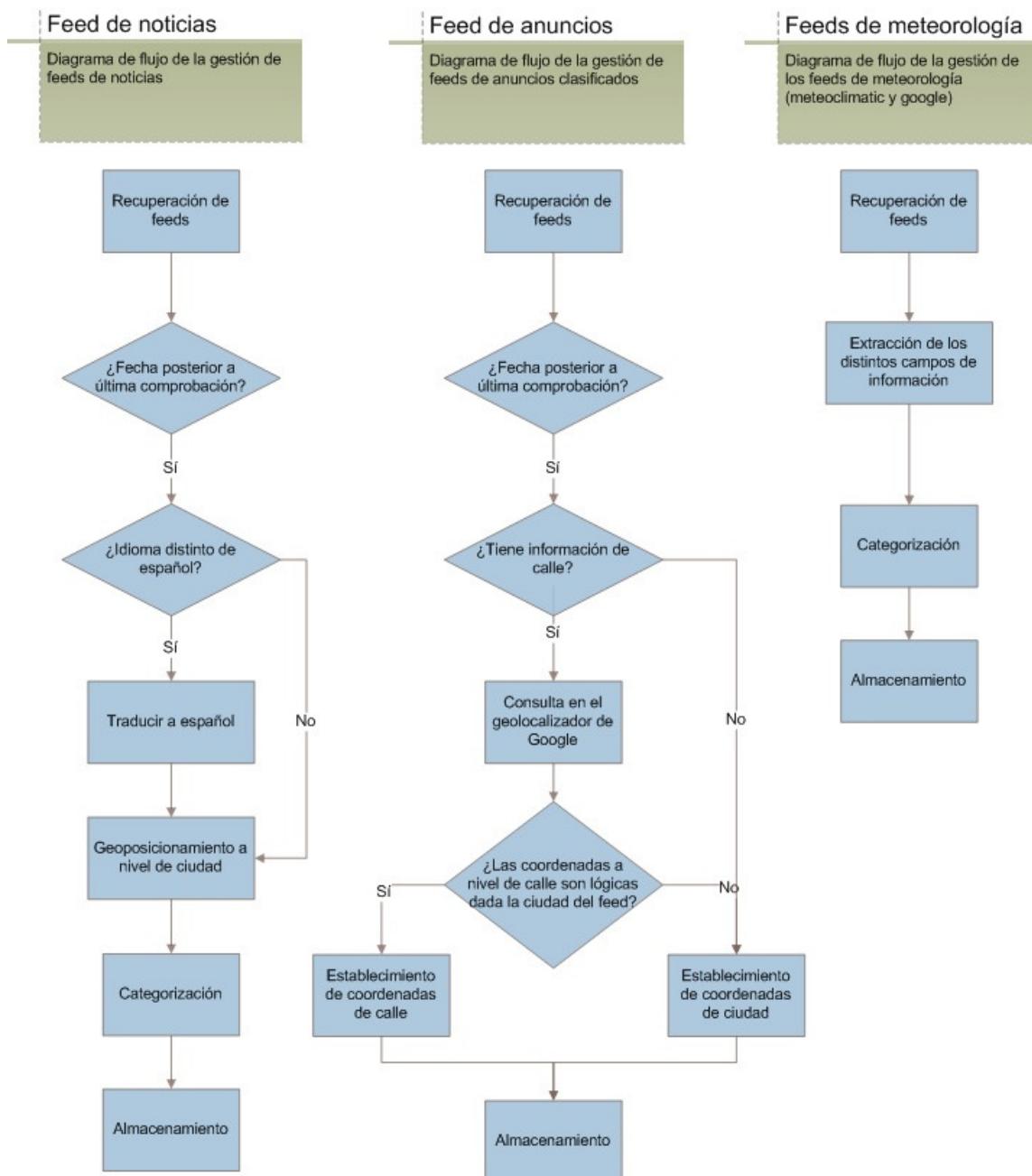
## Gestión de feeds

La mayor parte de las fuentes de datos empleadas son difíciles de ser consultadas en el mismo momento de la navegación del usuario en los respectivos sitios web de origen, dado que es necesario realizar una consulta para cada estado del mapa. Esto es, que cada vez que el usuario realiza zoom o desplaza el mapa es necesario realizar una consulta. Además, diversas capas de datos utilizan varias fuentes (por ejemplo en la capa de Youtube también se muestra la meteorología del punto central del mapa y los puntos de Wikipedia).

Al tiempo necesario para obtener estos datos se añade la sobrecarga del servidor, tanto a nivel de procesamiento como de ancho de banda. La solución más factible era almacenar los datos en bases de datos en el servidor. Esto mejora mucho el tiempo de respuesta y reduce el ancho de banda (dado que no es necesario reprocesar los datos y sólo se devuelven a la interfaz los datos necesarios).

Los feeds son actualizados realizando una llamada a fuentes.jsp. Dicha llamada está establecida en una tabla cron. En el sistema operativo Unix, cron es un administrador regular de procesos en segundo plano (demonio) que ejecuta programas a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el archivo crontab. Es una forma automática de lanzar la actualización de los feeds. En la configuración actual está establecido a todos los días a las 5 de la mañana, pues es una hora lo suficientemente temprana como para no sobrecargar el servidor con visitas de usuarios y es un momento adecuado para poder obtener las noticias de ese día.

El esquema básico de los feeds que se actualizan periódicamente es el siguiente.



A continuación describimos los feeds que se actualizan de esta forma.

### Feeds de noticias

Se añaden las nuevas noticias a la base de datos, llevando a cabo un proceso de geoposicionamiento y categorización de las mismas. Las noticias pertenecen a periódicos de distintos ámbitos geográficos y temáticas. En el caso de noticias de periódicos escritos en idiomas distintos del castellano, estas noticias son traducidas previamente a su posicionamiento y categorización, pero son almacenadas en su lengua original.

### Feeds de anuncios clasificados

Se añaden nuevos anuncios clasificados a la base de datos, obtenidos a través de Loquo, que nos proporciona una buena información sobre la localización y categoría del anuncio. En estos momentos obtenemos información sobre ofertas de trabajo y compra/venta de inmuebles.

## Feeds de meteorología

Podemos dividir los feeds de meteorología en 2 (meteoclimatic y google), que son utilizados de forma combinada para proporcionar una cobertura global y previsión meteorológica.

Meteoclimatic<sup>9</sup> es una gran red de estaciones meteorológicas automáticas no profesionales en tiempo real y un importante directorio de recursos meteorológicos. Se puede seguir el tiempo y el clima de todos los observatorios del ámbito de cobertura de Meteoclimatic: Península Ibérica, los dos archipiélagos, sur de Francia y la África cercana al Estrecho de Gibraltar. Los datos proporcionados por este sitio (mediante RSS) nos son muy útiles por contener bastante información:

- Temperatura actual
- Temperatura mínima
- Temperatura máxima
- Humedad actual
- Humedad mínima
- Humedad máxima
- Presión atmosférica actual
- Presión atmosférica mínima
- Presión atmosférica máxima
- Velocidad del viento actual
- Velocidad del viento máxima
- Precipitación

Como la información es proporcionada por los usuarios la cobertura es bastante variable, pero es bastante buena en España (especialmente en Cataluña y Comunidad Valenciana) con cobertura de poblaciones con una cifra de habitantes relativamente baja.

**Google Weather** es un API de Google que proporciona información meteorológica de muchas ciudades repartidas por todo el mundo. En España tiene una cobertura menor que Meteoclimatic, por lo que la combinación de ambas fuentes resulta ser una opción muy interesante por disponer información de todo el mundo y una información más exhaustiva en España. Los datos que se pueden obtener son:

- Temperatura actual
- Humedad actual
- Velocidad del viento
- Dirección del viento

Además proporciona información de previsión meteorológica para el día actual y los 3 días siguientes. De cada día tiene datos para:

- Temperatura mínima
- Temperatura máxima
- Previsión (ícono)

---

<sup>9</sup> Meteoclimatic tiene su sitio web en <http://www.meteoclimatic.com>

Nosotros almacenamos toda la información que nos proporciona Google Weather para poder hacer mapas de situación meteorológica actual y previsiones. La situación actual es representada empleando los iconos que nos dan una visión más amena y real del estado del tiempo (no hace falta que intentemos averiguar qué tiempo hace a partir de unas temperaturas determinadas). Como en la capa de meteorología se cargan los iconos de múltiples ciudades lo que hacemos es que cada cierto tiempo actualizamos la base de datos, de forma que podemos dar una respuesta rápida al usuario y no tenemos que hacer 50 consultas de ciudades a Google en cada movimiento.

La información proporcionada por estas fuentes es combinada para proporcionar la **información meteorológica del punto central del mapa**. Constantemente, en la parte superior derecha se le muestra al usuario la información meteorológica disponible del punto más cercano al centro del mapa. Esta combinación se realiza obteniendo la información del punto más cercano con datos de meteoclimatic y los datos del punto más cercano con información de google. Los datos mostrados son los del punto (de los dos) que más cerca se encuentre del centro. Los datos mostrados son:

- Estación meteorológica/ciudad
- Temperatura actual
- Humedad actual

Un ejemplo de salida es la siguiente:

*El tiempo en Madrid-Fuencarral (Madrid) Temp: 21.4ºC Hum: 57.0%*

### Uso del API de Google Weather

El API de Google Weather no está tan documentada como el API para realizar búsquedas en Youtube o geoposicionar. Básicamente consiste en hacer una solicitud a la url

`http://www.google.com/ig/api?weather=<ciudad>&hl=<idioma>`

donde <ciudad> es el nombre de la ciudad y <idioma> el idioma (en, es, fr...) en el que obtener los datos.

La respuesta obtenida es un documento XML con los datos meteorológicos

Por ejemplo

`http://www.google.com/ig/api?weather=Alicante&hl=es`

Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

```
-<xml_api_reply version="1">
  -<weather module_id="0" tab_id="0">
    -<forecast_information>
      <city data="Alicante, Valencia"/>
      <postal_code data="Alicante"/>
      <latitude_e6 data="" />
      <longitude_e6 data="" />
      <forecast_date data="2008-07-02"/>
      <current_date_time data="2008-07-02 15:10:19 +0000"/>
      <unit_system data="SI"/>
    </forecast_information>
    -<current_conditions>
      <condition data="Despejado"/>
      <temp_f data="82"/>
      <temp_c data="28"/>
      <humidity data="Humedad: 57%"/>
      <icon data="/images/weather/sunny.gif"/>
      <wind_condition data="Viento: SE a 27 km/h"/>
    </current_conditions>
    -<forecast_conditions>
      <day_of_week data="Hoy"/>
      <low data="23"/>
      <high data="31"/>
      <icon data="/images/weather/sunny.gif"/>
      <condition data="Despejado"/>
    </forecast_conditions>
    -<forecast_conditions>
      <day_of_week data="jue"/>
      <low data="21"/>
      <high data="31"/>
      <icon data="/images/weather/mostly_sunny.gif"/>
      <condition data="Mayormente soleado"/>
    </forecast_conditions>
    -<forecast_conditions>
      <day_of_week data="vie"/>
      <low data="21"/>
      <high data="29"/>
      <icon data="/images/weather/sunny.gif"/>
      <condition data="Despejado"/>
    </forecast_conditions>
    -<forecast_conditions>
      <day_of_week data="sáb"/>
      <low data="23"/>
      <high data="31"/>
      <icon data="/images/weather/sunny.gif"/>
      <condition data="Despejado"/>
    </forecast_conditions>
  </weather>
</xml_api_reply>
```

### Otros feeds

Hay otras fuentes de información que son no persistentes y no necesitamos mantener en nuestra base de datos. Tal es el caso de **Youtube** cuya información es obtenida de forma muy rápida desde los servidores de Youtube y que, al ser muy variable, es conveniente obtenerla directamente.

Los vídeos de Youtube son obtenidos a través de llamadas a unas direcciones HTML especificadas en el API de Youtube, en las que se especifican una serie de parámetros para definir las distintas opciones. Nosotros hemos usado las librerías en lenguaje Java proporcionadas por Google para interactuar con Youtube realizando búsquedas de vídeos con determinadas palabras clave y ordenación.

La especificación del API puede ser encontrada en la página siguiente.

### Proveedores de puntos

La aplicación web necesita de páginas que proporcionen los datos de los puntos a cargar. Estos puntos son cargados asíncronamente utilizando la técnica denominada AJAX. En cada capa se carga un conjunto de puntos diferente con una serie de atributos (latitud, longitud, contenido...)

Inicialmente estos datos, estructurados utilizando JSON, eran proporcionados desde la aplicación Java, aunque progresivamente fuimos sustituyéndolos por scripts equivalentes en lenguaje PHP y en la actualidad algunos son proporcionados por la aplicación Java y otros por los scripts PHP.

## Búsqueda de videos

En esta sección se explica cómo utilizar el API para recuperar una lista de videos que coincida con un término de búsqueda especificado por el usuario. Para buscar videos, envía una solicitud GET HTTP a la siguiente URL, añadiendo los parámetros de consulta apropiados a tu solicitud.

```
http://gdata.youtube.com/feeds/api/videos
```

El ejemplo siguiente permite buscar el segundo conjunto de 10 videos subidos recientemente que coinciden con el término de consulta "football", pero que no coinciden con el término "soccer".

```
http://gdata.youtube.com/feeds/api/videos?
  vq=football+-soccer
  orderby=published
  start-index=11
  max-results=10
```

Las solicitudes de búsqueda pueden incluir cualquiera de los parámetros siguientes. En primer lugar se muestran los parámetros utilizados con mayor frecuencia.

