

## **Capítulo II**

### **Trabajos Relacionados**

En este capítulo se analizan diferentes propuestas de algunos SIG's que requieren de la exportación de datos ya sea para compartirla o para utilizarla en el contexto del Web. Las aplicaciones son tanto comerciales como de investigación.

## **2. Trabajos relacionados**

Existen diversas propuestas de los que debe ser un SIG, sin embargo cada herramienta cuenta con su propio formato de datos, siendo este la forma en que se almacenan los datos para su empleo en un SIG determinado, como lo son los archivos MIF de MapInfo, DGN de AutoDesk o los Shapefiles de ArcView. En algunos SIG se pueden leer formatos de otras aplicaciones y pasarse al propio formato de la herramienta, lo cual no siempre es lo mejor, en algunos casos existe pérdida de información, en otros no es posible y en otros los datos no proporcionan la información que se debería, estos problemas se dan casi siempre cuando se busca realizar el intercambio de datos entre herramientas de diferente casa de software. Como ejemplo podemos citar la necesidad de cambiar de formato un archivo dwg de AutoCad a un formato Shp de ArcView, aquí intervienen factores como la plataforma, ArcView permite pasar del formato dwg al Shp siempre y cuando el ArcView sea versión para PC, de ser versión para estación de trabajo, no es posible realizar dicha conversión. En el primer caso es posible exportar la información requerida sin pérdida de características de la imagen, sin embargo todas las capas se plasman en una sola. Tomando el caso de que el ArcView sea para estación de trabajo, requeriríamos de otra aplicación como lo es AutoDesk, con la cual es posible hacer la exportación de los datos, pero la exportación es deficiente ya que se pierden algunas características y todas las capas las plasma en una sola.

## **2.2 Aplicaciones de Investigación**

A continuación se describen dos proyectos de investigación que trabajan a través de Internet, exportan datos y ejecutan las funciones básicas de un SIG.

### **2.2.1 GAEA, a Java-based map Applet**

GAEA (Geographic Accessories for Efficient Applications)[Kotzinos,99] es un applet que permite visualizar y manipular datos geográficos en línea, es decir, el applet consta de todas las funcionalidades de un SIG (con excepción de la edición de datos) y el trabajo se efectúa del lado del servidor sin afectar al cliente. GAEA fue desarrollado por el “Regional Analysis Division of the Institute of Applied and Computational Mathematics of Forth, Greece” y se preocupa principalmente por cuidar los siguientes aspectos:

1. El producto no debe necesitar la intervención del usuario para ser bajado o instalado (transparencia).
2. El producto debe de correr en cualquier plataforma y navegador con un mínimo de esfuerzo (portabilidad).
3. El producto debe de realizar las operaciones lo más rápido posible sin importar la carga que tenga el servidor (eficiencia).

El Java GAEA Applet es recibido a la computadora del usuario con una capa base del conjunto de datos. Las capas restantes quedan residentes en el servidor y no son recibidas por el usuario hasta que este las solicite. La arquitectura de GAEA se muestra en la figura 2.1. El Applet establece comunicación con el servidor cuando se requiere información adicional de la base de datos, en este caso del lado del servidor existe una aplicación escrita

en Java la cual sirve de interfaz entre el Applet y el RDBMS (Relational Data base Management System). Esta aplicación pasa la petición del usuario al RDBMS y regresa la información del RDBMS al Applet.

