(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.119-133. ISBN 84-931933-1-3.

Acondicionando grandes ortoimágenes para su visualización por Intranet, una aproximación basada en Java[1]

Ó. Cantán, P. Fernández, M. Á. Latre, J. Nogueras, J. Valiño

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
María de Luna 3, 50015 Zaragoza, Spain
{ocantan, pedrofb, latre, jnog}@ebro.cps.unizar.es, juanv@posta.unizar.es

Resumen: Este trabajo aborda el problema de visualización de grandes imágenes geográficas por intranet. Los problemas se derivan de las limitaciones de velocidad de acceso a la red, lo que conlleva estrictos requisitos en el tamaño de la información a transmitir. En el artículo se muestran algunas de las ventajas prácticas derivadas de la utilización de Java en el desarrollo de este software y la estrategia adoptada para abordar los problemas de limitación de tamaño, problemas que se resuelven tanto desde el punto de vista de los usuarios como desde el de los requerimientos de red. La aproximación adoptada permite a su vez integrar las ortoimágenes con otro tipo de información geográfica vectorial. El desarrollo sobre el que versa este artículo también ejemplifica cómo proporcionar a los usuarios un sistema portable, capaz de coordinar la gestión de imágenes de gran tamaño desde distintos puntos de la red.

Palabras clave: *Java, SIG, grandes imágenes, applet, servidor web, sistema de información, arquitecturas distribuidas Java, orientación a objeto*

Introducción

Java está ganando una rápida y gran aceptación como lenguaje de programación en los últimos años. [SM98]. Sus capacidades de programación, sus implementaciones multiplataforma, sus prestaciones para el desarrollo de aplicaciones distribuidas que incluyen un mecanismo de comunicaciones de alto nivel, las potentísimas librerías estándar y la gran disponibilidad de software de dominio público han hecho de Java uno de las herramientas más adecuadas para el desarrollo de software.

En la actualidad, Internet y el web son, sin duda, el medio más utilizado para compartir de información y uno de los mecanismos más poderosos para proporcionar acceso a información de carácter público, donde se requiere de algún modo interacción con el usuario.

La comunidad SIG está viviendo un impresionante incremento del interés que despiertan las tecnologías basadas en Java y el web. En este sentido, las compañías comerciales más relevantes del mundo del SIG y de la información geográfica están dedicando recursos importantes al desarrollo de productos software con capacidades para internet, muchos de ellos escritos en Java o con capacidad para interactuar con este lenguaje de programación (Esri, MapInfo, Oracle, Intergraph, ...). El lector puede acudir a [CLH99] o [LIM99] si desea obtener reseñas y comparaciones de estos productos. Decenas de applets con funcionalidad SIG pueden encontrarse en la red, muchos de ellos de disponibilidad pública (véase [SM98] para obtener un informe con enlaces web sumamente interesantes). Podemos mencionar sistemas como Descartes [AAC99], GAEA [KP99], GeoToolKit [BBBCS99], InovaGIS [GC99], … Por otra parte, los esfuerzos de organizaciones dedicadas a la estandarización también se han volcado sobre los mundos de Java e internet. Por ejemplo, el consorcio OpenGis ha dado mayor prioridad a la especificación de las interfaces de servidores web de mapas y sus implementaciones en Java frente a otro tipo de estándares.

[1] **Agradecimientos**: La tecnología básica de este sistema ha estado parcialmente financiada por la *Comisión Interministerial de Ciencia y Tecnología* (CICYT) mediante del proyecto TIC98-0587.

El trabajo de P. Fernández Bel ha estado parcialmente costeado mediante la beca predoctoral B109/99 financiada por el Gobierno de Aragón y el Fondo Social Europeo.

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.119-133. ISBN 84-931933-1-3.

Explicaremos a continuación un ejemplo real típico donde este tipo de tecnología puede ser aplicada. Hace dos años, el Ministerio de Medio Ambiente decidió adquirir las imágenes tomadas por los satélites SPOT y Landsat correspondientes a una de sus zonas de interés: la cuenca hidrográfica del río Ebro. Las imágenes de cuenca son utilizadas como ayuda para los geógrafos a la hora de demarcar ríos, pozos de agua, tierras irrigadas y tipos de cultivo. Las imágenes cubren una extensión de más de 85000 km^2. Con el objetivo de aumentar el beneficio social de esta inversión, la administración decidió otorgar acceso a esta información tanto dentro del Ministerio, como al público en general.

Para aumentar la utilidad de esta información para el usuario y para facilitar el acceso a la misma, el Ministerio decidió proporcionar el software necesario para su visualización. El software debía poseer características y funcionalidades propias de los sistemas de visualización SIG tradicionales, junto con la capacidad de combinar las imágenes con otro tipo de información geográfica vectorial, todo ellos junto a unos requisitos mínimos referentes al acceso y velocidad de conexión. Ofrecer una apariencia similar a la de los sistemas SIG tradicionales puede, por ejemplo, ayudar a los geógrafos a localizar de forma rápida accidentes geográficos sobre las imágenes, aprovechando de este modo, junto a la sobreimpresión de coberturas vectoriales, el gran depósito de imágenes de que se dispone. El problema surge cuando es necesario trabajar con imágenes de gran tamaño que deben ser gestionadas y enviadas a través de una red.