Nombre	Definición								
vq	El parámetro <code>vq</code> permite especificar un término de consulta para una búsqueda. YouTube buscará todos los metadatos de los videos que coincidan con el término. Los metadatos de los videos incluyen títulos, palabras clave, descripciones, nombres de usuario de los autores y categorías.  Ten en cuenta que a todos los espacios, las comillas y otros signos de puntuación del valor del parámetro se les debe aplicar un formato de escape URL.  Para buscar una frase exacta, escribe la frase entre comillas dobles. Por ejemplo, para buscar videos que coincidan con la frase "spy plane", establece el parámetro <code>vq</code> en "%22spy+plane%22".  Tu solicitud puede utilizar los operadores booleanos NOT (-) y OR (+) para excluir videos o buscar videos asociados a uno de los distintos términos de búsqueda. Por ejemplo, para buscar videos que coincidan con "boating" o "sailing", establece el parámetro <code>vq</code> en "boating%7Csailing". (Ten en cuenta que al carácter de barra vertical se le debe aplicar un formato de escape URL.) De forma similar, para buscar videos que coincidan con "boating" o "sailing" pero no con "fishing", establece el parámetro <code>vq</code> en "boating%7Csailing+!fishing".								
orderby	El parámetro <code>orderby</code> permite especificar el valor que se utilizará para ordenar videos en el conjunto de resultados de la búsqueda. Los valores válidos de este parámetro son <code>relevance</code> , <code>published</code> , <code>viewCount</code> o <code>rating</code> . Además, puedes solicitar los resultados más relevantes para un idioma en particular estableciendo el valor del parámetro <code>relevance_lang_code</code> , donde <code>languageCode</code> es un <a href="#">código de idioma de dos letras ISO 639-1</a> (Utiliza los valores zh-Hans para chino simplificado y zh-Hant para chino tradicional). Además, ten en cuenta que los resultados en otros idiomas se obtendrán si presentan un alto nivel de relevancia con respecto al término de consulta de la búsqueda. El valor predeterminado es <code>relevance</code> .								
start-index	El parámetro <code>start-index</code> permite especificar el índice del primer resultado coincidente que se deberá incluir en el conjunto de resultados. Este parámetro utiliza un índice de base 1, lo que significa que el primer resultado es 1, el segundo es 2 y así sucesivamente. Este parámetro funciona en combinación con el parámetro <code>max-results</code> para determinar qué resultados se ofrecerán. Por ejemplo, para solicitar el segundo conjunto de 25 resultados, es decir, los resultados del 26 al 50, establece el parámetro <code>start-index</code> en 26 y el parámetro <code>max-results</code> en 25.								
max-results	El parámetro <code>max-results</code> especifica el número máximo de resultados que se deben incluir en el conjunto de resultados. Este parámetro funciona en combinación con el parámetro <code>start-index</code> para determinar qué resultados se ofrecerán. Por ejemplo, para solicitar el segundo conjunto de 26 resultados, es decir, los resultados del 26 al 50, establece el parámetro <code>max-results</code> en 26 y el parámetro <code>start-index</code> en 26. El valor predeterminado de este parámetro es 25 y el valor máximo es 50.								
author	El parámetro <code>author</code> restringe la búsqueda a videos subidos por un usuario de YouTube en particular. En la sección <a href="#">Videos subidos por un usuario específico</a> se explica este parámetro de forma más detallada.								
alt	El parámetro <code>alt</code> permite especificar el formato del feed que se devolverá. Los valores válidos de este parámetro son <code>atom</code> , <code>rss</code> , <code>json</code> y <code>json-in-script</code> . El valor predeterminado es <code>atom</code> y en este documento sólo se explica el formato de las respuestas Atom. Para obtener más información sobre el uso de las respuestas del API en JavaScript, consulta <a href="#">Uso de JSON con las API de datos de Google</a> .								
format	El parámetro <code>format</code> permite especificar en qué formato deben estar disponibles los videos. Tu solicitud podrá especificar cualquiera de los formatos siguientes:								
	<table border="1"> <thead> <tr> <th>Valor</th><th>Formato de video</th></tr> </thead> <tbody> <tr> <td>1</td><td>URL de transmisión RTSP para la reproducción de videos de dispositivos móviles. Vídeo H.263 (hasta 176x144) y audio AMR.</td></tr> <tr> <td>5</td><td>URL HTTP en el reproductor (SWF) insertable de este video. Este formato no está disponible para los videos que no se pueden insertar. Los desarrolladores añaden con frecuencia <code>&amp;format=5</code> a sus consultas para restringir los resultados a los videos que se pueden insertar en sus sitios.</td></tr> <tr> <td>6</td><td>URL de transmisión RTSP para la reproducción de videos de dispositivos móviles. Vídeo MPEG-4 SP (hasta 176x144) y audio AAC.</td></tr> </tbody> </table>	Valor	Formato de video	1	URL de transmisión RTSP para la reproducción de videos de dispositivos móviles. Vídeo H.263 (hasta 176x144) y audio AMR.	5	URL HTTP en el reproductor (SWF) insertable de este video. Este formato no está disponible para los videos que no se pueden insertar. Los desarrolladores añaden con frecuencia <code>&amp;format=5</code> a sus consultas para restringir los resultados a los videos que se pueden insertar en sus sitios.	6	URL de transmisión RTSP para la reproducción de videos de dispositivos móviles. Vídeo MPEG-4 SP (hasta 176x144) y audio AAC.
Valor	Formato de video								
1	URL de transmisión RTSP para la reproducción de videos de dispositivos móviles. Vídeo H.263 (hasta 176x144) y audio AMR.								
5	URL HTTP en el reproductor (SWF) insertable de este video. Este formato no está disponible para los videos que no se pueden insertar. Los desarrolladores añaden con frecuencia <code>&amp;format=5</code> a sus consultas para restringir los resultados a los videos que se pueden insertar en sus sitios.								
6	URL de transmisión RTSP para la reproducción de videos de dispositivos móviles. Vídeo MPEG-4 SP (hasta 176x144) y audio AAC.								
lr	El parámetro <code>lr</code> permite restringir la búsqueda de videos que tengan el título, la descripción o las palabras clave en un idioma específico. Los valores válidos del parámetro <code>lr</code> son <a href="#">códigos de idioma de dos letras ISO 639-1</a> . Puedes utilizar los valores zh-Hans para chino simplificado y zh-Hant para chino tradicional. Este parámetro también se puede utilizar para solicitar cualquier feed de videos que no sea un feed estándar.								
racy	El parámetro <code>racy</code> permite incluir contenido restringido y contenido estándar en el conjunto de resultados de una búsqueda. Los valores válidos de este parámetro son <code>include</code> y <code>exclude</code> . De forma predeterminada, se excluirá el contenido restringido. Las entradas de feeds de los videos que incluyen contenido restringido presentan un elemento <code>category</code> adicional.								
restriction	El parámetro <code>restriction</code> identifica la dirección IP que se debe utilizar para filtrar videos que sólo se pueden reproducir en países específicos. De forma predeterminada, el API excluye los videos que no se pueden reproducir en el país desde el que se envían las solicitudes del API. Esta restricción se basa en la dirección IP de la aplicación del cliente.  Para solicitar videos reproducibles desde un equipo específico, incluye el parámetro <code>"restriction"</code> en tu solicitud y establece como valor del parámetro la dirección IP del equipo en el que se reproducirán los videos, por ejemplo, <code>restriction=255.255.255.255</code> .  Para solicitar videos reproducibles desde un país específico, incluye el parámetro <code>"restriction"</code> en tu solicitud y establece como valor del parámetro el código de dos letras ISO 3166 del país en el que se reproducirán los videos, por ejemplo, <code>restriction=DE</code> .								
time	El parámetro <code>time</code> , que sólo está disponible para los feeds estándar <code>top_rated</code> y <code>most_viewed</code> , restringe la búsqueda para videos subidos dentro del período de tiempo especificado. Los valores válidos de este parámetro son <code>today</code> (1 día), <code>this_week</code> (7 días), <code>this_month</code> (1 mes) y <code>all_time</code> . El valor predeterminado de este parámetro es <code>all_time</code> .								

## [Wikipedia](#)

Proporciona información enclopédica sobre los puntos de la región.

La llamada tiene el siguiente formato

```
http://www.sindicacionenmapas.com/feeds/wiki.php?ne=<punto noreste>&sw=<punto suroeste>&c=<numPuntos>&f=W
```

Los datos facilitados son:

- **latitud:** La latitud del punto
- **longitud:** La longitud del punto
- **poblacion:** El número de habitantes
- **nombre:** El nombre del artículo de Wikipedia en el idioma de la interfaz actual
- **nombreES:** El nombre en español del artículo de Wikipedia
- **contenido:** Extracto del contenido del artículo

Ejemplo:

```
http://www.sindicacionenmapas.com/feeds/wiki.php?ne=45.058001,7.470703&sw=38.462192,-13.842773&c=50&f=W
```

```
var points = { p7687:{latitud:40.4166666667,longitude:-3.75,poblacion:45200737,nombre:'España',nombreES:'España',contenido:'España, oficialmente Reino de España, es un país soberano miembro de la Unión Europea, constituido en Estado social y democrático de Derecho, y cuya forma de gobierno es la monarquía parlamentaria. Su territorio, con capital en Madrid, ocupa la mayor parte de la península Ibérica, al que se añaden los archipiélagos de las Islas Baleares, en el mar Mediterráneo occidental, y el de las Islas Canarias, en el océano Atlántico nororiental, así como en el norte del continente africano, las plazas de s...'}, p24068:{latitud:41.6666666667,longitude:2,poblacion:5309404,nombre:'Provincia de Barcelona',nombreES:'Provincia de Barcelona',contenido:'La provincia de Barcelona es una provincia española situada en el nordeste del país, en la comunidad autónoma de Cataluña. Limita con la provincia de Tarragona por el sudoeste, la de Lérida por el noroeste; Gerona por el nordeste y con el mar Mediterráneo por el sudeste. Su capital es Barcelona, donde viven algo menos de una tercera parte de los 5.309.404 habitantes (según INE 2006) de la provincia. La provincia tiene una extensión de 7.733 km². Dado que los límites administrativos comarcas y p...'}, p15863:{latitud:38.7,longitude:-9.1833333333,poblacion:11317192,nombre:'Portugal',nombreES:'Portugal',contenido :'Portugal, oficialmente la República Portuguesa (en portugués: República Portuguesa; pron. IPA ), es un país soberano miembro de la Unión Europea, constituido en democrático de Derecho. Su territorio, con capital en Lisboa, está situado en el sudoeste de Europa, en la Península Ibérica. Limita al este y al norte con España, y al sur y oeste con el océano Atlántico. Comprende también los archipiélagos autónomos de las Azores y de Madeira, situados en el hemisferio norte del océano Atlántico, además...'},....}
```

## Wikimedia Commons

Proporciona imágenes de la región obtenidas a partir del repositorio de Wikimedia Commons.

La llamada tiene el siguiente formato

```
http://www.sindicacionenmapas.com/Sindicacion/puntos.jsp?ne=<punto
noreste>&sw=<punto suroeste>&f=C
```

Los datos facilitados son:

- **latitud:** La latitud de la imagen
- **longitud:** La longitud de la imagen
- **nombre:** El nombre de la imagen en Wikimedia Commons
- **descripcion:** Extracto de la descripción de la imagen
- **cargar:** Indica si debe cargar la imagen o no (sólo se carga una imagen en un punto con múltiples imágenes para evitar sobrecargar el navegador)

Ejemplo:

```
http://www.sindicacionenmapas.com/Sindicacion/puntos.jsp?ne=45.058001,7.47070
3&sw=39.436193,-13.842773&f=C
```

```
var points = {p16177:{latitud:42.9884833333,longitude:-
2.61620555556,nombre:'Image:Vipera seoanei
05.jpg',thumb:'th16177.jpg',url:'http://upload.wikimedia.org/wikipedia/commons/t
humb/b/b6/Vipera_seoanei_05.jpg|800px-Vipera_seoanei_05.jpg',descripcion:'Vipera
seoanei viper, a species living in the northern part of the Iberian peninsula
and southwestern corner of France. This specimen measures about 25 cm long.
Vipera seoanei edo kantabriar sugegorria Iberiar penintsulako iparraldean eta
Frantziako hegomorentzak bizi den sugegorria da. Ale honek 25 cm inguru
neurtzen du. ?????? ??????? (Vipera
seoanei) ? ???, ??????? ? ???????? ?????? ?????????????? ??????????? ? ?????? ??????
???????. ??? ???????? ?????? ?????? 25 ??. {{Translate description',cargar:1}
,p1257:{latitud:41.3703888889,longitude:2.15025,nombre:'Image:Jfader barca
pavillion.jpg',thumb:'th1257.jpg',url:'http://upload.wikimedia.org/wikipedia/com
mons/7/78/Jfader_barca_pavillion.jpg',descripcion:'N?mecký pavilon na
Mezinárodní výstav? v Barcelon?. Postavil jej Ludwig Mies van der Rohe v roce
1929. P?estav?n byl v letech 1983?1989. Der Barcelona-Pavillon, erbaut von
Ludwig Mies van der Rohe 1929 für die Weltausstellung in Barcelona. Der Pavillon
wurde 1930 abgebrochen und 1983?1989 rekonstruiert. The Barcelona Pavilion.
Built by Ludwig Mies van der Rohe in 1929 for the Universal exhibition.
reconstruction 1983?1989 Pavillon de l'\ exposition mondiale à Barcelone,
Espagne. Le pavillon a été...',cargar:1}
,p6904:{latitud:39.5116666667,longitude:-9.1422222222,nombre:'Image:The
Photographer.jpg',thumb:'th6904.jpg',url:'http://upload.wikimedia.org/wikipedia/
commons/thumb/f/f7/The_Photographer.jpg/454px-
The_Photographer.jpg',descripcion:'Old black and white photo: man walking in a
tunnel with a camera. Taken at S. Martinho do Porto, West coast of Portugal. It
is uncertain whether the man is approaching the camera or walking away from it.
Une vieille photo en noir et blanc. Un homme marche dans un passage vouté en
portant un appareil photo. On ne sait pas si l'\homme s\'approche ou s\'éloigne.
Photo prise à S. Martinho do Porto, Côte ouest portugaise. Portekiz\de, S.
Martinho do Porto\de bir tünelin içinde yürümektedir.',cargar:1}}
```

## Meteorología

Proporciona información meteorológica a partir de los datos obtenidos de Google Weather y Meteoclimatic.

La llamada tiene el siguiente formato

```
http://www.sindicacionenmapas.com/Sindicacion/puntos.jsp?ne=<punto
noreste>&sw=<punto suroeste>&f=M3
```

Los datos facilitados son:

- **latitud:** La latitud del punto
- **longitud:** La longitud del punto
- **icono:** La ruta al icono que representa el estado meteorológico
- **temp:** Temperatura en grados centígrados
- **wind:** Velocidad y dirección del viento
- **hum:** Humedad relativa

Ejemplo:

```
http://www.sindicacionenmapas.com/Sindicacion/puntos.jsp?ne=46.769968,11.755
371&sw=36.509636,-9.558105&f=M3
```

```
var points = {m168:{latitud: 45.4637, longitud:
9.1881,nombre:'Milán',icono:'cloudy.gif',temp:24,wind:'0.0 -> E',hum:'56%'},
m306:{latitud: 45.0667,longitude:
7.7000,nombre:'Turín',icono:'cloudy.gif',temp:22,wind:'0.0 -> NE',hum:'65%'},
m185:{latitud: 37.3833,longitude: -
5.9965,nombre:'Sevilla',icono:'sunny.gif',temp:22,wind:'6.0 -> NE',hum:'42%'},
m194:{latitud: 43.6044,longitude:
1.4430,nombre:'Toulouse',icono:'rain.gif',temp:17,wind:'3.0 -> N',hum:'84%'},
m276:{latitud: 39.5695,longitude: 2.6500,nombre:'Palma de
Mallorca',icono:'mostly_cloudy.gif',temp:20,wind:'9.0 -> N',hum:'48%'},
m308:{latitud: 45.4333,longitude:
10.9833,nombre:'Verona',icono:'cloudy.gif',temp:25,wind:'0.0 -> E',hum:'67%'},
m205:{latitud: 36.8402,longitude: -
2.4679,nombre:'Almería',icono:'sunny.gif',temp:21,wind:'0.0 -> NE',hum:'91%'},
m258:{latitud: 45.8285,longitude:
1.2617,nombre:'Limoges',icono:'cloudy.gif',temp:16,wind:'8.0 -> N',hum:'72%'},
m236:{latitud: 41.9818,longitude:
2.8237,nombre:'Gerona',icono:'fog.gif',temp:20,wind:'0.0 -> SW',hum:'98%'},
m247:{latitud: 46.1581,longitude: -1.1536,nombre:'La
Rochelle',icono:,temp:17,wind:'0.0 -> N',hum:'68%'}, m210:{latitud:
43.5807,longitude:
7.1209,nombre:'Antibes',icono:'mostly_cloudy.gif',temp:22,wind:'11.0 ->
N',hum:'78%'}, m123:{latitud: 41.9186,longitude:
8.7369,nombre:'Ajaccio',icono:'sunny.gif',temp:18,wind:'11.0 -> NE',hum:'83%'},
m265:{latitud: 44.0176,longitude:
1.3589,nombre:'Montauban',icono:'rain.gif',temp:15,wind:'4.0 -> NW',hum:'88%'},
m218:{latitud: 45.1586,longitude: 1.5326,nombre:'Brive-la-
Gaillarde',icono:'rain.gif',temp:15,wind:'11.0 -> NW',hum:'94%'}, m303:{latitud:
43.2339,longitude: 0.0753,nombre:'Tarbes',icono:'rain.gif',temp:13,wind:'12.0 ->
NW',hum:'100%'}, m50:{latitud: 41.3879,longitude:
2.1699,nombre:'Barcelona',icono:'mostly_cloudy.gif',temp:23,wind:'1.0 ->
W',hum:'82%'}...}
```

### **Meteorología (punto central del mapa)**

Proporciona información meteorológica con los datos meteorológicos de la población más cercana a las coordenadas proporcionadas. Combina Google Weather y Meteoclimatic para poder conseguir los datos del punto más cercano.