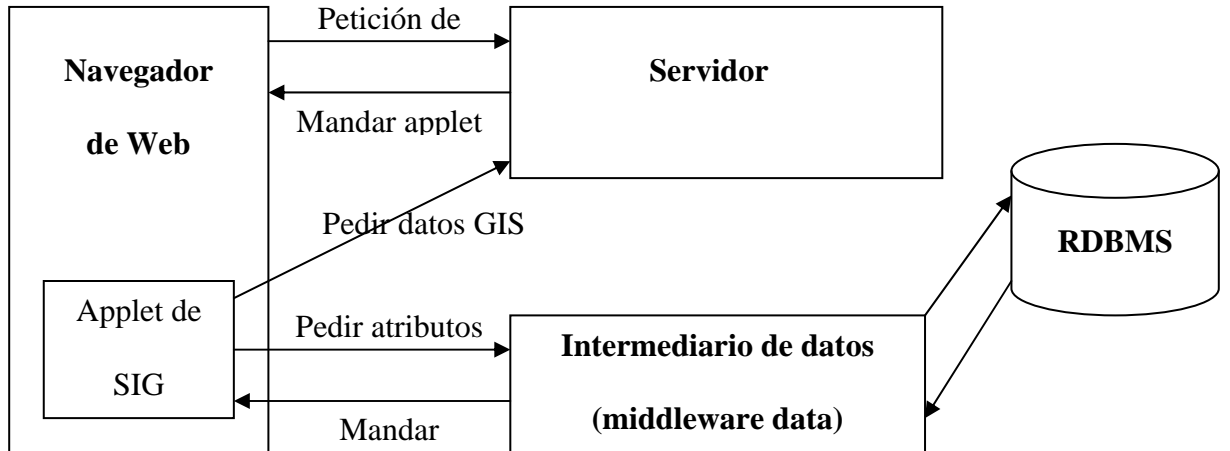


Figura 2.1 Arquitectura de GAEA

Como GAEA esta implementado en Java, puede correr en plataformas de Unix y de Windows NT™ Server. Lo cual no ata a la aplicación a ninguna plataforma en específico, RDBMS u ODBC.

La aplicación que trae las peticiones del usuario al RDBMS y regresa la información al servidor funciona como un intermediario (middleware) entre el applet y el RDBS. Como Java no puede interactuar directamente con el ODBC se emplean los “JDBC drivers” del RDBMS. Por cada requisición del “applet” la aplicación intermediaria crea un nuevo hilo en el servidor y este nuevo hilo se conecta a través de un nuevo “socket” con el “applet” de la máquina del cliente, la siguiente figura muestra el procesamiento de peticiones.

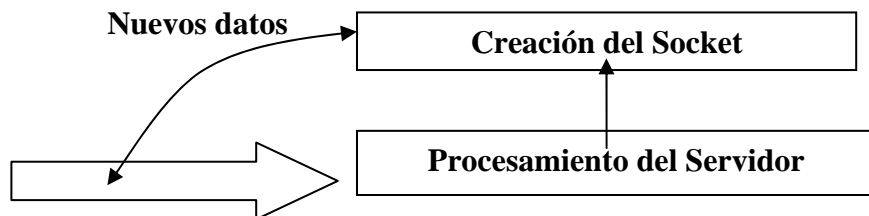


Figura 2.2 Procesamiento de las peticiones del applet a través del intermediario (middleware)

Cuando los resultados de la base de datos son regresados al cliente, el “socket” se cierra y el hilo deja de ejecutarse.

En la base de datos cada capa de información geográfica es almacenada en una tabla separada. Por cada tabla existe también una tabla adicional la cual contiene información de las variables contenidas en las tablas de la base de datos. La información que se guarda de las variables se refiere los atributos de la misma, por ejemplo, a que mapa corresponde, si se le puede realizar consultas o si esta disponible al usuario..

El formato que se emplea para guardar la información es MIF (MapInfo™ Interchange Format). Este formato guarda la información en archivos de texto, cada capa de un proyecto determinado es almacenada en un archivo diferente.

Con respecto a la aplicación, los archivos de datos y descriptivos son empaquetados usando el formato “Java Archive (JAR)”, lo cual reduce el tiempo de recepción del applet ya que aproximadamente se reduce el tamaño del archivo hasta un 40 %. Por otra parte usando el “JAR” se reúnen todos los archivos necesarios y se hace una sola transacción a través del HTTP lo que reduce significativamente el tiempo en que tarda en ejecutarse la aplicación.

Los usuarios pueden ver y manipular vectores de mapas de una cierta área. Después de recibir el “applet” y la capa base los usuarios pueden realizar operaciones como:

1. Zoom In (los usuarios seleccionan el área donde realizar la operación)
2. Zoom Out (los usuarios seleccionan el área donde realizar la operación)
3. Pan (los usuarios seleccionan un nuevo centro para el mapa)
4. Regresar el mapa a su estado inicial

5. Agregar nuevas capas

Al usuario se le permite interactuar con la base de datos (que esta del lado del servidor) y puede:

1. Obtener información de objetos específicos del mapa.
2. Agregar y quitar capas adicionales (tantas como estén disponibles).
3. Personalizar mapas.
4. Consultar y visualizar los resultados de la consulta en el mapa.