Para obtener el software necesario, estudiamos alternativas que contemplaban distintos productos comerciales, observando que, por norma general, exigían el pago de costosas licencias y establecían restricciones para el desarrollo y programación con ellos. Con el objeto de evitar estos problemas, se tomó la decisión de desarrollar nuestros propios componentes de visualización SIG para manipular las imágenes y la información vectorial, siendo Java el lenguaje escogido para el desarrollo de estos componentes, ya que permite de forma sencilla la distribución de los mismos en una red. Al utilizar Java como lenguaje de programación, también nos beneficiamos de la capacidad de las clases especializadas en el tratamiento de imágenes.

El objetivo de este artículo es el de describir esta experiencia técnica. El resto de este artículo se ha estructurado de la siguiente manera: el siguiente apartado describe con más detalle el contexto de la aplicación, junto con los problemas que plantea. La tercera sección muestra la estructura del depósito de imágenes construido para la gestión de imágenes de gran tamaño. La cuarta sección describe la arquitectura de la solución adoptada finalmente, para finalizar con una sección de conclusiones.

Contexto del sistema

El objetivo de los autores es el de desarrollar una tecnología capaz de facilitar el acceso a un repositorio de imágenes con grandes cantidades de información de uso frecuente desde distintos puntos de conexión dentro de una red de área local o intranet.

Se ha elegido Java como lenguaje de programación por ser éste un lenguaje multiplataforma que puede ser interpretado en cualquier entorno en el que se disponga de una máquina virtual Java. Esta característica, junto con las facilidades que proporciona para el desarrollo de software distribuido (incluyendo la gestión de la comunicación entre objetos distribuidos) y la posibilidad de ser interpretado en los navegadores tradicionales de internet, han sido los principales factores que han influido en esta elección. Otras de sus características también han sido de gran utilidad en el desarrollo de la aplicación, como, por ejemplo, la librería de gestión de imágenes que proporciona.

La arquitectura que se propone cuenta como módulo principal con un componente que ya había sido desarrollado previamente, al que denominamos Visual GIS. Este módulo está encargado de localizar, cargar y mostrar las imágenes e información vectorial procedentes del repositorio. Las capacidades de este módulo han sido aumentadas para adecuarse a los requisitos de funcionamiento de la aplicación.

Antes de afrontar el proceso de construcción del software, es necesario prestar atención a las restricciones impuestas por la arquitectura, sobre todo las relativas al tiempo de acceso y a la velocidad de transmisión. La gran cantidad de datos contenida en el repositorio obliga a la búsqueda y desarrollo de una solución eficaz y eficiente al problema de la gestión de las imágenes. El enfoque más simple (consistente en meramente construir una imagen completa de la zona de interés y trabajar con ella) habría hecho disminuir enormemente la complejidad de la aplicación, puesto que se podría trabajar utilizando un sistema tradicional SIG. No es probable, sin embargo, que los usuarios estuvieran satisfechos con las prestaciones que este enfoque les proporcionara, puesto que una operación sencilla, como un zoom, podría tardar varios minutos en ser procesada. Las causas de los altos tiempos de respuesta de esta aproximación de esta aproximación son, principalmente, el

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.119-133. ISBN 84-931933-1-3.

gran tamaño de la imagen con la que debería trabajarse y, en menor medida, la disponibilidad de un ancho de banda limitado en un entorno intranet.

Esta sencilla aproximación que se acaba de presentar olvida un aspecto básico: que los usuarios sólo necesitan una pequeña parte de la imagen en un determinado momento. La solución que se propone es la de partir las imágenes fuente disponibles en otras de menor tamaño, formando un repositorio de imágenes que puede ser usado por una aplicación específica. El repositorio no sólo estará compuesto de imágenes a la misma escala que la imagen original, sino que también se añadirán imágenes en las que el área cubierta por un solo píxel sea mayor que la cubierta en la original. De esta forma, aumenta la velocidad de visualización de zonas cuyo tamaño sea mayor que las cubiertas por una sola de las imágenes de pequeño tamaño [BGS99].

Se ha realizado un esfuerzo por conseguir un módulo versátil con las características SIG básicas que, inicialmente, sólo está preparado para trabajar en un entorno local. Este módulo, denominado Visual GIS, ha sido desarrollado para poder ser mejorado, con poco esfuerzo adicional, de forma que sea posible su uso en un entrono en red. Para ello, el módulo mantiene su capacidad de procesado y se añade un cliente capaz de acceder a la información, mientras que en un equipo servidor se ubican las imágenes y un proceso servidor de éstas.

Trabajando con conjuntos de imágenes de gran tamaño

En las secciones siguientes se describe tanto el proceso de partición y composición que se ha aplicado a las imágenes originales, con el objetivo de formar el depósito de imágenes, como los pasos requeridos por una aplicación que desee buscar y obtener imágenes de este depósito.