La llamada tiene el siguiente formato

```
http://www.sindicacionenmapas.com/Sindicacion/meteo.jsp?lat=<latitud>&lon=<longitude>
```

Los datos facilitados son:

- **nombre:** El nombre del punto
- **icono:** La ruta al icono que representa el estado meteorológico
- **temp:** Temperatura en grados centígrados
- **hum:** Humedad relativa

Ejemplo:

```
http://www.sindicacionenmapas.com/Sindicacion/meteo.jsp?lat=38.292636837348866&lon=-0.5548095703125001
```

```
var pmeteo =  
{0:{nombre:'Alicante',temp:27,hum:'49',icon:'/images/weather/sunny.gif'}}}
```

## Youtube

Proporciona vídeos de Youtube de la región especificada, ordenados según se establezca.

La llamada tiene el siguiente formato

```
http://www.sindicacionenmapas.com/Sindicacion/puntos.jsp?ne=<punto noreste>&sw=<punto suroeste>&f=Y&order=<ordenación>
```

donde el campo de ordenación puede ser:

- **fecha**
- **puntuacion**
- **relevancia**
- **vistas**

Los datos proporcionados en la cadena de respuesta son, por una parte, un array con los nombres de las ciudades o puntos de información de dicha región sobre los que se han buscado los vídeos y, por otra, una lista JSON con los siguientes campos:

- **titulo**: El título del vídeo
- **desc**: Descripción del vídeo
- **fecha**: Fecha del vídeo
- **thumb**: URL de la imagen de miniatura
- **enlace**: URL del vídeo en Youtube

Ejemplo:

```
http://www.sindicacionenmapas.com/Sindicacion/puntos.jsp?ne=38.561053,-0.539017&sw=38.427505,-0.87204&f=Y&order=fecha
```

```
ciudades= new Array ("Agost","Elda","Petrel","Monóvar","Sax"); var points = {yLB1CdP9QZ50:{titulo:'Encuentran abandonado un canguro en un campo de Elda (Alicante)',desc:'Villena (Alicante), 27 feb (EFE-TV).- Un canguro, aparentemente abandonado, ha sido encontrado en un campo de Elda en un estado deficiente de salud, ha informado la Sociedad Protectora de Animales de ...',thumb:'http://img.youtube.com/vi/LB1CdP9QZ50/2.jpg',fecha:'2008-02-27T02:43:13.000-08:00',thumb:'http://img.youtube.com/vi/LB1CdP9QZ50/2.jpg',enlace:'http://www.youtube.com/v/LB1CdP9QZ50'}, y1lSd_WOLJN8:{titulo:'Full Elda',desc:'competicion full en elda',thumb:'http://img.youtube.com/vi/1lSd_WOLJN8/2.jpg',fecha:'2008-07-05T18:09:20.000-07:00',thumb:'http://img.youtube.com/vi/1lSd_WOLJN8/2.jpg',enlace:'http://www.youtube.com/v/1lSd_WOLJN8'}, yWbJ4kst0dQg:{titulo:'COMO LLEGAR A SCOOTER HONDA',desc:'Estamos en Valencia: PATERNA-POLIGONO FUNETE DEL JARROEn la calle Ciudad de Elda, 7b',thumb:'http://img.youtube.com/vi/WbJ4kst0dQg/2.jpg',fecha:'2008-07-05T14:35:58.000-07:00',thumb:'http://img.youtube.com/vi/WbJ4kst0dQg/2.jpg',enlace:'http://www.youtube.com/v/WbJ4kst0dQg'}, y2ceJTp_Eofg:{titulo:'DÍA DE AURE \'08',desc:'Celebración en Villajoyosa de la jornada más importante del Club Arrecife de Elda: el día de Aure.',thumb:'http://img.youtube.com/vi/2ceJTp-Eofg/2.jpg',fecha:'2008-07-03T23:25:17.000-07:00',thumb:'http://img.youtube.com/vi/2ceJTp-Eofg/2.jpg',enlace:'http://www.youtube.com/v/2ceJTp-Eofg'}}}
```

## Anuncios clasificados

Proporciona anuncios clasificados obtenidos a partir de Loquo.

La llamada tiene el siguiente formato

```
http://www.sindicacionenmapas.com/Sindicacion/puntos.jsp?ne=<punto
noreste>&sw=<punto suroeste>&f=A
```

Los datos proporcionados en la cadena de respuesta son:

- **latitud:** La latitud del punto donde se ha determinado que está situado el anuncio
- **longitud:** La longitud del punto donde se ha determinado que está situado el anuncio
- **título:** El título del anuncio clasificado
- **enlace:** El enlace a la web donde está el anuncio clasificado
- **contenido:** El contenido del anuncio clasificado

Ejemplo:

```
http://www.sindicacionenmapas.com/Sindicacion/puntos.jsp?ne=38.561053,-
0.539017&sw=38.458697,-0.87204&f=A
```

```
var points = {36452:{latitud:38.475334,longitud:-0.787103,titulo:'120.000 EUR
-PLATA BAJA EN LA
NUCIA',enlace:'http://alicante.loquo.com/es\_es/post/321486/plata-baja-en-la-nucia',contenido:'SE VENDE PLANTA BAJA EN URBANIZACION PINAR DE GARAITA, EDIFICIO
LA PINADA EN LA CALLE MURILLO. COMEDOR, COCINA AMUEBLADA, BAÑO Y DOS
HABITACIONES (UNA MAYOR Y OTRA DE MENOR...'), 36131:{latitud:38.49025,longitud:-
0.799942,titulo:'288.000 EUR -VENDO PISO ZONA CENTRO - PLAZA
LUCEROS*foto',enlace:'http://alicante.loquo.com/es\_es/post/320190/vendo-piso-zona-centro-plaza-luceros',contenido:'Vendo piso en el centro de Alicante, zona
Plaza de los Luceros. Se trata de una quinta planta con ascensor. El piso es muy
luminoso, con cuatro habitaciones, sala, amplia cocina totalmente...'},
32541:{latitud:38.478,longitud:-0.790464,titulo:'Venta piso en
Elda(Centro)*foto',enlace:'http://alicante.loquo.com/es\_es/post/312095/venta-piso-en-elda',contenido:'Se vende piso moderno, por cambio de localidad. Ubicado
en pleno centro de Elda. 105m. Todo exterior, muy luminoso. 4º piso con ascensor
recien reformado y amueblado, sin estrenar. Posee:...Mapa:Plaza Sagasta Elda'},
29453:{latitud:38.4902,longitud:-0.799942,titulo:'PON TU MI PRECIO!!!! QUE SEA
RAZONABLE, CLARO;;(ALICANTE
CENTRO)',enlace:'http://alicante.loquo.com/es\_es/post/206319/pon-tu-mi-precio-que-sea-razonable-claro',contenido:'POR TRASLADO PROFESIONAL DE MI DUEÑO A OTRA
PROVINCIA ME VENDEN. TENGO 90M2, ESTOY EN EL CENTRO DE ALICANTE, A DOS MINUTOS
DE LA PLAZA DE LUCEROS. ME ACABAN DE REFORMAR (LA REFORMA SE...'),
29454:{latitud:38.4902,longitud:-0.799942,titulo:'PON TU MI PRECIO!!!! QUE SEA
RAZONABLE, CLARO;;(ALICANTE
CENTRO)',enlace:'http://alicante.loquo.com/es\_es/post/206319/pon-tu-mi-precio-que-sea-razonable-claro',contenido:'POR TRASLADO PROFESIONAL DE MI DUEÑO A OTRA
PROVINCIA ME VENDEN. TENGO 90M2, ESTOY EN EL CENTRO DE ALICANTE, A DOS MINUTOS
DE LA PLAZA DE LUCEROS. ME ACABAN DE REFORMAR (LA REFORMA SE...'),
25659:{latitud:38.5371,longitud:-0.817786,titulo:'132.222 EUR -VENDO PISO EN SA
JOSÉ, MUY LUMINOSO, REFORMADO Y CON GARAJE(SAN JOSÉ -
SALAMANCA)*foto',enlace:'http://salamanca.loquo.com/es\_es/post/51119/vendo-piso-en-sa-jose-muy-luminoso-reformado-y-con-garaje',contenido:'Piso de tres
dormitorios, salón-comedor con acceso a balcón cerrado, un cuarto de baño
amueblado con ducha, cocina amueblada con acceso a una galeria y garaje con
acceso...Mapa:C/Maestro Vives, 1-3, 2ºIzq. Salamanca'}}}
```

## Puntos propios

Proporciona los puntos que han subido los usuarios a través del sitio web.

La llamada tiene el siguiente formato

```
http://www.sindicacionenmapas.com/feeds/puntopropio.php?ne=<punto noreste>&sw=<punto suroeste>
```

Los datos proporcionados en la cadena de respuesta son:

- **publico:** Indica si el punto es público (1) o privado (0)
- **latitud:** La latitud del punto donde se ha posicionado el contenido
- **longitud:** La longitud del punto donde se ha posicionado el contenido
- **nombre:** El nombre del punto propio
- **contenido:** El contenido del punto propio
- **url:** El enlace a la imagen
- **urlPromo:** El enlace a una web que pueda haber establecido el usuario al crear el punto propio
- **thumb:** Nombre del archivo generado en el servidor que contiene la miniatura.
- **tam:** Ancho / alto de la imagen a mostrar en el mapa, determinada basándonos en el número de visitas de cada punto propio

Ejemplo:

```
http://www.sindicacionenmapas.com/feeds/puntopropio.php?ne=38.560516,-0.609055&sw=38.458159,-0.942078
```

```
var points = {
p15:{publico:1,usuario:'Jarke',latitud:38.465568797775,longitud:-0.808546543121338,nombre:'Drink Team',contenido:'Drink Team en el CEE. Tricampeones.',url:'http://img364.imageshack.us/img364/5911/img0035008.jpg',urlPromo:,thumb:'th15.jpg',tam:'25'},
p25:{publico:1,usuario:'Jarke',latitud:38.4984324103761,longitud:-0.700684189796448,nombre:'Comida en Rabosa',contenido:'Comiendo una gachamiga en Rabosa, noviembre de 2007.',url:'http://img101.imageshack.us/img101/9876/rabosa2er5.jpg',urlPromo:,thumb:'th25.jpg',tam:'28'},
p97:{publico:1,usuario:'jrafaelnavarro',latitud:38.477798869174,longitud:-0.7965087890625,nombre:'Calle Nueva (Elda)',contenido:,url:'http://img387.imageshack.us/img387/7675/06112007012za0.jpg',urlPromo:,thumb:'th97.jpg',tam:'25'},
p98:{publico:1,usuario:'jrafaelnavarro',latitud:38.4668120590865,longitud:-0.842063426971436,nombre:'Ermita de las Cañas (Elda)',contenido:,url:'http://img367.imageshack.us/img367/4923/hpim7157pv1.jpg',urlPromo:'http://www.alicantevivo.org/2008/05/finca-lacy-y-las-caadas.html',thumb:'th98.jpg',tam:'25'},
p177:{publico:1,usuario:,latitud:38.4802681510875,longitud:-0.69587230682373,nombre:'Rincón Bello',contenido:'Puente de madera en Rincón Bello, Petrel.',url:'http://img403.imageshack.us/img403/7711/dsc01503sa6.jpg',urlPromo:'http://www.espacioblog.com/peterparker/post/2007/12/31/de-petrel-rincain-bello',thumb:'th177.jpg',tam:'30'}}}
```

## Categorizador de textos

Hemos implementado un categorizador de noticias (aunque es aplicable a cualquier texto) que dada una noticia de entrada la clasifica en una de 8 categorías. Las categorías disponibles son:

- Deportes
- Salud
- Cultura
- Tecnología
- Economía
- Medio ambiente
- Política
- Sucesos

El motivo por el que realizamos este categorizador es, por un lado, poner en práctica el procesamiento del lenguaje natural aplicando técnicas de clasificación y, por otro, establecer un método flexible de clasificar noticias sin importar de qué feed provengan ni del tipo de periódico o fuente. De esta forma se pueden añadir a la aplicación RSS de noticias diversas (o en un desarrollo futuro permitir que sea el usuario quien especifique qué feeds de noticias desea visualizar) y que el sistema sea capaz de clasificar las noticias, facilitando la navegación entre las mismas al ser divididas en categorías accesibles desde la interfaz.

Además considerábamos que una noticia podía pertenecer a varias categorías simultáneamente y que, si nos ceñíamos a la categorización ofrecida en origen, podían aparecer noticias similares en varias categorías (por ejemplo, la oferta de compra de Microsoft a Yahoo! aparecía tanto en secciones de tecnología como de economía).

### Noticias

Desde el principio del desarrollo del proyecto fuimos obteniendo noticias de diversos medios, en especial de aquellos cuyos feeds pertenecieran a una categoría determinada. Así, preferíamos los feeds de *fútbol* o *baloncesto* a los de *deportes* y los feeds de *Estados Unidos* a los de *Internacional*. Una vez tuvimos suficientes noticias comenzamos la división en categorías. Inicialmente, en el clasificador realizado en diciembre de 2007, había un total de 11 categorías, que comprendían lugares geográficos (Asia, América Latina, Estados Unidos, etc.) y categorías como ciencias, salud o cultura).