Actualmente esta aplicación es usada en varios proyectos

1. Proyecto TEMISIA. GAEA provee las funcionalidades de un SIG a un “web site” que muestra las zonas industriales de Europa y las oportunidades de inversión que existen en cada zona. Los usuarios seleccionan un área determinada y obtienen información sobre ella e inclusive se pueden realizar consultas de acuerdo a los parámetros deseados.
2. Proyecto Condiciones del Tráfico en tiempo real, en este proyecto GAEA esta en etapa de adaptación y por consiguiente quedan problemas por resolver. Se busca establecer comunicación con las bases de datos de tráfico e integrar modelos que permitan predecir carga vehicular, control y manejo de accidentes.
3. Proyecto de Monitoreo de Vehículos de Emergencias Médicas de la ciudad de Heraklio, Creta, Grecia. En este proyecto se recibe la información de los GPS´s instalados en las ambulancias y se proporciona información descriptiva como lo es la severidad del accidente al que se dirige y la disponibilidad de la unidad. Esta información es almacenada en bases de datos de tiempo real. GAEA

consulta estas bases de datos y actualiza la información de los vehículos en la interfaz cada cierto tiempo.

Este proyecto nos mostró la importancia de aspectos como el tamaño del applet que se corre en el cliente y la memoria que consume. También la existencia de un “intermediario de datos” que sirve de enlace entre el applet y la aplicación que corre con el RDBMS y finalmente nos dio a conocer que los archivos MIF son fáciles de exportar debido a que solamente hay que leer archivos de texto sin tener que preocuparse por archivos binarios y el orden de precedencia de los bytes, por consiguiente en un futuro serían fáciles de incorporar al sistema que proponemos.

Podemos resumir sus características en la siguiente tabla.

Características de GAEA	
Lenguaje	Java (applet)
Procesamiento	La operación se realiza en el cliente y en el servidor
Requerimientos de instalación	Ninguno
Operaciones	Acercamientos, alejamientos, paneos, volver al estado inicial la vista del mapa, agregar información
Formato que emplea	MIF (MapInfo Interchange Format)
Nivel en que se encuentra	Aplicación de tres proyectos y prototipo de otros
Otras características	Emplea una aplicación que sirve de enlace entre el applet y el DBMS y emplea "drivers" de jdbc

Tabla 2.1 Características de GAEA

### **2.2.2 Interactive Map Applet for Illustrative Purposes**

El Interactive Map Applet for Illustrative Purposes [Sorokine,98] surgió a raíz de la inconformidad de investigadores cuyo propósito es el de tener en el “Web” mapas de temas específicos. Crearon un prototipo que les permite interactuar fácilmente con los mapas en el Web consultando sus propias bases de datos, sin tener necesidad de instalar ningún software especial.

Primeramente establecieron objetivos específicos para el producto final y pensaron que el producto debía de satisfacer las siguientes características:

1. El producto y los datos no deben de exceder de los 100k a la hora de recibirlo por la red.
2. El producto debe de tener bajos requerimientos de memoria a la hora de ser ejecutado en el cliente.
3. El producto debe tener baja demanda de ancho de banda
4. El producto debe tener un proceso estándar y sencillo para la creación de información
5. El producto debe ser capaz de interpretar los formatos populares de los SIG.
6. El producto debe ser flexible en cuanto a la presentación de sus métodos.

El lenguaje de programación empleado es el JDK 1.1, de tal forma que se puedan aprovechar al máximo sus características de multihilos, sin importar el precio de la memoria que necesite ni su desempeño.

Se realizó el proyecto en dos etapas, la primera consistió en el desarrollo de un script en Perl que prepara los datos para ser leídos por el applet. El script debe ser capaz de leer



diferentes formatos de SIG y debe ser fácilmente extendible, de otra forma se estaría restringido al uso de un determinado software de SIG. Sería ideal, según los autores del artículo [Sorokine,98] que el software no solamente leyera la información de los formatos de SIG más conocidos, sino que también obedeciera a las especificaciones de OpenGis.

La segunda etapa del proyecto consistió en re-escribir el programa final en Java. Para la implementación del applet se tomaron en cuenta los siguientes aspectos:

1. Capacidad de efectuar acercamientos, alejamientos y paneos
2. Activar y desactivar capas
3. Cambiar el estilo de líneas, color y texturas
4. Encontrar características a través de dar un click
5. Despliegue de coordenadas
6. Personalizar la interfaz