Generación del depósito de imágenes

Las imágenes originales forman una malla irregular y están organizadas en 217 ficheros. Cada fichero es una imagen georreferenciada SPOT en formato BIP de 30 x 20 km aproximadamente y una escala de 5 metros por píxel. Las imágenes que forman la cuadrícula están superpuestas, de forma que cada imagen tiene un tamaño mayor que el de el cuadro que le correspondería en la malla. La cuadrícula está almacenada en formato Shapefile de ESRI.

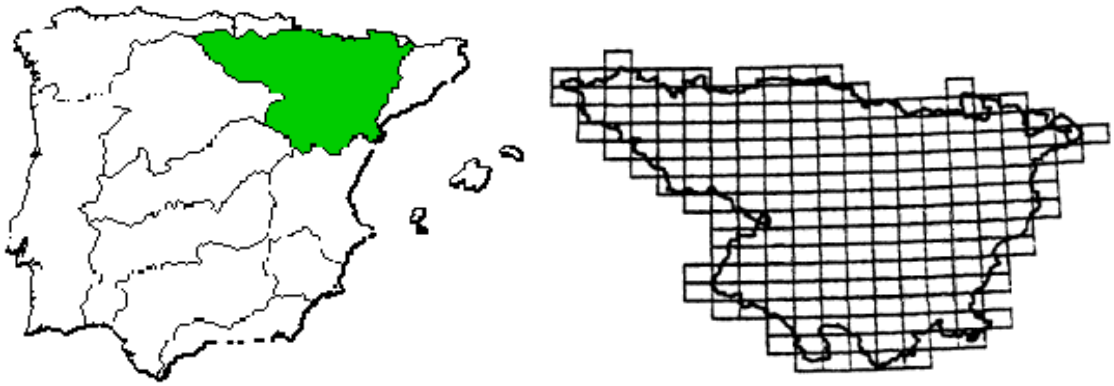

Fig. 1. Cuenca hidrológica del Ebro y la malla formada por las imágenes SPOT.

Cada imagen es de aproximadamente 6000 × 4000 píxeles y ocupa más de 80 Mb de espacio en disco. El formato BIT almacena puntos RGB, sin aplicar ningún tipo de compresión. Otros formatos gráficos, como JPEG, GIF o PNG permiten reducir el tamaño de la imagen, ya sea sin pérdida de calidad o con una pérdida mínima). Utilizando cualquiera de estos formatos, el tamaño de las imágenes se puede reducir a, aproximadamente, tan sólo 21Gb. En este caso, se eligió el formato JPEG debido a su excelente tasa de compresión y a que Java v1.2 proporciona métodos estándar para convertir imágenes en ficheros JPEG.

Para aumentar la velocidad de acceso a las imágenes, la compresión de las mismas no es suficiente. Es necesario partir las imágenes originales en pequeñas imágenes JPEG, de forma que se facilite su transferencia y carga. El depósito de imágenes resultante estará compuesto por gran cantidad de imágenes de pequeño tamaño organizadas en diferentes niveles de escala. Todas las imágenes de un nivel de escala conforman una imagen virtual de la cuenca. El primer nivel es una imagen completa de la misma, mientras que el último está compuesto

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.119-133. ISBN 84-931933-1-3.

de imágenes a la misma escala que la de las imágenes originales. Una imagen de un determinado nivel cubre la misma área que cuatro imágenes del nivel inmediatamente inferior, aunque su tamaño no es cuatro veces mayor, sino igual al de una de las imágenes del nivel inferior.

La construcción del depósito de imágenes está formada por dos procesos. El primero de ellos consiste en partir las imágenes originales, con el objetivo de componer las del último nivel. El segundo, en combinar las imágenes de un nivel para construir las del nivel superior. Ambos procesos se basan en una estructura ideal de cuadriculado. Estos procesos crean un depósito de datos de siete niveles de escala distintos. El cuarto nivel, el de un nivel de escala medio, coincide exactamente con el tamaño del rectángulo de la cuadrícula. La escala correspondiente a este tamaño, 1:50000, es la escala de trabajo habitual con imágenes en las oficinas de la Confederación Hidrográfica del Ebro. El siguiente nivel, el quinto, divide de forma exacta cada imagen del nivel previo en cuatro imágenes, cada una de ellas a escala 1:25000. De la misma forma, otros dos niveles inferiores son construidos, alcanzándose así el séptimo. La aplicación del procedimiento inverso permite la construcción de los niveles superiores.

Para implementar estos procesos, podría haberse utilizado un método de trabajo manual, utilizando, por ejemplo, ArcView. El problema reside en el número de imágenes que finalmente componen el depósito:

Niveles superiores: 1+ 217/(4×4) + 217/4
Nivel intermedio: 217
Niveles inferiores: 217×4 + 217×4×4 + 217×4×4×4
Número total de imágenes ≅ 18600

Se puede apreciar que el proceso manual de generación de las imágenes no es viable, o, al menos, tremendamente costoso. Por ello, se han desarrollado dos pequeños programas software en Java que realizan estas dos tareas automáticamente. El primer programa, el *cutter*, tiene como entrada todas las imágenes y la información de la cuadrícula y genera los niveles comprendidos entre el cuarto y el séptimo, trabajando con una sola imagen original cada vez. El segundo, el *composer*, agrupa las imágenes del cuarto nivel, creando los tres niveles superiores. Los niveles inferiores (5-7) no son procesados por el *composer*, ya que la pérdida de calidad no se aprecia en éstos (1-3).