### Categorías inadecuadas

Los resultados de clasificación de nuevas noticias usando técnicas bayesianas proporcionaban un acierto de entre el 50% y el 60%. La mayor parte de errores se debían a que las categorías *Nacional* e *Internacional* no definían adecuadamente el contexto de la noticia, pues hacían referencia a la localización, mientras el resto hacía referencia al tema sobre el cual se estaba hablando. Además, el carácter internacional del proyecto hacía que no tuviera mucho sentido hablar de *nacional* e *internacional* pues lo que era nacional en España es internacional en Argentina y viceversa.

Así, decidimos eliminar las categorías nacional e internacional para que la categorización no tuviera en cuenta el lugar de origen de la noticia ni el lugar al que se refería (para eso ya disponemos del módulo de geolocalización de textos). Además incluimos la categoría de Medio Ambiente, un tema de creciente interés en los últimos meses y cuyas noticias no acababan de encuadrarse adecuadamente en salud ni en tecnología.

En las pruebas de clasificación periódicas que realizamos, para conjuntos de noticias nuevas cuya sección conocíamos por el feed, la categorización con nuestro sistema lograba unos porcentajes de acierto sobre el 90% (teniendo en cuenta sólo la categoría más factible dada como salida en nuestro categorizador de múltiples categorías).

### Mejorando el sistema

#### *Categorías definitivas*

Una vez que el sistema estaba en marcha en el sitio web, y una vez añadidos feeds de noticias de periódicos regionales (por ejemplo de Albacete, Alicante y Murcia) para ver qué tal se comportaba el sistema, observamos cómo había noticias que no se clasificaban correctamente. Esto es porque las pruebas la habíamos realizado con noticias cuya categoría sabíamos que se encontraba entre las que nuestro categorizador reconocía, esto es, con las categorías a las que pertenecían las noticias del conjunto de entrenamiento.

Estas noticias solían hablar de accidentes de tráfico, asesinatos y demás sucesos, por lo que decidimos ampliar el categorizador con una categoría de "Sucesos" para cubrir este tipo de noticias.

#### *Internacionalización de la capa de noticias*

Por último, con la internacionalización de la interfaz de Sindicación en Mapas, pensamos que era una buena idea introducir noticias en otros idiomas (inglés, francés, italiano y catalán). Para aprovechar la técnica de clasificación ya implementada junto con los datos de noticias en castellano obtenidos durante varios meses decidimos que era más sencillo traducir las noticias que realizar un categorizador para cada idioma.

Para ello hemos utilizado el API de Google para traducir textos<sup>10</sup> que se puede utilizar fácilmente empleando las librerías que proporciona para Java. Para cada feed sabemos el idioma de las noticias de dicho feed (aunque usando dicho API podría detectarse también el idioma). Cada noticia del feed es traducida al español para poder ser categorizada, pero es almacenada finalmente en la base de datos en su idioma original para que pueda ser consultada a través de la web.

### *Técnica de clasificación*

Contamos con cerca de 50.000 noticias obtenidas desde el 4 de diciembre de 2007 recorriendo periódicamente un conjunto de feeds de origen de diversos periódicos. Hemos probado varias técnicas para realizar la clasificación:

- **Extracción de características:** Se extraen las palabras que permitan un error mínimo en la clasificación. Esta técnica tomaba mucho tiempo y, aunque reducía la dimensionalidad del problema, no aprovechábamos palabras con pocas apariciones que podían resultar determinantes para la clasificación.
- **Clasificador conceptual:** Al añadir la categoría de Medio Ambiente y no disponer de noticias previamente clasificadas con esa categoría, pensamos que podíamos definir categorías usando palabras clave y, a partir de esas palabras, generar una lista con las palabras que aparecen más frecuentemente junto a esas palabras en las noticias. De esta forma se podían definir categorías dinámicamente e incluso el usuario podría definirlas. Por ejemplo, si un usuario añadiera sus propios feeds y le interesara la informática, sus noticias serían categorizadas probablemente como

---

<sup>10</sup> Se explica su uso en <http://www.tufucion.com/ajax-language-api>

Tecnología. Sin embargo, el usuario podría definir las categorías "software libre", "gadgets" y "juegos" y establecer palabras relacionadas con esas categorías ('linux', 'iphone' y 'fifa' respectivamente). Probamos este clasificador con nombres propios y funcionaba relativamente bien, pero era muy sensible a las palabras clave encontradas y en cuanto aparecían palabras como nombres de ciudad el conjunto de palabras comenzaba a incluir términos poco relacionados con los buscados.

- **Clasificador por frecuencia de aparición:** Éste es el método que hemos empleado. El procedimiento seguido para implementar el clasificador es el siguiente:
  1. A partir de una base de datos de noticias extraemos las palabras (tanto del titular como del contenido)
  2. Eliminamos las palabras comunes que no aportan información sobre categoría (*stop words*)
  3. Aplicamos un stemmer
  4. Exportamos el contenido a un archivo

## Aprendizaje y preprocesamiento

### Extracción de palabras

En primer lugar extraemos las palabras. Hemos dividido los textos por los espacios en blanco y signos de puntuación y hemos eliminado las palabras que contuvieran números.

### Eliminación de stop words

Las *stop words* o palabras vacías es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc. que son filtradas antes o después del procesamiento de datos en lenguaje natural (texto). A Hans Peter Luhn, uno de los pioneros en recuperación de información, se le atribuye la acuñación de la locución inglesa *stop words* y el uso del concepto en su diseño. Está controlada por introducción humana y no automática.

No hay una lista definitiva de palabras vacías que todas las herramientas de procesamiento de lenguajes naturales incorporen. No todas las herramientas de PLN usan una lista de palabras vacías. Algunas herramientas evitan usarlo específicamente para soportar búsquedas por frase.

Se pueden encontrar listas de palabras vacías en español en Internet<sup>11</sup>. Algunas de ellas son:

---

<sup>11</sup> El extracto de palabras ha sido obtenido de la lista disponible en <http://reina.usal.es/utiles/vacias.txt>

a	algunos	aquellos	catorce
aca	alla	aqui	cerca
ademas	alli	asi	cien
ahi	ante	aun	ciento
ahora	antes	aunque	cientos
al	aparte	bajo	cierto
algo	apenas	bastante	cinco
algun	aqueل	bien	cincuenta
alguna	aquella	bueno	como
algunas	aqueellas	cabe	con
alguno	aqueلlo	casi	

## Stemming

Stemming es un método para reducir una palabra a su raíz o mejor a un stem o tema. Hay algunos algoritmos de stemming que ayudan en sistemas de recuperación de información. Stemming aumenta el recall que es una medida sobre el número de documentos que se pueden encontrar con una consulta. Por ejemplo una consulta sobre "bibliotecas" también encuentra documentos en los que solo aparezca "bibliotecario" porque el stem de las dos palabras es el mismo ("bibliotec").

Nosotros hemos empleado Snowball<sup>12</sup>, que es un pequeño lenguaje de programación para el manejo de cadenas que permite implementar fácilmente algoritmos de stemming. Se puede generar código en ANSI C y Java. Las páginas de Snowball contienen stemmers para 12 idiomas (incluido el castellano).

## Tabla de frecuencias

En una tabla hash almacenamos para cada raíz el número de apariciones en cada categoría de noticia. Nuestra solución tiene en cuenta únicamente los unigramas, aunque está preparada para soportar bigramas. Los bigramas son conjuntos de 2 palabras, lo que significa que dividimos el texto en todas las posibles parejas de 2 palabras sucesivas. Esto mejoraba ligeramente el error de clasificación pero incrementaba mucho el tiempo requerido para llevarla a cabo dado que aparecían muchas combinaciones de palabras y la tabla crecía considerablemente.

El tamaño de la tabla puede ser reducido usando alguna de estas opciones:

- **Eliminar n-gramas con pocas apariciones:** Pueden ser debidas a errores ortográficos o ser tan poco frecuentes que no nos estén aportando información correcta sobre la categoría en la que esa palabra aparecerá más frecuentemente.
- **Eliminar n-gramas que no aporten contenido de categoría:** Esto se podría determinar comprobando qué palabras están presentes en una proporción similar en todas las categorías. De hecho, una *stop word* debería ser eliminada en este paso, aunque nosotros lo hagamos previamente debido a que a priori sabemos qué conjunto de palabras es relativamente frecuente y no aportan nada. La ventaja de este método es que se puede adaptar a distintos contextos y no es necesario establecer una lista inicial de palabras a eliminar.

Hemos considerado las posibles reducciones en el proceso de clasificación para no tener que generar las frecuencias con cada posible mejora o cambio en el algoritmo.

---

<sup>12</sup> Snowball está disponible en <http://snowball.tartarus.org>

## Exportación

Hemos exportado los datos a un archivo de texto plano, que es cargado cuando se activa el módulo de actualización de feeds para luego realizar la clasificación de las noticias y para ser probado desde el módulo de prueba del sitio web.

El formato del archivo es:

```
raíz:nDeportes:nSalud:nCultura:nTecnología:nEconomía:nMedioAmbiente:nPolítica:nS
ucesos
```

Un extracto del archivo es el siguiente:

aa:0:0:1:0:1:0:0:0	abarc:0:2:2:1:1:2:0:0
aadpc:0:0:3:0:0:0:0:0:0	abarrot:1:0:1:0:1:0:0:0:0
aai:0:0:0:0:1:0:0:0:0	abas:0:0:0:1:0:0:0:0:0
aarass:0:0:0:0:0:0:2:0	abascal:0:0:2:0:0:0:0:0:0
aarhus:3:0:0:0:0:0:0:0:0	abast:0:0:0:2:0:0:0:0:0
aaron:3:0:1:0:0:0:0:0:0	abastec:0:0:0:1:7:11:1:0
aat:1:0:0:0:0:0:0:0:0	abat:0:0:5:0:1:2:0:0:0
ab:0:0:0:1:0:0:0:0:0	abba:0:0:6:0:0:0:0:0:0
abad:0:0:3:0:0:0:0:0:0	abbad:0:0:1:0:0:0:0:0:0
abadiñ:0:0:0:0:0:0:1:0	abbey:0:0:0:0:1:0:0:0:0
abaj:8:0:3:0:4:0:0:0:0	abbiati:2:0:0:0:0:0:0:0:0
abakanowicz:0:0:1:0:0:0:0:0:0	abc:43:0:55:25:5:0:0:0:0
abam:1:0:0:0:0:0:0:0:0	abcd:0:0:4:0:0:0:0:0:0
aban:0:0:1:0:0:0:0:0:0	abdal:0:0:1:0:1:0:0:0:0
abander:1:0:0:0:0:0:0:0:0	abdall:0:0:0:0:1:0:0:0:0
abandon:67:3:13:34:47:14:12:2	abdelatif:0:0:3:0:0:0:0:0:0
abandons:1:0:0:0:0:0:0:0:0	abdelil:0:0:0:0:0:0:0:2:0
abarat:0:1:1:2:15:0:0:0:0	

El archivo incluye casi 30.000 raíces diferentes con las frecuencias de aparición en cada una de las 8 categorías de noticias.

El número de noticias se almacena en el archivo cantidades.txt, cuyo contenido tiene un aspecto como el siguiente:

```
5199:1318:3798:2776:4971:688:1158:149
```

donde las cifras se corresponden con las siguientes categorías:

```
Deportes:Salud:Cultura:Tecnología:Economía:MedioAmbiente:Política:Sucesos
```

## Clasificador

La clasificación de una noticia se realiza eliminando las *stop words*, realizando un *stemming* y contabilizando el número de apariciones de cada palabra en cada categoría. Como en el archivo de texto están almacenadas el valor absoluto de número de apariciones, esta cifra debe ser dividida entre el número de noticias de dicha categoría (suponemos que la longitud de cada noticia es similar independientemente de la categoría).

Así, para la raíz *abandon*, de la que teníamos los siguientes datos:

abandon:67:3:13:34:47:14:12:2

no tiene por qué ser más probable en la categoría de Deportes (67 apariciones en 5.199 noticias) que, por ejemplo, en Medio ambiente (14 apariciones en 688 noticias), dado que los valores son apariciones absolutas y usamos el número de noticias para extraer la proporción.

Cada unígrafo, a la vez es ponderado de forma que se divide el número de apariciones en su categoría entre el número de apariciones (normalizada) en el resto de categorías. Así, un n-grama tiene puntuación máxima si no aparece en ninguna otra categoría, y va disminuyendo conforme aumenta su presencia en el resto de categorías (lo que implica, de alguna forma, que esa palabra no sirve para determinar la categoría de la noticia). En pruebas prácticas mejoraba el acierto en 7 puntos aproximadamente (de un 73 a un 80% de acierto).

Así, podemos sacar como conclusión que la *puntuación* de un n-grama para pertenecer a una categoría dada depende de:

1. El número de apariciones en las noticias de dicha categoría
2. El número de noticias usadas para el aprendizaje perteneciente a dicha categoría
3. El número de apariciones en las noticias de otras categorías

### Podas

Las posibles reducciones de la tabla de frecuencias han sido consideradas en el proceso de clasificación (eliminar palabras con pocas apariciones o eliminar palabras que no ayuden a discernir a qué categoría pertenecen).

Basándonos en pruebas prácticas, la primera de las posibles reducciones hacia que el error de clasificación creciera, así que decidimos mantener todas las palabras extraídas. La segunda de las reducciones sí la hemos llevado a la práctica. La función `Historico.tenerEnCuenta()` se encarga de determinar si se debe tener o no en cuenta el elemento actual en base al número de apariciones en cada categoría de noticias. Si la división entre el máximo y el mínimo no supera un umbral, no es tenido en cuenta. El umbral está establecido a 12, que proporciona un buen resultado, descartando aproximadamente un 16% de unigramas.

Además hemos realizado otra poda, de forma que descartamos, para una noticia dada, las categorías que no alcanzan una probabilidad mínima (calculada en base al número de n-gramas de la noticia y la media del valor otorgado a cada uno de ellos para dicha categoría).

### Resultados

El categorizador de noticias se comporta bastante bien. Basta con echar un vistazo a la capa de noticias del sitio web. Está especialmente indicado para noticias, pues ha sido con este tipo de textos con los que ha sido entrenado. La salida del clasificador es un conjunto de categorías probables, ordenado de mayor a menor probabilidad. Esto permite que una noticia pueda ser clasificada en varias categorías.

Tal es el caso de la siguiente captura realizada el 24 de junio:

**Noticias de la zona****24/06/2008 - Cantabria****Los elevados precios de la energía preocupa al 78% de las empresas cántabras**

Un 78,2% de las empresas cántabras considera que su principal problema energético es el elevado precio que se paga por la energía que consume, según constata un estudio elaborado por la Cámara de Come...

**24/06/2008 - España****Zapatero destaca el 'sorpasso' de España a Italia en economía y fútbol**

El presidente destacó ayer cómo España, tras superar hace unos meses a Italia en renta per cápita, dio el 'sorpasso' el domingo a ese país «amigo y querido» en un campo de fútbol. Zapatero hizo esta r...

**24/06/2008 - Murcia****Lamata pide a Cospedal que ratifique su postura sobre el trasvase**

El vicepresidente de Castilla-La Mancha, Fernando Lamata, dijo ayer que le «tranquilizaría» que la nueva secretaria general del PP, María Dolores de Cospedal, se ratificara en la defensa de la caducid...