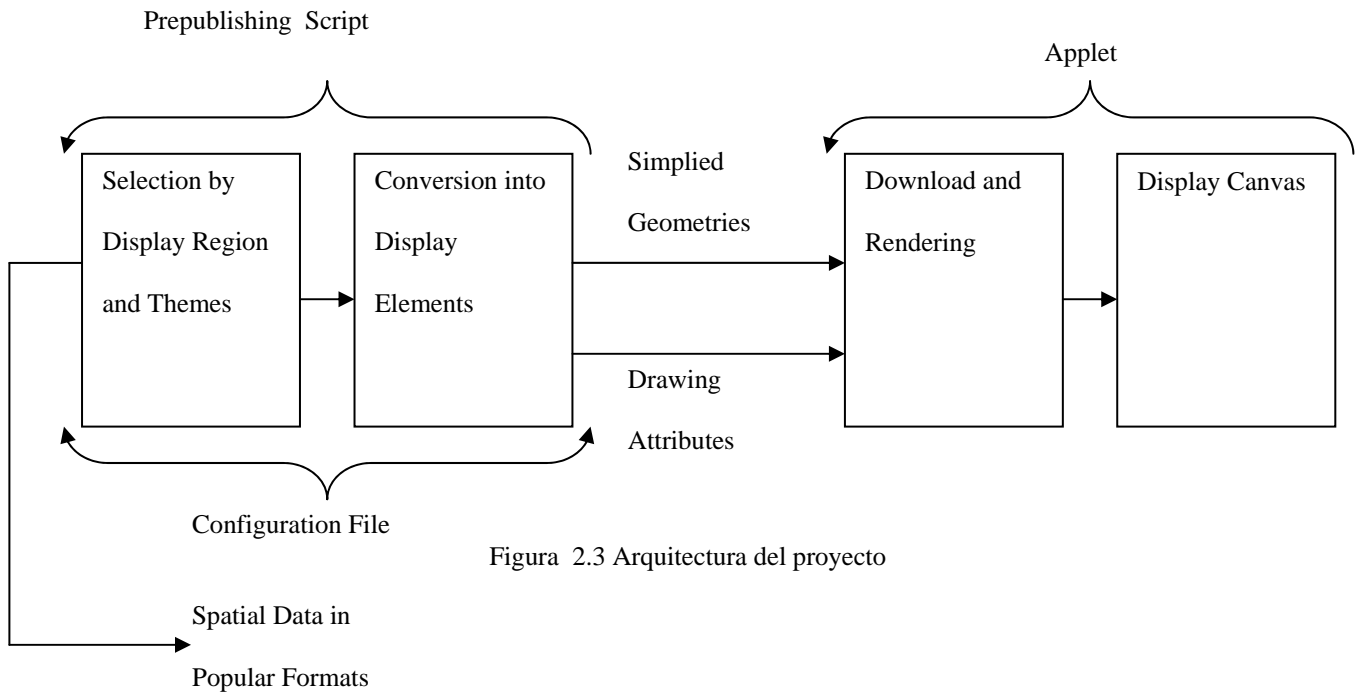


Figura 2.3 Arquitectura del proyecto

Después de que el applet es inicializado, la sesión principal recibe un pequeño archivo de configuración el cual contiene los URLs con las localizaciones de los archivos de datos y otros parámetros de configuración, como lo son el sistema de coordenadas, estilos de líneas y demás características para desplegar el mapa. La siguiente requisición es recibir el subsistema el cual generará diferentes multihilos los cuales recibirán del servidor las geometrías primitivas y los parámetros del dibujo, todo esto será almacenado en la memoria cache. Un hilo de representación (thread) es activado cada vez que es necesario regenerar una imagen. Para repintados de aplanados, la representación (rendering) es hecha en una imagen que esta en memoria, la cual es pintada en la pantalla según los intervalos deseados mientras que el hilo de representación (thread) esta ocupado. Para incrementar el rescalamiento aplanado (smoother rescaling) se emplea la memoria cache, la cual esta organizada en cuadros que son generados por el empleo de un "quadtree". Cada vez que el usuario hace un acercamiento en el mapa se genera un nuevo nivel en el "quadtree" el cual tiene mayor nivel de detalles geométricos.

A diferencia del proyecto anterior, el MapApplet emplea "Quadrees" algoritmos de dibujo, también propone que cualquier SIG sea capaz de interpretar cualquier tipo de archivo, así como propone el empleo del estándar OpenGIS.

Las características de la aplicación se resumen en la siguiente tabla

Características de MapApplet	
Lenguaje	Java (applet) y Perl (scirpt)
Procesamiento	La operación se realiza en el cliente y en el servidor

Características de MapApplet	
Requerimientos de instalación	Ninguno
Operaciones	Acercamientos, alejamientos y paneos
Formato geográfico que emplea	Vectorial ( no especifica cual )
Nivel en que se encuentra	Prototipo
Otras características	Emplea Quadrees y algoritmos de visualización

**Tabla 2.2 Características de MapApplet**

## **2.3 Aplicaciones Comerciales**

Actualmente en el mercado existen diversas aplicaciones, pero solamente haremos énfasis en las aplicaciones más populares para el Web y que comparten datos con otras aplicaciones.