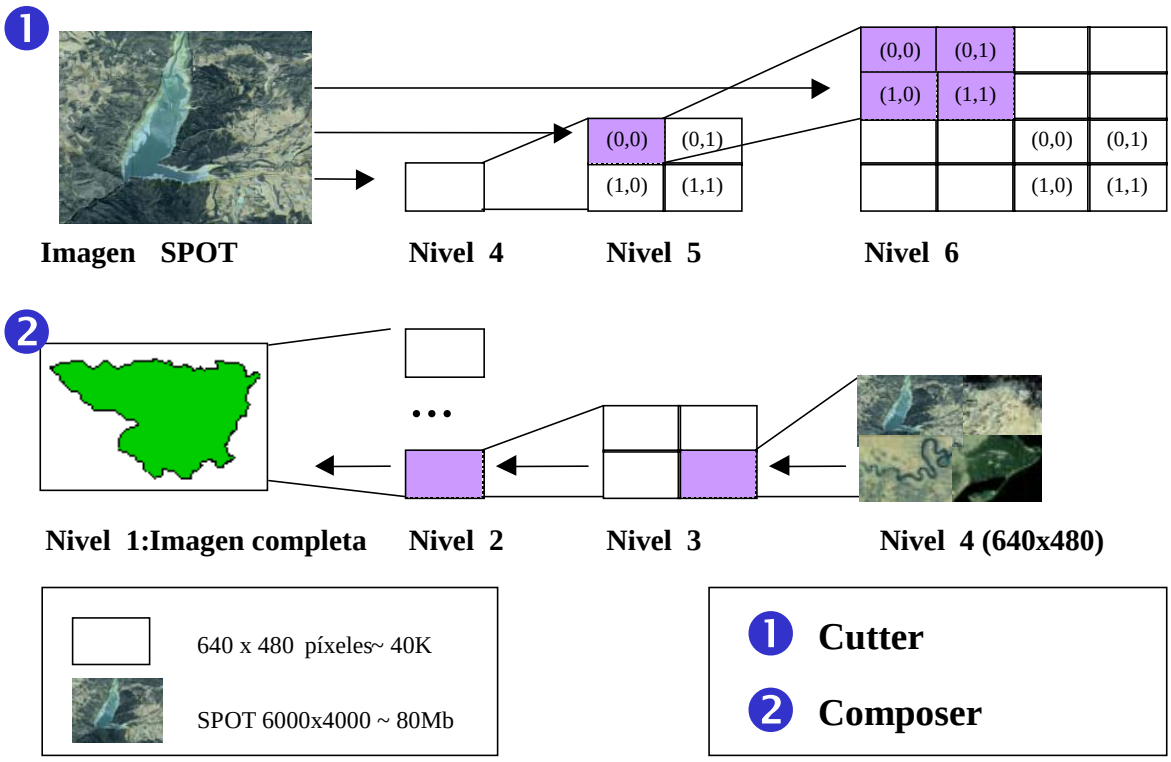

Fig. 2. Proceso de generación del depósito de imágenes

Finalmente, el depósito completo, compuesto por imágenes comprimidas es formato JPEG tiene un tamaño total de 1 Gb. Cada imagen JPEG tiene una dimensión de 640 x 480 píxeles y su tamaño es de unos 40 Kb,

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.119-133. ISBN 84-931933-1-3.

obteniéndose un factor de compresión al utilizar el formato JPEG de 1:25. El tamaño total del depósito de datos no se ve afectado en demasía por la replicación de imágenes que requiere el método, obteniéndose finalmente un relación entre el tamaño del depósito y el original de 1:20 (1 Gb:21 Gb), ligeramente mayor que el ratio de compresión obtenido con el formato JPEG, mejorándose de forma sustancial el tiempo de acceso a las zonas comprendidas en los niveles más altos. Este pequeño incremento del tamaño total se debe a que el número de imágenes decrece de forma cuadrática de un nivel a otro: el séptimo nivel está formado por 18600 imágenes, el sexto por 4750, mientras que el quinto ya sólo tiene unas 1200 imágenes, aproximadamente.

Recuperación de las imágenes

Las imágenes son almacenadas utilizando la estructura de la cuadrícula. Cada imagen se identifica dentro de la cuadrícula por su número de fila y de columna. Por ejemplo, la imagen 5×12 se almacena en un fichero de nombre *h5_12.jpg* (*h* de hidrográfico). En los niveles inferiores se utiliza una nomenclatura similar. Las cuatro partes en que se subdivide la imagen de un nivel se denominan a través de números de columna y fila relativos. La parte noroeste es la imagen 0×0, mientras que la sudeste es la 1×1, de forma que en el ejemplo anterior, a la imagen *h5_12.jpg* del cuarto nivel le corresponderían las imágenes *h5_12_0_0.jpg*, *h5_12_0_1.jpg*, *h5_12_1_0.jpg* y *h5_12_1_1.jpg* del quinto nivel. El nombrado de las imágenes en el resto de los niveles inferiores sigue el mismo criterio.