Se puede apreciar cómo asigna a la noticia central varias categorías (economía y deportes en este caso).

## Extracción de origen

### Extracción de ciudad

Uno de los módulos esenciales del proyecto es el que se encarga de la extracción del origen a partir de un texto dado. Esta función extrae las ciudades que encuentra en un texto, de tal forma que pueda ser geoposicionado sin necesidad de disponer de información a priori sobre la situación geográfica en la que se pueda encontrar. Las características principales de este módulo son:

- **Definición de palabras previas a un punto de información:** Hemos definido unas palabras que preceden a la probable población. De esta forma conseguimos, por un lado, reducir el número de cadenas candidatas a ser poblaciones (que serían todas las que comenzaran por mayúscula) y por otro, evitar que un nombre propio pueda ser considerado una población.

Tomando como ejemplo varios extractos de noticias, vemos como en éstas tiene sentido posicionar la noticia en Elda:

*La alcaldesa **de Elda**, Adela Pedrosa, ha anunciado la supresión definitiva de la ORA en toda la ciudad. La medida se ha tomado con el apoyo de los...*

*La supresión de la zona azul ha provocado la división de comerciantes y vecinos **en Elda**. Gran parte de los afectados han acogido con...*

Pero no ocurre lo mismo con las siguientes:

*El juramento deportivo estuvo a cargo del jugador José Luis Lemus Borque, mientras que la patada inaugural la realizó SGM **Elda** María...*

*En el interior del vehículo compacto viajaban Inocencio Flores López Pérez, de 60 años; **Elda** Gómez Castellanos de 51 y una menor de edad, de quien no se...*

Palabras como **en** o **de** son las que usamos para delimitar las posibles ciudades. En la expresión regular utilizada para buscar las ciudades se buscan estas palabras seguidas de una que empiece por mayúsculas y cogemos como candidata dicha palabra que probablemente se corresponde con un nombre propio. Las palabras que empleamos (lo que llamamos *cadenasOrigen*), por orden de comprobación, son:

- "en"
- "Universidad de"
- "Ayuntamiento de"
- "alcalde de"
- "ciudad de"
- "localidad de"
- "municipio de"
- "Aeropuerto de"
- "de"
- **Concatenación del nombre:** A partir de ahí usamos otro conjunto de palabras que nos sirven para ir completando el nombre de la población. Así, **San Juan de Alicante** no se quedaría sólo en **San** (debemos continuar porque hay una palabra a continuación que también empieza por mayúsucula) ni **San Juan** (seguimos concatenando porque **de** es una de las palabras que consideramos que

forma parte del nombre de la ciudad una vez se ha comenzado a formar el nombre). El conjunto de palabras (*palabrasCiudades*) que hemos empleado es:

- "de"
- "del"
- "el"
- "la"
- "las"
- "los"

- **Búsqueda de la cadena en la base de datos de Wikipedia:** La cadena candidata a corresponder con un punto geográfico es buscada en la tabla de Wikipedia para comprobar si existe y asignar entonces el nombre del artículo, que nos servirá para posteriormente utilizar sus coordenadas geográficas.

#### Otros aspectos tenidos en cuenta

- **Nombre de la población al principio de introducción o contenido seguido de símbolo:** En algunas noticias, dependiendo normalmente del periódico, se incluye al principio de la misma (bien en el titular o bien en el contenido) información sobre el origen de la noticia. Este dato no se podía conseguir utilizando la técnica explicada anteriormente, por lo que, cuando nos llega una noticia, primero se busca un conjunto de palabras hasta el primer símbolo que no sea letra, y si el conjunto tiene una cantidad de palabras de 5 ó menos, entonces se procede a comprobar en la base de datos de Wikipedia si se corresponde con un punto de información. Esto se hace tanto para el titular como para el contenido. Así, para noticias como las siguientes:

**BUENOS AIRES** -- *Libertad de Sunchales apabulló a Belgrano de San Nicolás al vencerlo por 94-56 en la...*

**QUITO** -- *El argentino Martín Mandra le otorgó el sábado la victoria por 1-0...*

obtendríamos correctamente el origen.

- **Nombre de la población en otro idioma (catalán, inglés...):** Dado que en Wikipedia en español las poblaciones españolas tienen el nombre en castellano (cuando éste existe) y en muchas ocasiones éste no es nombre más utilizado (por ejemplo *Alcira* y *Alzira*), en la búsqueda que realizamos para comprobar si una cadena se corresponde con un punto de información no sólo buscamos si coincide con el nombre de un artículo de la Wikipedia en español, sino que también lo comprobamos con la Wikipedia en inglés, dado que tiene una gran cobertura y suele respetar el nombre oficial de las poblaciones. Esto permite que un mayor número de noticias sean geoposicionadas.

#### Efectividad

A fecha de 25 de junio de 2008 teníamos en nuestra base de datos un total de 50.398 noticias, de las que 17.034 tienen asignado el nombre de un artículo de Wikipedia con coordenadas geográficas, lo que representa un 33,8%. Hay que pensar que, por un lado, no todas las noticias contienen información acerca del lugar geográfico (puede no importar, darse por sabido o que sea incluido en el cuerpo de la noticia definitiva y no en el resumen que obtenemos a través de RSS) o puede que ese lugar no tenga artículo en Wikipedia o bien dicho artículo no tenga asignadas coordenadas geográficas (en ambos casos dicho artículo no estaría en nuestra base de datos).

En total tenemos noticias de unos 1.474 lugares. Estos lugares dependen directamente de los feeds empleados y del ámbito de los periódicos (no es lo mismo un periódico nacional que uno comarcal). Los puntos que tienen un mayor número de noticias geoposicionadas son Madrid (1.069 noticias), España (803), Albacete (773), Barcelona (510), Estados Unidos (406), Buenos Aires (337), Gaza (241) y Alicante (237). El gran número de noticias de Albacete y Alicante responden a la inclusión en el listado de feeds empleados los periódicos La Verdad de Albacete y Diario Información de Alicante.

### **Extracción de dirección**

También hemos implementado la extracción de información *a nivel de calle* para poder posicionar mejor los anuncios clasificados. Este módulo busca en el texto mediante expresiones regulares información sobre calles, avenidas o plazas, junto con su número. Una vez hemos extraído la cadena buscamos sus coordenadas con el posicionador de Google.

Este módulo es usado para posicionar anuncios clasificados, en especial los que hablan sobre pisos en venta, que tienen una mayor probabilidad de especificar este tipo de información. Los feeds para recoger la información de los anuncios proviene de Loquo y hay un feed por provincia y categoría, por lo que conocemos información sobre la zona a la que pertenecen los anuncios. Esta información es empleada para 2 cosas:

1. Si no encontramos texto que identifique una calle o avenida, o si no obtenemos coordenadas del texto buscado, el anuncio es posicionado en la capital de provincia.
2. Si encontramos dicho texto y obtenemos coordenadas calculamos la distancia a la capital de provincia. Si está dentro de un límite aceptamos dichas coordenadas y en caso contrario asignamos las de la capital de provincia. Esto hace que no dependamos ciegamente del servicio de posicionamiento de Google, que podría darnos una información incorrecta debido a ambigüedades (al igual que hacemos con el corrector de coordenadas de los artículos de Wikipedia).

## De qué se habla hoy

El módulo implementado, aunque no utilizado finalmente en la interfaz del sitio web, sirve para realizar futuras ampliaciones proporcionando información sobre lugares, personas, asociaciones, etc de las que se está hablando en las noticias del día. De este modo se podría mostrar en un apartado estos conceptos con enlaces a páginas para ampliar información (las mismas páginas de noticias de las que se extraen estos nombres propios, artículos de Wikipedia en el caso de que existan...).

Básicamente el método imprime por la salida estándar de qué se está hablando en las noticias de hoy extrayendo los nombres propios de las noticias de hoy y comparando con los de las noticias de la última semana premiando aquellos que no han aparecido en esos últimos días, para evitar que los nombres propios que sean relativamente frecuentes en las noticias (por ejemplo *Estados Unidos, unión Europea*) aparezcan muy asiduamente.

### Funcionamiento

Primero buscamos, en las noticias de las últimas 24 horas, secuencias de 2 ó más palabras comenzadas por mayúsculas que aparezcan juntas o separadas por unas palabras preestablecidas (de, del, el, la, los, las). Un ejemplo de la salida (para el 3 de julio) es:

```
Enders Brooke
Hospital Virgen del Rocío de Sevilla
Corte Federal del Distrito Sur de Nueva York
Según la BPI
Top Spin
Time Warner
News Corp
Según Mozilla
El BCE
La CEOE
Si el Gobierno
British Air
El Brent
Banco Central Europeo
José Blanco
Palacio de Congresos de Madrid
Ministerio de Exteriores
El Cultural
Lengua Común
Fabrice Delloye
La ONU
The Audacity
Barack Obama
Hillary Clinton
El Fenerbahce
Luis Aragonés
El Fenerbahce
Oriol Giralt
Oriol Giralt
Joan Laporta
La Federación Francesa
Sergio Batista
Juegos de Atenas
...
```

En un segundo paso eliminamos las *stop words* del inicio de cada una de las secuencias (eliminando *El*, *La*, etc) cuya mayúscula sea por inicio de oración y no por nombre propio. Seguidamente comprobamos si las secuencias pertenecen a ciudades. En nuestro caso las ciudades son eliminadas porque nuestra

intención era más extraer nombres propios que no fueran ciudades (dado que las ciudades de estas noticias ya las conocemos gracias a haber aplicado previamente un geoposicionador más completo).

Después, con las secuencias más repetidas (en nuestro caso 30) consultamos en la base de datos de noticias las apariciones en los últimos 7 días (valor que divide al número de repeticiones) y ordenamos las secuencias en orden descendente, otorgando mayor peso a las más repetidas y que durante la última semana no hayan aparecido mucho.

Continuando el procesamiento del conjunto de secuencias tenemos como resultado:

Banco Central Europeo  
Jean-Pierre Escalettes  
American Airlines  
Federación Francesa de Fútbol  
Don Quijote  
Filmoteca Nacional  
Luis Buñuel  
Marc Gasol  
Organización del PSOE  
Ateneo Albacetense  
British Airways  
Confederación Hidrográfica del Miño  
Congreso del PP  
Continental Airlines  
José Blanco  
Pepe Sánchez  
Raymond Domenech  
Pau Gasol  
Real Madrid  
Axa Barcelona  
Carmen Oliver  
CC OO  
Copa Libertadores  
Policía Local  
Juegos Olímpicos de Beijing  
Ingrid Betancourt  
Partido Popular  
...

que nos da una idea de sobre qué se habla. De hecho, algunas de las noticias del día son:

- El **Banco Central Europeo** (BCE) ha decidido hoy subir los tipos básicos de interés en 25 puntos básicos, hasta el 4,25%, con lo que ha situado el precio del dinero en la zona euro...
- Domenech sigue al frente de la selección nacional de Francia. Está prevista para mañana una rueda de prensa por parte de **Jean-Pierre Escalettes**, Presidente de la Federación Francesa de Fútbol, donde se darán más detalles al respecto...
- La crisis del sector lleva a **American Airlines** a recortar 7.000 empleos...

Estos ejemplos ofrecen una visión general del funcionamiento del algoritmo, y como da prioridad a conceptos de los que no se ha hablado en los últimos días pero en las últimas 24 horas han aparecido frecuentemente.

## Componentes empleados

### *Slimbox*

Slimbox<sup>13</sup> es un clon visual del popular Lightbox JS v2.0, realizado por Lokesh Dhakar, escrito usando el framework ligero mootools. Fue diseñado para ser pequeño, eficiente y 100% compatible con el Lightbox v2 original.

Slimbox proporciona una forma atractiva para mostrar las imágenes de la capa de Commons y puntos propios, permitiendo además la navegación entre imágenes de una galería. El código de esta librería realizada en lenguaje javascript ha sido modificado por nosotros durante el desarrollo para mejorar el comportamiento y visualización de las imágenes. Estas mejoras consisten en:

- **Inclusión de timeout:** Dado que las imágenes son alojadas en servidores externos al nuestro, si se elimina alguna de estas imágenes nosotros no recibimos confirmación. La visualización de una imagen inexistente usando slimbox provoca un bloqueo en la página web, de forma que no se puede continuar navegando porque se queda indefinidamente cargando. Para solucionarlo hemos implementado un temporizador en javascript que permite cancelar la carga y avisar al usuario en caso de que la imagen no se cargue transcurrido n segundos (en nuestro caso establecido a 10).
- **Reescalado de las imágenes:** La librería original presupone que las imágenes están escaladas adecuadamente para quedar encuadradas en la ventana de navegación. Esta presunción hace que las imágenes excesivamente grandes hagan que la *ventana* emergente sea excesivamente grande y quede oculto el botón de cerrar, además de poder observar la imagen de forma completa. Nosotros hemos reescalado la ventana de la imagen según el tamaño de la imagen a visualizar y el área de navegación del cliente, dejando asimismo un espacio en la parte inferior para añadir la descripción de la imagen.

### *Videobox*

Videobox<sup>14</sup> es un script de 6kb que permite mostrar vídeos en una página en una capa superior. Está inspirado en Lightbox.v2 y utiliza parte del código de Slimbox. Está escrito también usando la librería mootools y usa un objeto swf para embeber flash.

Videobox es a los vídeos lo que Slimbox a las imágenes, y así lo hemos utilizado, compartiendo incluso imágenes y hojas de estilo css entre ambas librerías. Videobox permite mostrar vídeos de páginas como Youtube, Metacafe, Google Video, iFilm y películas de Flash.

Nosotros lo hemos utilizado para mostrar los vídeos de Youtube al hacer click sobre la miniatura que representa el vídeo, que es cargado en una capa y comienza su reproducción inmediatamente.

### *MooTools*

MooTools<sup>15</sup> es un framework JavaScript orientado a objetos, compacto y modular diseñado para el desarrollador JavaScript intermedio y avanzado. Permite escribir código potente, flexible y multinavegador con una API elegante, bien documentada y coherente.

---

<sup>13</sup> Slimbox está disponible en <http://www.digitalia.be/software/slimbox>

<sup>14</sup> Videobox está disponible en <http://videobox-lb.sourceforge.net>

<sup>15</sup> Mootools se puede encontrar en <http://mootools.net>

MooTools nos ha proporcionado una plataforma para poder realizar llamadas asíncronas utilizando Ajax fácilmente, así como ser la base para Slimbox y Videobox. Además hemos utilizado alguno de sus efectos (como el efecto Acordeón<sup>16</sup> en la página de *Añadir contenido*).

### **PDMarker**

PDMarker<sup>17</sup> es una librería escrita en Javascript que extiende el API de Google Maps, haciendo más fácil personalizar el comportamiento de los marcadores. Nosotros lo hemos utilizado para modificar el índice z de los marcadores cuando se pasa el ratón por encima, ya que cuando se solapan marcadores cercanos es difícil saber a cuál se hace referencia. Esta mejora se ha incluido junto al uso de un *tooltip* con el nombre del marcador para facilitar al usuario conocer qué marcador quiere abrir.