### **2.3.1 MapObjects Internet Map Server (MapObjects IMS) by ESRI**

Este producto es una extensión de MapObjects [Plewe,97], el cual es una colección de componentes para construir mapas y aplicaciones de SIG usando toda la funcionalidad de MapObjects (el cual incluye gran parte de la funcionalidad de ARC/INFO). Este aplicación esta basada en controles de ActiveX siendo esta su gran desventaja pues solamente puede correr en ambientes que soporten los controles de ActiveX.

Esta aplicación lee todos los formatos de archivos que maneja ESRI, así como imagenes en TIFF, BMP y otros formatos. También puede generar imágenes como GIF y JPEG.

### **2.3.2 ArcView Internet Map Server (AIMS) de ESRI**

El nombre de este producto es similar al de MapObjects [Plewe,97], producto de la misma compañía (antes descrito), pero con un enfoque totalmente diferente. AIMS esta diseñado para ser un programa fácil de usar y por consiguiente para un mayor mercado, pero es menos flexible que MapObjects IMS, debido a que el antes mencionado posee mayores recursos.

Cuando AIMS recibe una petición esta es enviada a ArcView y un script de Avenue comienza a correr, dicho script está diseñado para correr solamente un número limitado de aplicaciones en línea. Este script es virtualmente idéntico a cualquier otro script de Avenue,

excepto por que no existe ningún GUI (Geographical User Interface) como tal, en lugar de este los mapas son dibujados en una “nueva ventana virtual” la cual devuelve las imagenes en un formato raster al cliente.y mostrarlos en una página de HTML.

Para incrementar las capacidades de AIMS, ESRI creo un applet de Java llamado “MapCafé” el cual puede ser configurado de acuerdo a las necesidades. El applet puede ser opcional pues se pueden usar solamente las páginas de HTML.

Este producto es actualmente un módulo de extensión de ArcView, no es un programa diferente. Toda la configuración es tomada del ArcView que corre en la computadora del servidor de Web lo cual nos permite tener nuestro mismo ambiente.

#### **2.3.4 MapInfo ProServer by MapInfo**

MapInfo proserver[Plewe,97] es una solución robusta para exportar todo el conjunto de datos de MapInfo así como otros tipos de datos de otras bases de datos. Este producto no es un puente (gateway) pero se comunica mediante un CGI, el cual es responsable de pasar las consultas al MapInfo para que las procese.

Esta entrega extra de nivel de datos permite al ProServer ser usado en otras aplicaciones de Intranet a parte de las del Web. Una vez recibida la petición de consultas, se activa el script MapBasic de MapInfo, diseñado especialmente para esta aplicación, el cual tiene acceso a todos los recursos de MapInfo.

Esta aplicación solamente funciona en Windows 95/NT y puede generar imágenes raster en formato GIF.

## 2.4 Comparación de trabajos relacionados

En la siguiente tabla se muestra un cuadro sinóptico de las principales características de las aplicaciones mostradas en este capítulo.

	<b>GAEA</b>	<b>Map Applet</b>	<b>Map Objects IMS</b>	<b>ArcView IMS</b>	<b>Map Info</b>	<b>UDLASIG*</b>
Java	Si	Si		Si		Si
ActiveX			Si		Si	No
Requiere instalación	No	No	Si	Si	Si	No
Formato propietario	Si		Si	Si	Si	No
Zooms			Si	Si	Si	Si
Pan			Si	Si	Si	Si
Despliegue de coordenadas			Si	Si		Si
DBMS Relacional	Si					Objeto- relacional
Archivos			Si	Si	Si	No
2D	Si	Si	Si	Si		Si
Procesamiento En el servidor			Si	Si	Si	Si
Procesamiento en el cliente	Si	Si				Si
Facilidad de expansión	Si	Si	Si	Si		Si
Volumen de usuarios	Bajo	Bajo	Bajo	Alto		Bajo

Tabla 2.3 Comparación de trabajos relacionados.

\* es el proyecto UDLASIG ya con las características que proponemos

## **2.5 Conclusión**

Después de ver las diferentes aplicaciones existentes podemos decir, que una aplicación de SIG debe de ser extendible, no debe necesitar instalación, debe correr en cualquier plataforma, debe requerir poca memoria y debe permitir intercambiar información con otros sistemas.