Una imagen de un nivel superior agrupa las imágenes de su nivel inmediatamente inferior de cuatro en cuatro y se nombra con el número de imagen que corresponde a su imagen noroeste del nivel inferior, añadiéndole un prefijo indicando el nivel al que pertenece la imagen. Continuando con el ejemplo introducido anteriormente, la imagen *hl3_4_12.jpg* pertenece al nivel 3 y se compone de las imágenes *h4_12.jpg*, *h4_13.jpg*, *h5_12.jpg* y *h5_13.jpg* De esta forma, en el nivel tres, los números que forman parte de los nombres de las imágenes son múltiplos de 2, en el segundo nivel son múltiplos de 4 y en primer nivel, de 8, lo que facilita las labores de búsqueda.

El primer paso para la obtención de las imágenes que forman una zona geográfica concreta es el de encontrar el nivel de escala apropiado. El nivel más adecuado será aquél que cubra la zona con el menor número de imágenes pequeñas. El código que sigue corresponde al algoritmo que calcula el nivel de una zona.

```
ViewDimension = Dimension (width, height) of the selected view zone
LevelDimension1 = GridSquareDimension x 2 x 2 x 2
LevelDimension(n) = LevelDimension1 / 2^n

n = 7
while ( (ViewDimension > LevelDimension(n)) || (n >= 1) )
   n = n - 1;
end while
ViewLevel = n;
```

Fig. 3. Algoritmo para la determinación del nivel de escala más adecuado

Paralelamente a la determinación del nivel de escala, se procede a la detección de los rectángulos de la cuadrícula que contienen los cuatro vértices de la zona cuya imagen se desea obtener. Para esta operación se utiliza el mecanismo de búsqueda ofrecido por el módulo de visualización, buscando sobre la capa vectorial que contiene la cuadrícula.

Conocidos el nivel de escala y las posiciones de los vértices, y utilizando las relaciones de tamaño entre distintos niveles, es posible establecer las imágenes del nivel seleccionado que contienen los vértices. Finalmente, sólo es necesario obtener los ficheros que corresponden a las imágenes que se encuentran dentro del rectángulo definido por los cuatro vértices.

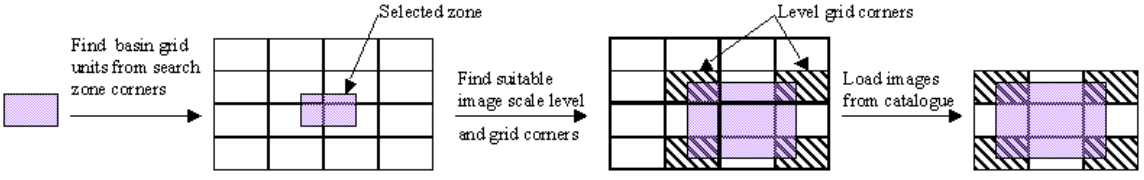

Fig. 4. Pasos para la recuperación de imágenes

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.119-133. ISBN 84-931933-1-3.

Arquitectura del sistema

La arquitectura del sistema, que puede verse en la figura, es básicamente cliente – servidor. El servidor almacena los recursos compartidos (depósito de imágenes y coberturas vectoriales) que utiliza la aplicación. Cada cliente está compuesto por una aplicación desarrollada como applet de Java. Cada uno de los componentes tanto del cliente como del servidor se explican con detalle a continuación.