### **Dragzoom**

Dragzoom<sup>18</sup> es un control que se añade al mapa de Google Maps para definir una región sobre la que hacer zoom pinchando y arrastrando, de forma que el usuario no tenga que ir haciendo zoom y arrastrando para poder acceder a la región deseada. El control se añade al mapa como los controles de zoom o búsqueda, de forma que se integra perfectamente.

La inclusión del dragzoom responde a la necesidad de facilitar al usuario la navegación por el mapa, complementándolo con zoom al realizar doble click o mover la rueda del ratón.

---

<sup>16</sup> La página donde se explica el efecto acordeón es <http://demos.mootools.net/Accordion>

<sup>17</sup> PDMarker tiene su web en <http://www.pixeldevelopment.com/pdmarker.asp>

<sup>18</sup> La explicación del uso de Dragzoom <http://googlemapsapi.blogspot.com/2007/08/dragzoom-marker-manager-cluster-zoom.html>

## Geoposicionamiento por IP

Cuando un usuario entra por primera vez al sitio web queríamos que pudiera ver información relacionada con la zona donde está situado. A través de la dirección IP de la visita hay servicios y bases de datos que proponen una coordenada geográfica para posicionar esa dirección, en base a las direcciones que contienen los distintos ISP y otros datos.

Tras probar diversos servicios de este tipo decidimos utilizar finalmente la base de datos GeoLite City de Maxmind<sup>19</sup>. GeoLite City es una base de datos de cobertura global que permite operaciones de localización de IPs. Es gratuita, al contrario que la base de datos GeolP City, pero es menos precisa. Permite geoposicionar una dirección IP a nivel de ciudad, aunque tiene una precisión variable.

### Precisión

Desde Maxmind se da una precisión del 99,3% a nivel de país y 74% a nivel de ciudad para Estados Unidos en un radio de 25 millas. Para España se da un 51% de acierto a nivel de ciudad en 25 millas, 45% de error y un 4% de direcciones que no son cubiertas a nivel de ciudad. Hemos realizado multitud de pruebas desde varios equipos y hemos comprobado que en ocasiones el posicionamiento no era correcto (por ejemplo visitas de la provincia de Alicante eran situadas en Cataluña o País Vasco). En cambio, a nivel de país, funciona correctamente dado que hemos comprobado que el posicionamiento de las visitas coincide con el otorgado por Google Analytics.

### Uso

La base de datos de GeoLite City es un archivo que es actualizado mensualmente (el primer día de cada mes) y que se puede obtener desde la web de Maxmind. El archivo, de unos 25MB, lo hemos alojado en geoip/GeoLiteCity.dat. Para su uso se facilitan múltiples API<sup>20</sup> para muchos lenguajes de programación (C, Perl, PHP, Java, Python...) cubriendo la práctica totalidad de lenguajes. Las API se acompañan de diversos archivos de ejemplo que muestran cómo utilizar el servicio. Nosotros hemos usado el módulo para PHP (aunque hay incluso un módulo para Apache).

Al obtener el país de la visita a través de la IP debemos buscar las coordenadas de dicho país, para lo cual hemos usado el geoposicionador de Google, que devuelve las coordenadas de, aproximadamente, el centro del país que se le pasa como lugar a posicionar. La información de país con sus coordenadas la hemos guardado en una tabla en la base de datos, lo que nos permite *ahorrar* llamadas al servicio de Google en cada visita que deba ser posicionada, así como llevar un seguimiento de los países de los cuales nos visitan.

## Otras formas de posicionar el mapa

Cuando el usuario sale de nuestro sitio web se almacena en una cookie los datos de posición y zoom del mapa para poder volver a cargar esa vista cuando vuelva a entrar. Además, es posible acceder al sitio a través de una url donde se especifican los parámetros necesarios para posicionar el mapa (generada a través de *Enlazar aquí*). Así, el orden en que se aplica cada una de las reglas para situar el mapa son:

1. Coordenadas en la URL
2. Coordenadas en la cookie
3. Geoposicionar IP

<sup>19</sup> La base de datos para la geolocalización está disponible en <http://www.maxmind.com/app/geolitecity>

<sup>20</sup> Las API de las que tienen soporte están en <http://www.maxmind.com/app/api>

## Planificación de actividades

El siguiente es un cuadro con las principales actividades realizadas en cada mes del desarrollo:

Actividad	Octubre	Noviembre	Diciembre	Enero
Investigación	Búsqueda de bases de datos de información de noticias y de contenido posicionado (dbpedia). Búsqueda de wikis basados en mediawiki. Establecimiento de herramientas de trabajo.	Búsqueda de servidores. Estudio de sistemas CVS y generadores de capa de acceso a datos DAO	Búsqueda de algoritmos de categorización de textos y herramientas stemmer. Estudio de los distintos geolocalizadores por IP. Pequeñas pruebas de Google Charts.	Búsqueda de métodos para realizar la actualización periódica de los feeds (crontab)
Base de datos	Uso de MySQL y PHPMyAdmin incluidos en paquete EasyPHP.	Tablas de feeds de noticias, noticias, artículos de wikipedia e información de meteoclimatic, así como gestión de feeds.	Exportaciones a servidor del contenido local. Tablas para Commons.	
Implementación interfaz	Primeras interfaces web y pruebas	Capas para noticias y wikipedia. Clasificador para posicionar artículos de Wikipedia en el mapa	Capa de Wikimedia Commons. Clasificador para posicionar imágenes de Commons. Agrupación de imágenes en un mismo punto.	Inclusión de encabezado. Internacionalización de la interfaz.
Implementación contenido	Primera versión del script Python para Wikipedia	Primeras pruebas de recuperación de información (meteoclimatic, noticias y wikipedia). Corrección de coordenadas en artículos de Wikipedia.	Obtención de imágenes de Wikimedia Commons. Obtención de noticias. Primeros esbozos de categorizadores de noticias	Obtención de vídeos de Youtube. Eliminación de paréntesis en artículos de Wikipedia.
Documentación	Primeros desarrollos de documentación y gestión de información sobre los diversos aspectos investigados.			

Actividad	Febrero	Marzo	Abril	Mayo
Investigación	Generación de miniaturas en PHP.	Obtención de imágenes de servidores externos con curl. Métodos para exportar bases de datos con muchos registros. Ejecución de scripts Python desde PHP.	Búsqueda de fuentes de anuncios clasificados. Envío de correo electrónico en el registro. Traducción en línea de noticias en otros idiomas.	
Base de datos		Uso de Bigdump. Tablas de usuarios y contribuciones.		Tablas para variables globales y registro de visitas.
Implementación interfaz	Página comunidad de usuarios. Página "acerca de" del proyecto.	Zonas pública y privada. Gestión de registro e identificación de usuarios. Primeras subidas de contenido desde la interfaz. Capa de Youtube. Añadido "enlazar aquí". Adición y mantenimiento de artículos de Wikipedia. Añadida capa de meteorología. Filtro de noticias por categoría.	Información de errores desde el globo. Marcado de informes como revisados. Cambio de datos del registro. Añadido de artículos de Wikipedia por el nombre. Añadida capa de anuncios. Añadido "Mi mapa" para usar mapas en iframes. Añadido filtro al contenido propio. Posicionamiento del mapa en país de origen de la visita. Subida de imágenes de Commons a partir del nombre. Añadida página para ver contribuciones de usuario. Añadida opción para marcar puntos como inapropiados. Solicitud de contraseñas autogeneradas. Noticias en múltiples idiomas.	Implementación de "cambios recientes". Timeout para slimbox. Ajustes para visualización correcta en todos los navegadores. Mejoras en la gestión de marcadores en mapas de Google. Clasificador de puntos para Wikimedia Commons
Implementación contenido	Generación de miniaturas a partir de imágenes.	Obtención de meteorología de Google. Implementación de CGIs basados en los scripts python	Implementación de informes de error. Subida de imágenes y miniaturización definitiva. Obtención de anuncios clasificados y geoposicionamiento. Obtención de noticias internacionales y traducción.	Implementación del cambio manual en el número de avisos de inapropiados para ocultar contenido. Registro de visitas. Se añaden noticias de sucesos
Documentación				Documentación de la aplicación WAR para Tomcat y reestructurado y limpieza de código. Documentación de la parte web y python.

## Conclusiones

El desarrollo de aplicaciones web interactivas con el uso de técnicas de actualización asíncronas proporciona al usuario la sensación de estar trabajando con una aplicación tradicional de escritorio. Hemos utilizado estas tecnologías para construir una aplicación web que permite al usuario interactuar con una gran cantidad de contenido variado y sentirse partícipe aportando modificaciones, información de errores existentes y nuevo contenido.

Hemos tenido algunas dificultades para integrar el desarrollo de este proyecto utilizando diversos lenguajes de programación (python, Php, Java, Javascript...) lo que ha hecho que en ocasiones hubiera problemas para utilizar una codificación de caracteres común.

El desarrollo de la parte Java ha sido algo dificultosa en la parte del despliegue ya que no es tan sencillo como trabajar con archivo PHP o Javascript, teniendo que actualizar la aplicación completamente con cada cambio aunque, eso sí, el trabajar con un Entorno de Desarrollo Integrado como Eclipse hace que los errores disminuyan y se puedan encontrar más fácilmente.

El desarrollo de la parte web en Javascript ha sido bastante complicado porque hemos tenido que enfrentarnos a las diferentes formas de interpretación por parte de cada navegador. Además Javascript es difícil de depurar sin una herramienta como Firebug, que nos ha facilitado bastante el desarrollo, dado que las llamadas asíncronas han sido más fáciles de controlar.

Python nos ha supuesto algunas dificultades, tanto en la interacción con Php y Mysql como en el desarrollo en sí mismo dado que no disponíamos de suficiente experiencia de trabajo con este lenguaje de programación. Además, la obtención de los datos de Wikipedia y Wikimedia Commons ha estado supeditada a los permisos establecidos desde la fundación Wikimedia y de la forma de recorrer los distintos artículos de un wiki basado en Mediawiki.

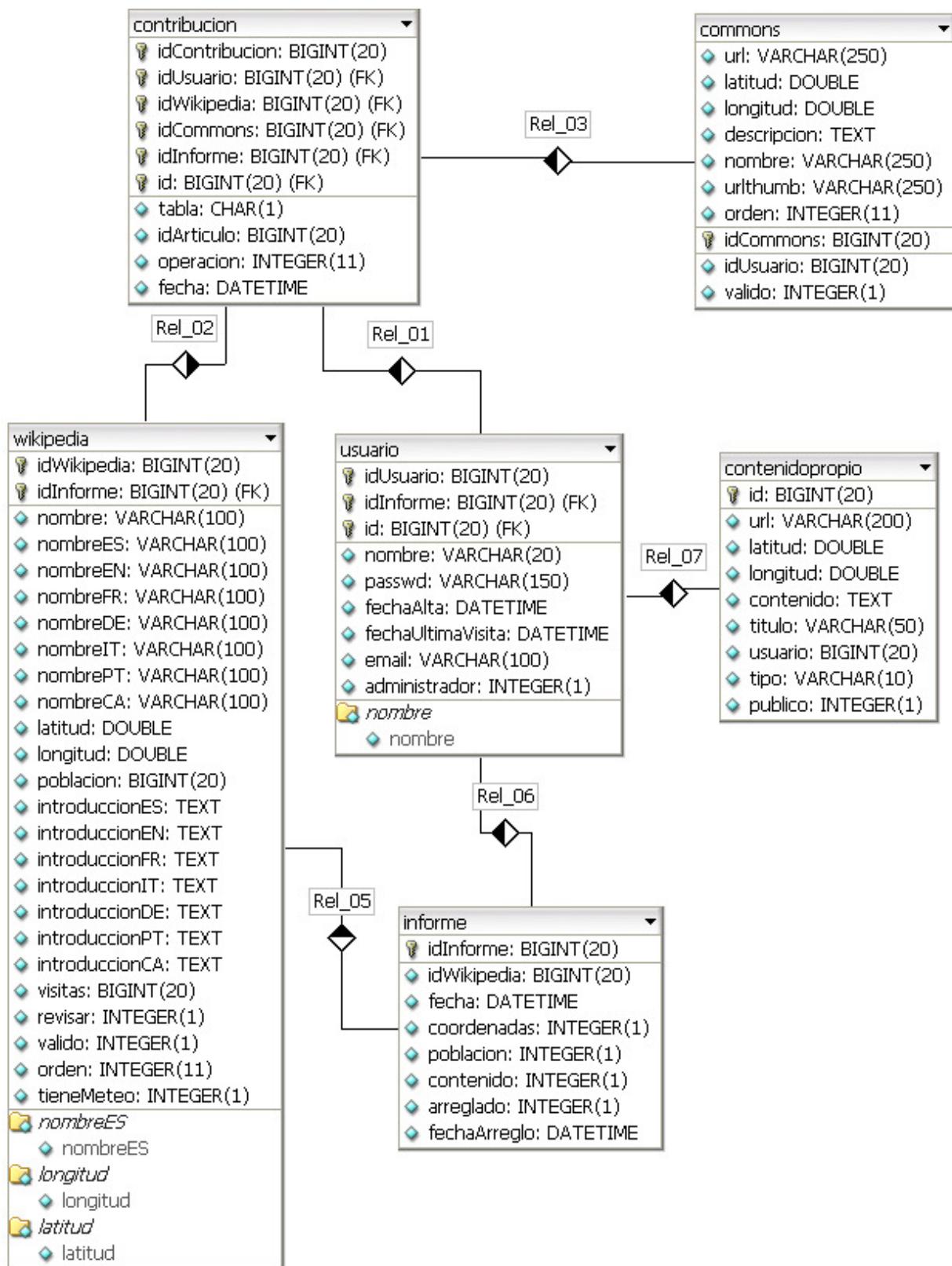
Para obtener los distintos contenidos (especialmente Wikipedia y Commons) ha sido necesaria la ejecución de scripts de Python durante muchas horas (incluso varios días seguidos) para recorrer las distintas versiones de Wikipedia. Las noticias han sido obtenidas durante un período largo de tiempo para disponer de información suficiente para realizar correctamente el clasificador.