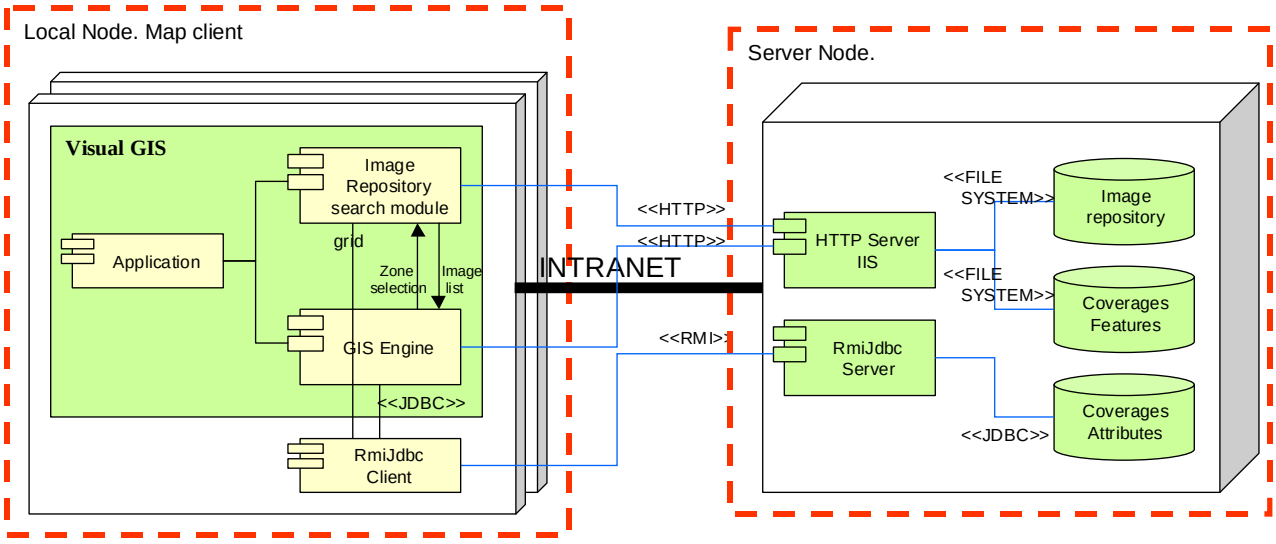

Fig. 5. Arquitectura del sistema

La aplicación principal, Visual GIS, ha sido desarrollada como applet de Java. Un applet es una pequeña aplicación que puede ser lanzada y ejecutada desde un navegador de internet a través de una página html. Este modo de desarrollo de software permite una fácil distribución de las aplicaciones. La aplicación se compone de dos módulos implementados en Java: el módulo de búsqueda de imágenes, encargado de obtener las imágenes de una zona de visualización concreta, y el módulo *GIS Engine*, capaz de mostrar coberturas vectoriales o ráster. Tal y como se ha explicado previamente, el lenguaje de programación en el que se han desarrollado estos módulos ha sido Java, debido a características como su gran portabilidad entre plataformas, a las facilidades que proporciona para el desarrollo de software distribuido, que permiten de forma cómoda la expansión y escalabilidad de la aplicación, y a la librería que ofrece para la gestión de imágenes, que facilita el trabajo de su desarrollo.

El modulo *GIS Engine* ha surgido a partir del fuerte incremento experimentado en las necesidades del mercado geográfico en los últimos años, causado a su vez por el aumento en la capacidad de procesamiento y almacenamiento de los computadores. Actualmente, los sistemas de información modernos incluyen software cuyo objetivo es el de mantener y gestionar información relativa a elementos geográficos. El módulo que hemos desarrollado ofrece herramientas útiles para la gestión de este tipo de información. Muestra datos vectoriales o imágenes georreferenciados e implementa las herramientas básicas y más comunes para trabajo con mapas, como puedan ser el zoom, el desplazamiento o la selección de elementos. El componente también incluye una interfaz gráfica de usuario que permite que otras aplicaciones integren este módulo sin necesidad de implementar toda la interfaz SIG.

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.119-133. ISBN 84-931933-1-3.

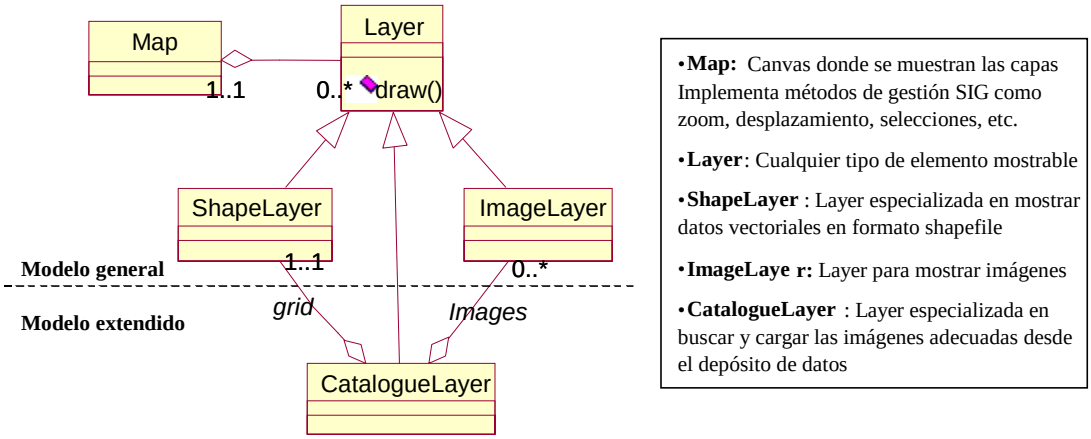

Fig. 6. Diagrama de los objetos principales del módulo de motor SIG

El módulo *GIS Engine* ha sido desarrollado en Java, siguiendo un diseño orientado a objeto que puede ser fácilmente enriquecido y aumentado. Como resultado de este proyecto, algunas de sus funcionalidades fueron mejoradas, mientras que otras fueron añadidas, como, por ejemplo, la de guardado de mapas. El desarrollo de un mecanismo genérico de visualización, costeado por varios proyectos de este tipo [FERNANDEZ99], ha resultado ser fundamental, tanto en lo que a la facilitación del desarrollo de nuevas aplicaciones SIG se refiere, como en lo relativo al coste final. Por ejemplo, MapObjects de ESRI [HART97] es un buen motor de visualización, e incluso podría haber sido adecuado para este proyecto, aunque la existencia de varios proyectos relacionados que han hecho uso de la mismas herramientas SIG básicas ha posibilitado la recuperación de la inversión inicial y la eliminación de la necesidad de pago de licencias de ejecución a terceros.

El módulo de búsqueda de imágenes es el encargado de trabajar con las imágenes. Dada una zona seleccionada, con sus coordenadas y escala, debe localizar y obtener aquellas imágenes del depósito que han de formar la zona seleccionada. El motor de búsqueda en el depósito es un componente local que ha sido integrado en la aplicación SIG. No necesita de ningún servidor intermedio para obtener los nombres de las imágenes, aunque, por supuesto, necesita conocer las reglas de nombrado y la ubicación del depósito.

Como servidor de la aplicación y de las coberturas puede utilizarse cualquier servidor de web corriente, como Apache, JWS, Netscape o ISS de Microsoft, que, finalmente, ha sido nuestra elección. El software de la aplicación se transfiere desde el servidor de la forma habitual, utilizando el mecanismo de html y Java. Es también el propio lenguaje Java el que proporciona los mecanismos para garantizar una ejecución correcta en distintos entornos con diferentes sistemas operativos. Las imágenes se ubican en un directorio accesible desde los clientes. Cada cliente posee un fichero de propiedades que le indica dónde encontrar las imágenes.

El trabajo con información vectorial es algo más complejo. Mientras que en una versión puramente local sería posible utilizar coberturas vectoriales que no estuvieran totalmente cargadas en memoria, trabajando directamente contra el almacenamiento físico, al utilizar servidor http, esta solución no es aplicable, ya que las coberturas vectoriales deberían ser cargadas previamente en memoria, para poder ser mostradas posteriormente por el applet. Para evitar esta carga en memoria, los atributos de los datos vectoriales no van a ser cargados en memoria, sino que cuando se solicitan los atributos de un elemento determinado, como resultado de una operación de selección, se genera una consulta contra la base de datos vectorial (compuesta por los ficheros DBase de los Shapefiles) en la parte del servidor. El servidor de base de datos es, simplemente, un traductor de consultas a bases de datos JDBC de cliente a servidor, a través del protocolo *Java Remote Interface (RMI)*. El nuevo controlador JDBC, denominado *RmiJdbcDriver* ha sido obtenido de GNU. El servidor correspondiente (*RmiJdbd*) también es distribuido por GNU de forma gratuita. [GNU97]

Esta arquitectura permite a una aplicación distribuida utilizar información compartida a través de un simple servidor apenas sobrecargado. Los clientes también obtienen beneficios, debido a que no se requiere almacenamiento local de la información y a que una modificación de la misma puede ser conocida inmediatamente. Uno de los requisitos que debe cumplir la intranet es el de proporcionar una velocidad relativamente alta y una carga no excesiva, de modo que proporcione a los clientes unas condiciones de máxima velocidad de acceso.

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.119-133. ISBN 84-931933-1-3.

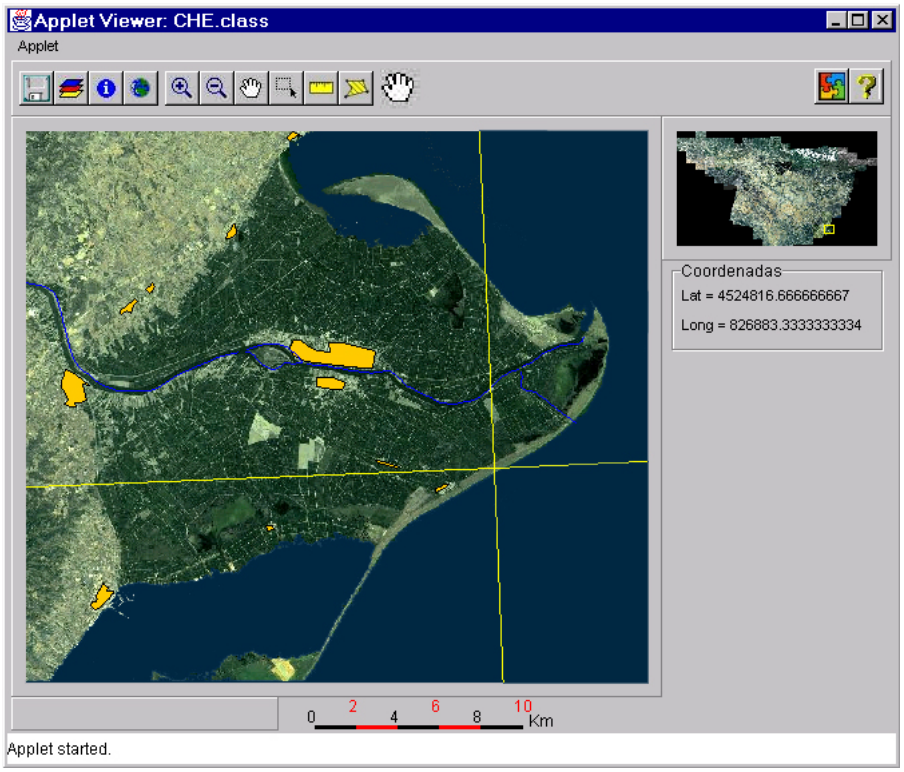

***Fig. 7. Vista general de la aplicación**. El centro de la imagen muestra el delta del Ebro, con coberturas de ríos y núcleos de población. Una cruz amarilla representa los límites de cuatro cuadrantes a escala 1:50000. La parte superior corresponde a la barra de herramientas, mientras que a la derecha puede observarse un mapa general de la cuenca del Ebro, que muestra la zona de visualización seleccionada. Debajo del mapa general, las coordenadas (latitud y longitud) actuales sobre las que se encuentra el ratón.*