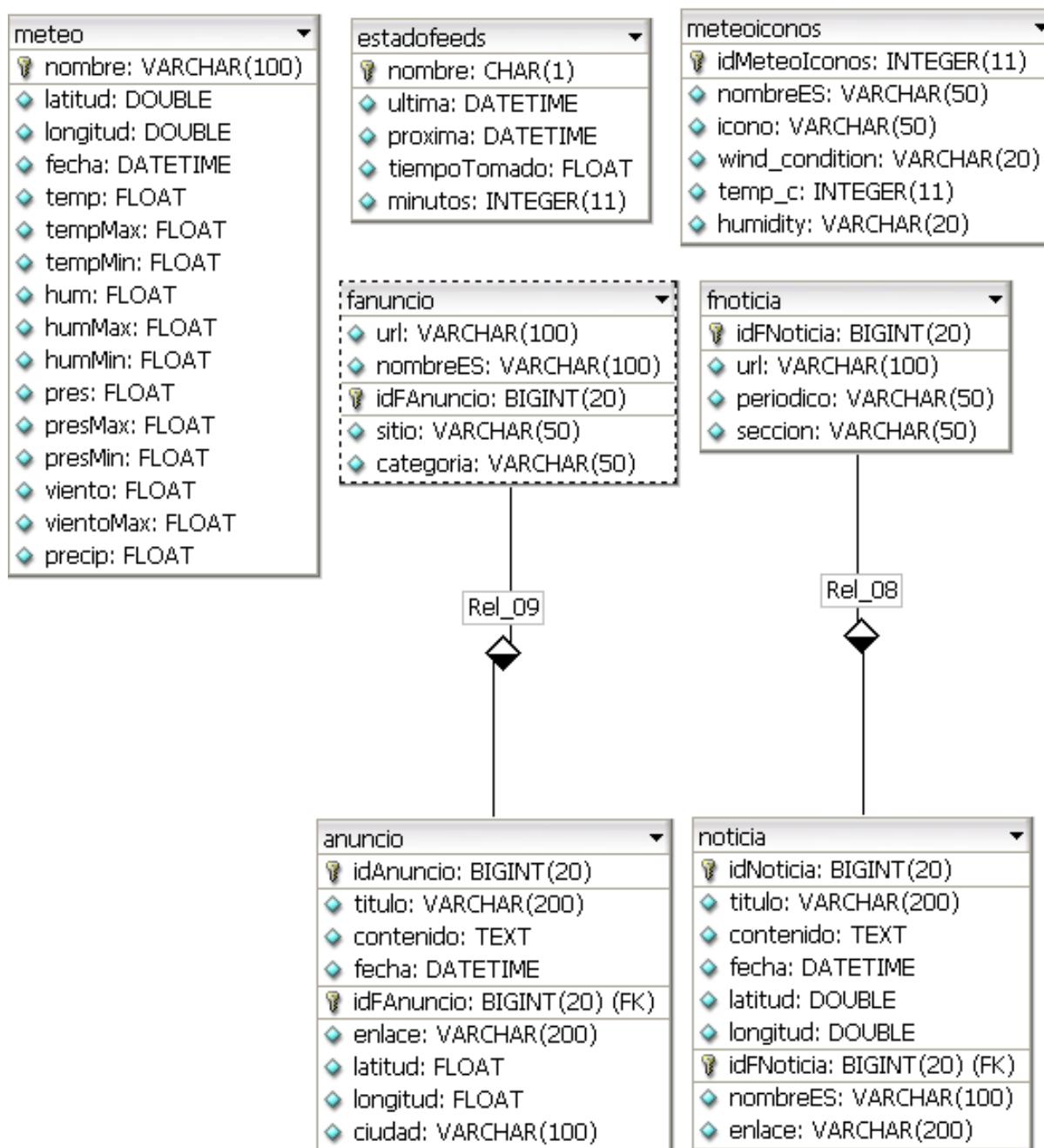
Pensamos que, en conclusión, este tipo de aplicaciones tienen una gran utilidad dado que son capaces de presentar de una forma clara, fácil y accesible grandes cantidades de información, representando un método distinto de acceso (primando su posición geográfica).

En general estamos satisfechos con el apoyo obtenido por parte del tutor del proyecto, Francisco Ortiz, así como de Juan Galiana que ha sabido gestionar el servidor web a la perfección, proporcionándonos las aplicaciones necesarias y algunos consejos sobre el desarrollo.

Pensamos que la realización de este proyecto ha mejorado nuestra experiencia en el desarrollo web y en la gestión de un sistema con grandes cantidades de datos, así como de la interacción entre diferentes módulos. Además, hemos podido aplicar técnicas de gestión de proyectos, dividiendo las distintas tareas entre los miembros, abordando eficazmente el proyecto a pesar de la gran parte de investigación y la cantidad de datos a obtener y procesar.

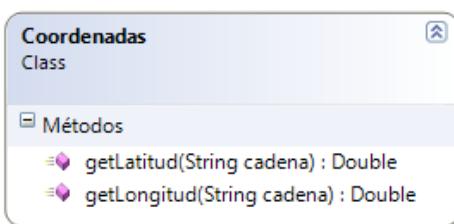
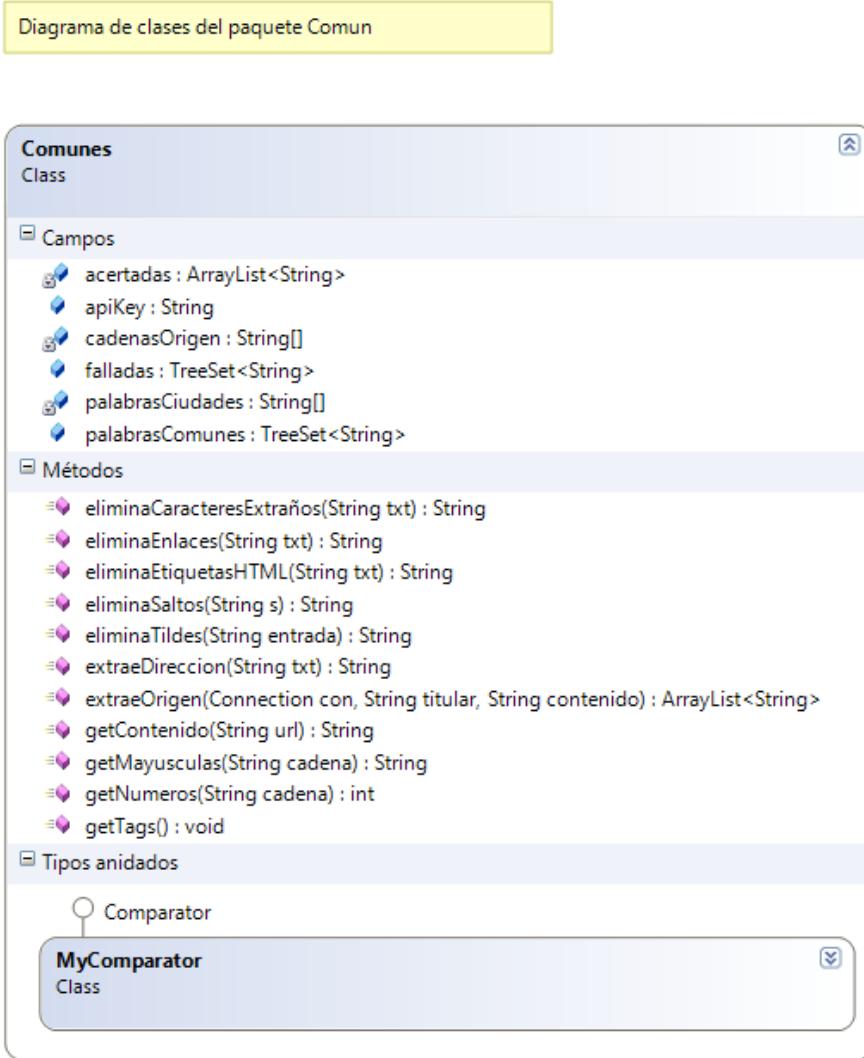
## Anexo I: Diagramas de la base de datos

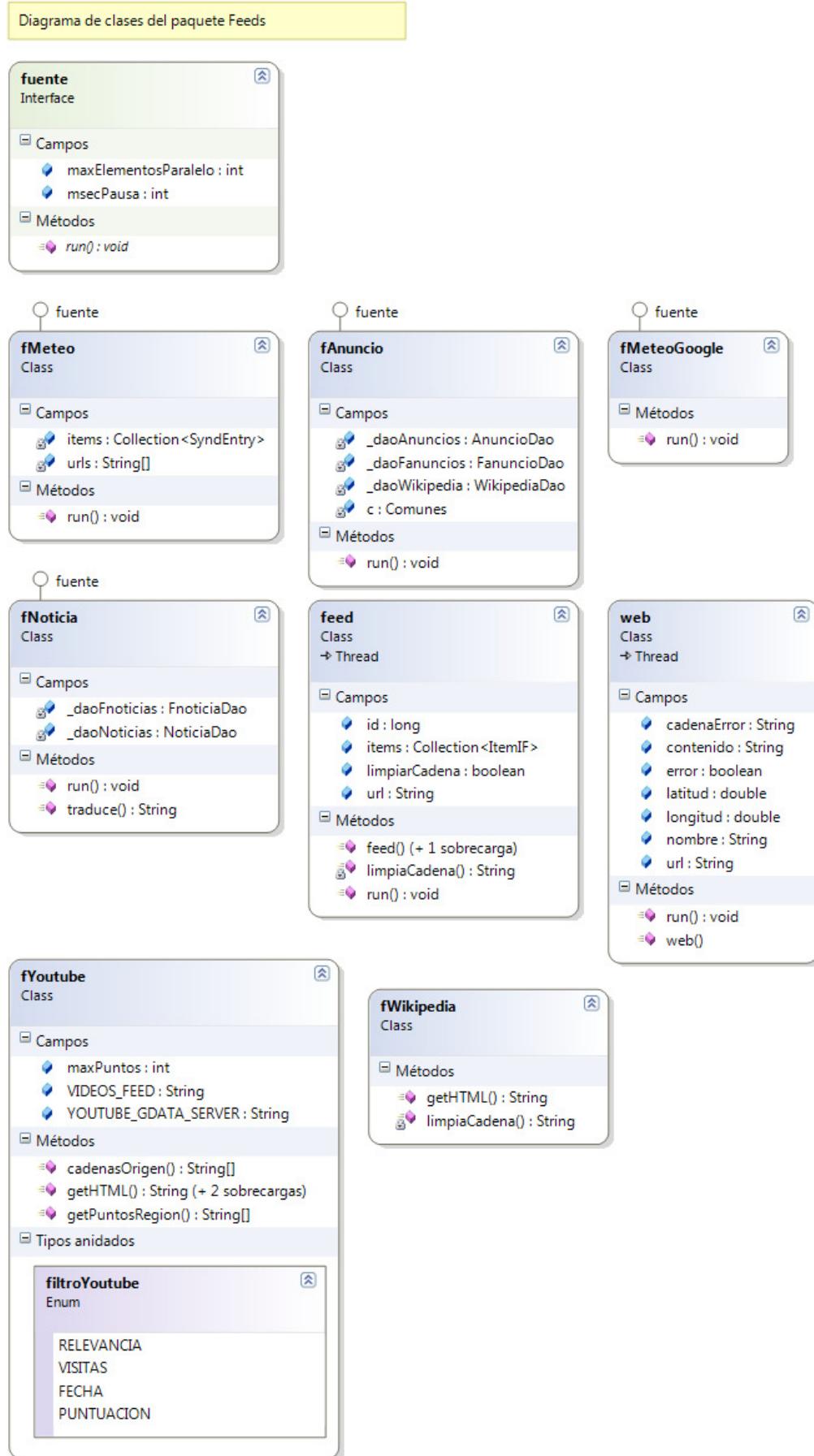


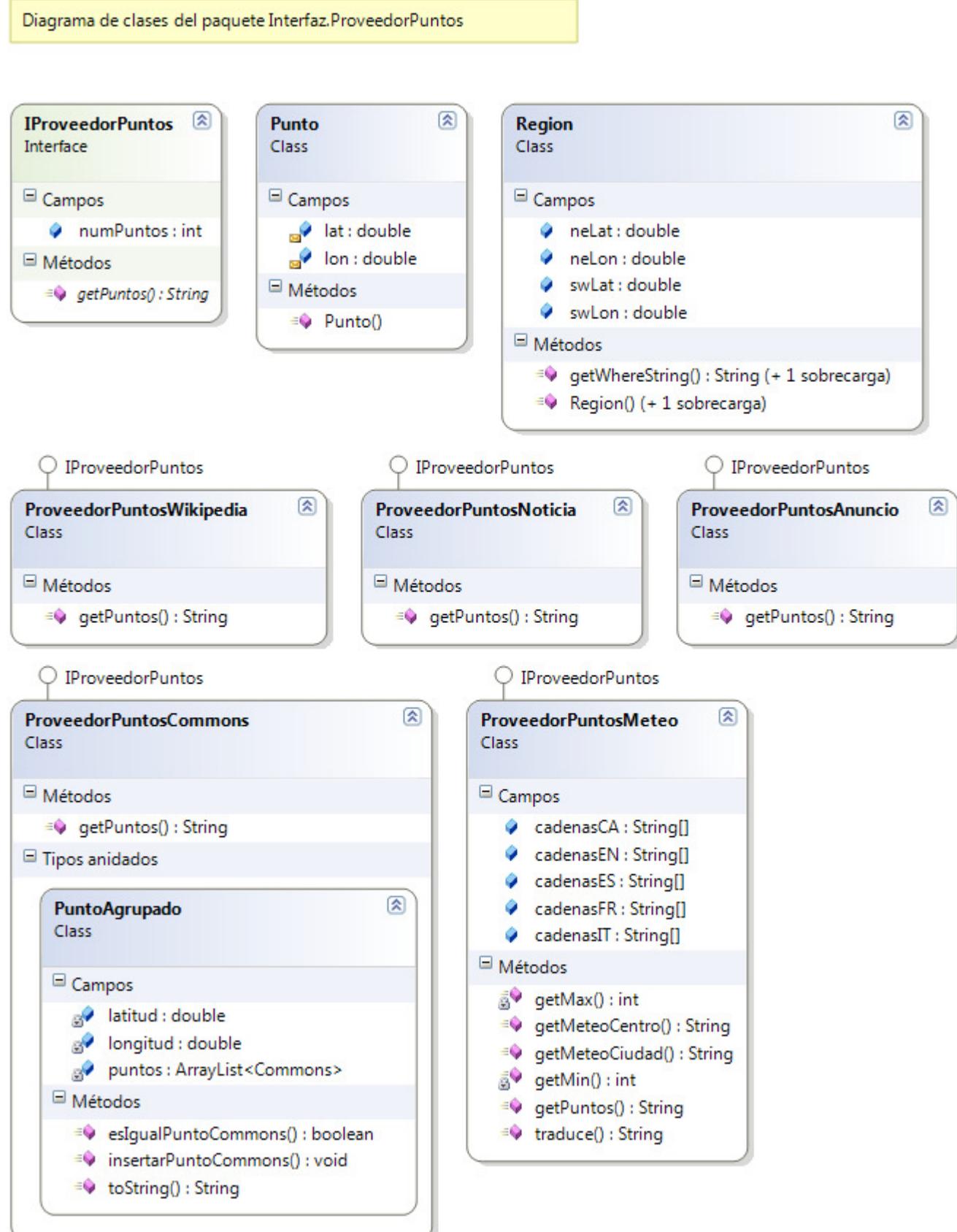


## Anexo II: Diagramas de clases de la aplicación Java

A continuación presentamos los diagramas de clases de la aplicación en lenguaje Java que es contenida en el servidor Tomcat.







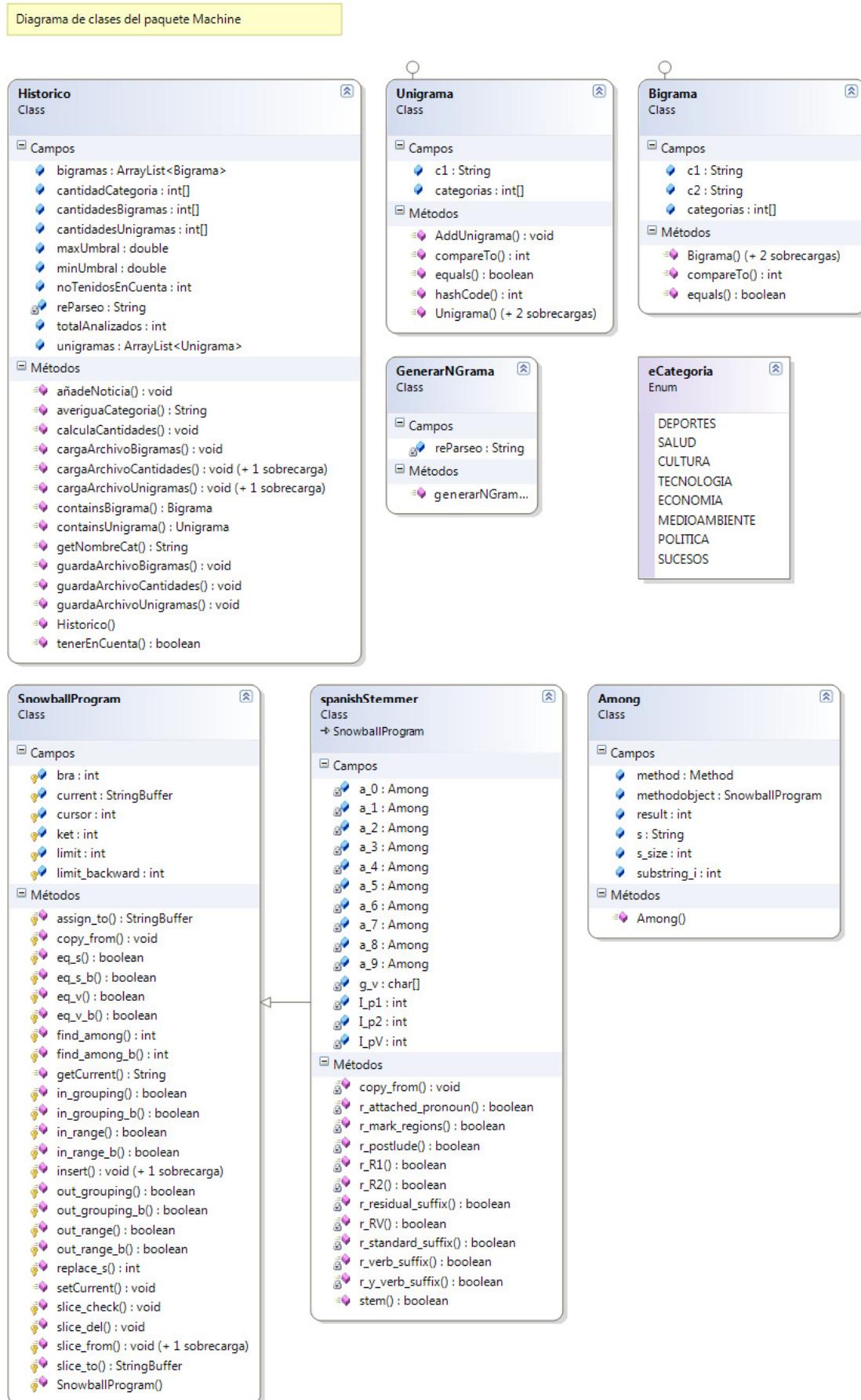


Diagrama de clases del paquete Mantenimiento

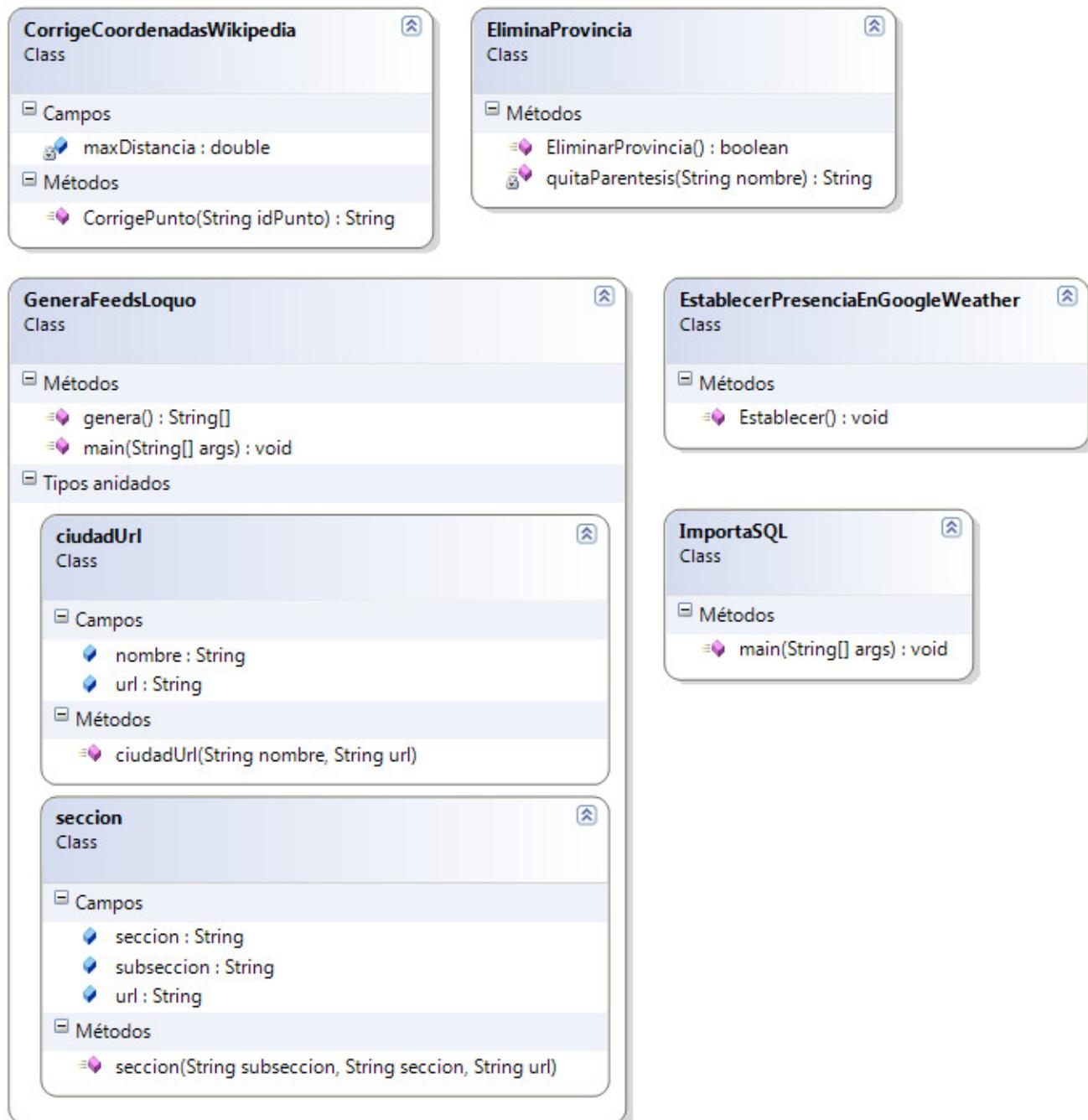
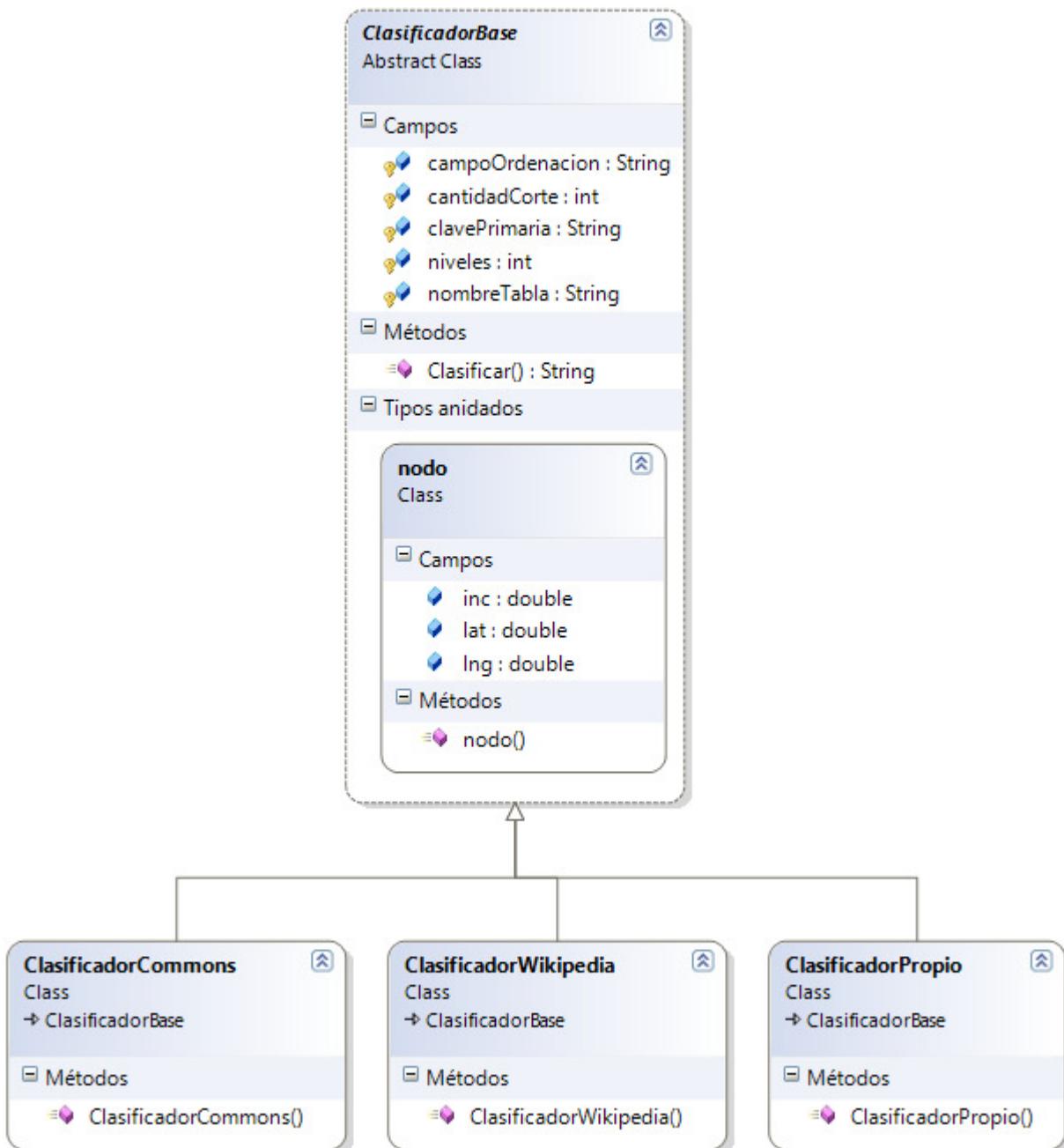
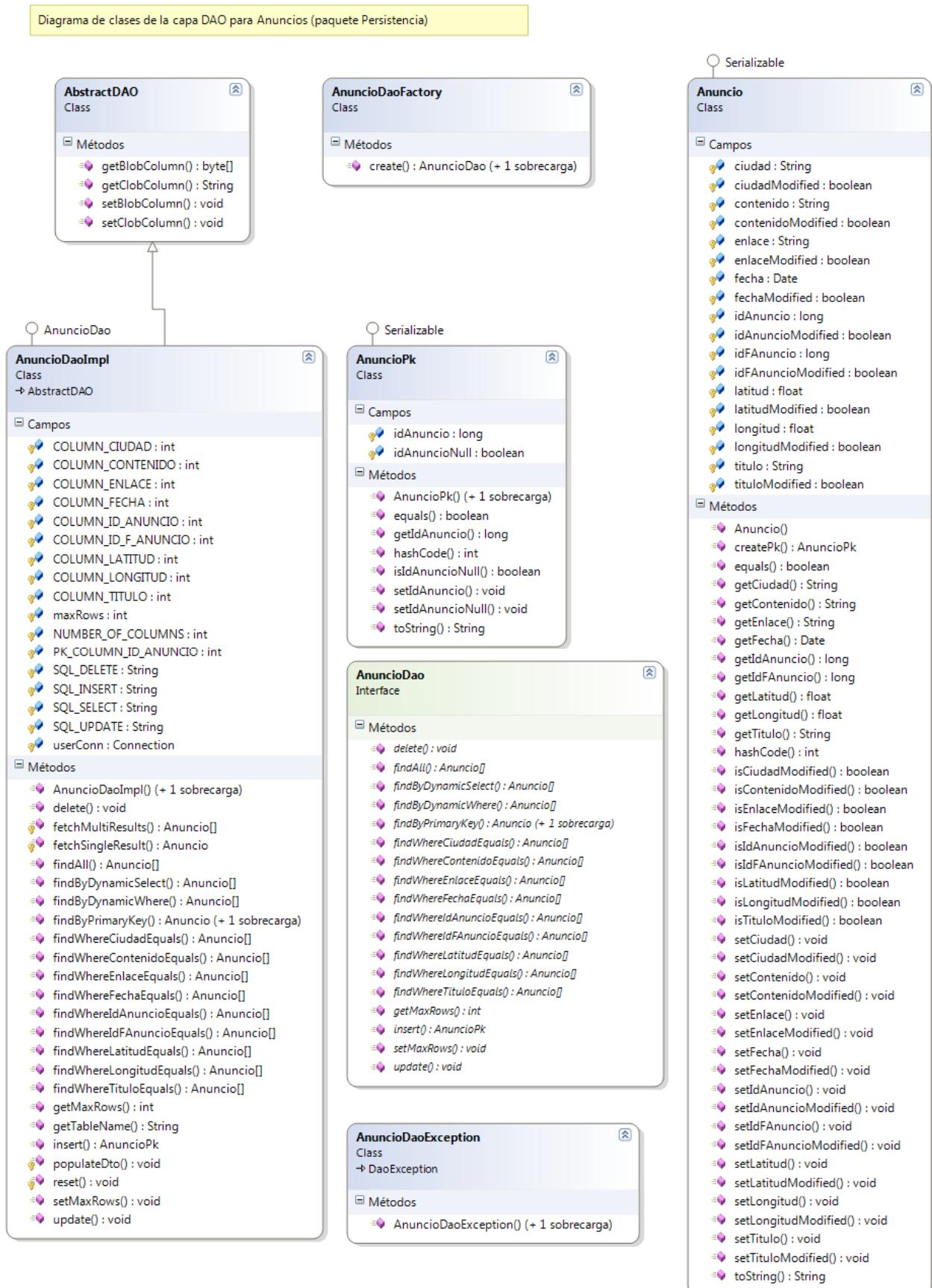


Diagrama de clases del paquete Mantenimiento.Clasificador





La estructura de clases para la entidad Anuncio se replica de forma similar para cada una de las tablas.