Conclusiones

En este artículo se han mostrado las ventajas del uso de Java para la creación de aplicaciones de visualización SIG a bajo coste. Nos hemos beneficiado de la disponibilidad de clases estándares de Java para el trabajo con imágenes, que hemos utilizado para gestionar un gran conjunto de imágenes de gran tamaño. Las capacidades de comunicación de Java han permitido el desarrollo de un conjunto de módulos portable y capaces de coordinar y acceder a imágenes de gran tamaño desde distintos puntos de una intranet. La utilización de Java también permitiría, por ejemplo, escalar la aplicación de forma sencilla, o aumentar la capacidad del sistema para proporcionar acceso por internet. El componente Visual GIS, sin necesidad de efectuar grandes modificaciones, pasaría de ser cliente a formar parte de un servidor de mapas que puede ser accedido por un applet ligero a través del servidor http utilizado. Si el servidor de mapas se construyera de acuerdo a la especificación OpenGIS, el incremento en la disponibilidad de esta información sería sensiblemente mayor, y la rentabilidad de la misma alcanzaría su punto máximo.

En estos momentos, la aplicación está totalmente operativa en el Ministerio de Medio Ambiente, siendo utilizada por todo el personal del mismo a través de navegadores de internet como Netscape o Internet Explorer, satisfaciendo los requisitos referentes a la velocidad de acceso y ofreciendo a los usuarios una completa variedad de herramientas de gestión. La aplicación simplifica el acceso a las imágenes y aumenta también la velocidad de este acceso, ya que previamente sólo estaban disponibles en cederrones, incrementando el uso de esta información por parte del Ministerio, y, por tanto, permitiendo amortizar parte del alto costo de las imágenes de los satélites.

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.119-133. ISBN 84-931933-1-3.

Referencias

[AAC99] G. Andrienko, N. Andrienko, J. Carter. "Thematic Mapping in the Internet: Exploring Census Data with Descartes". *TeleGeo'99, First International Workshop on Telegeoprocessing*. pp. 138--145. Lyon, France. May, 1999.

[BGS99] Tom Barclay, Jim Gray, Don Slutz. "Microsoft TerraServer: A Spatial Data Warehouse". 25th VLDB Conference 31-May-1999.

[BBBCS99] O.T. Balovnev, A. Bergmann, M. Breunig, A.B. Cremers, S. Shumilov. "Remote Access to Active Spatial Data Repositories". *TeleGeo'99, First International Workshop on Telegeoprocessing*. pp. 125--130. Lyon, France. May, 1999.

[CLH99] B. Cambray, C. Leclerc, J.R. Houllier. "Software Architectures Based on Cartographical Products". *TeleGeo'99, First International Workshop on Telegeoprocessing*. pp. 31--39 Lyon, France. May, 1999.

[ERMAP99] "ER Mapper Image Web Server". Earth Resource Mapping Pty Ltd. 1999

[ESRI98] Environmental System Research Institute, *"ESRI Shapefile Technical Description"* ESRI White Paper – July 1998

[FALBZM99] P. Fernández, P. Álvarez, M.A. Latre, R. Béjar, J. Zarazaga, Muro-Medrano "*Sistema de Información Geológico-Minero con capacidad de visualización SIG*" VIII Conferencia Nacional de Usuarios de ESRI 21-Oct-1999

[FNCZM00] P. Fernández, J. Nogueras, O. Cantán, J. Zarazaga, P.R. Muro-Medrano. "*Java Application Architectures to Facilitate Public Access to Large Remote Sensed and Vector Geographic Data*". Telegeo '2000. Second International Symposium on Telegeoprocessing. Sophia Antipolis, pp. 81-92.France, 10 –12 May, 2000.

[GC99] P.P. Gonzalves, M. Costa. "Local and Remote Geoprocessing Applications". *TeleGeo'99, First International Workshop on Telegeoprocessing*. pp. 53--60. Lyon, France. May, 1999.

[GNU97] "*A client/server JDBC Driver based on Java RMI"* http://dyade.inrialpes.fr/mediation/download/RmiJdbc/RmiJdbc.html

[HART97] R. Hartman, "*Focus on GIS Component Software. Featuring ESRI's MapObjects®*". OnWord Press. 1997.

[KP99] D. Kotzinos, P. Prastacos. "GAEA, a Java-based Map Applet". *TeleGeo'99, First International Workshop on Telegeoprocessing*. pp. 131--137. Lyon, France. May, 1999.

[LIMP99] W.F. Limp, "WEB MAPPING". *GEOEurope*. N. 8, pp. 18—22. Dec. 1999.

[OPENGIS99] Open GIS Consortium. "*OpenGIS Web Map Server Interface Specification*." 1999

[SM98] A. Sorokine, I. Merzliakova. "Interactive map applet for illustrative purposes". *Proceedings of the 6th International Symposium on Advances in Geographic Information Systems*. pp. 46—51. 1998.