

**PRACTICA EMPRESARIAL EN TYT LTDA**  
**TECNOLOGIAS DE INFORMACION Y TELECOMUNICACIONES**

**HOLMAND DARIO VILLALBA BARAJAS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**  
**FACULTAD DE INGENIERÍAS FISICOMECHANICAS**  
**ESCUELA DE INGENIERÍA DE SISTEMAS**  
**BUCARAMANGA**

**2006**

**PRACTICA EMPRESARIAL EN TYT LTDA**  
**TECNOLOGIAS DE INFORMACION Y TELECOMUNICACIONES**

**HOLMAND DARIO VILLALBA BARAJAS**  
**CÓDIGO 1991183**

**Trabajo de grado para obtener el título de Ingeniero de Sistemas**

**JOSÉ CÁRCAMO SEPÚLVEDA**

**Director Práctica Empresarial**  
**Docente UIS**

**OSCAR GONZÁLEZ PIMENTEL**

**Tutor Práctica Empresarial**  
**Gerente TYT Ltda**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**  
**FACULTAD DE INGENIERÍAS FISICOMECHANICAS**  
**ESCUELA DE INGENIERÍA DE SISTEMAS**  
**BUCARAMANGA**

**2006**

**TITULO:** PRÁCTICA EMPRESARIAL EN TECNOLOGÍAS DE INFORMACIÓN Y TELECOMUNICACIONES\*

**AUTOR:** HOLMAND DARIO VILLALBA BARAJAS\*\*

**Palabras claves:** SIG, MapServer, MapObjects, Servicio al Cliente

## RESUMEN

Tecnologías de la información y telecomunicaciones es una empresa desarrolladora de software que actualmente se encuentra interesada en la generación de proyectos aplicando Sistemas de Información Geográfica (SIG); por esta razón se decidió vincular a un estudiante en práctica empresarial que se encargara de realizar el estudio de herramientas de desarrollo para aplicaciones SIG.

Durante la práctica empresarial, se estudiaron algunas herramientas de desarrollo de las cuales se seleccionaron dos para profundizar en su estudio, estas fueron Mapserver y Mapobjects. Sobre cada una de ellas se realizaron pruebas mediante el desarrollo de pequeñas aplicaciones para observar sus funcionalidades gráficas y de consulta; de esta manera se observaron los alcances que tienen estos componentes

En el último mes de la práctica empresarial, se dejó un poco al lado el tema relacionado con los SIG, y se comenzó a trabajar en el desarrollo de la fase de inicio del módulo de Servicio al Cliente para empresas de acueducto, el cual hace parte de un proyecto que se desarrolla en La Empresa. La fase de inicio del módulo de Servicio al Cliente, se basó en la metodología RUP y las actividades que se realizaron corresponden al modelado del negocio y lo requisitos. Para ello, se participaron de charlas por parte de los Ingenieros de TYT Ltda., se hizo la lectura y análisis de documentos relacionados al módulo y a empresas de acueducto, se visitaron sitios web de empresas de este tipo, y participó en charlas con expertos en el tema. Al final de éste último período de la práctica se obtuvo un documento de especificaciones en el cual se plantean los requerimientos para realizar el módulo, y una lista inicial de casos de uso.

\* Trabajo de grado

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.  
Universidad industrial de Santander.  
Director. José Cárcamo Sepúlveda  
Tutor: ingeniero Oscar González Pimentel

**TITLE:** ENTERPRISE PRACTICE IN TYT LTDA \*

**AUTHOR:** HOLMAND DARÍO VILLALBA BARAJAS\*\*

**KEY WORDS:** GIS, MapServer, MapObjects, Customer Service

### **ABSTRACT**

Information and Telecommunications Technologies is a software developer company, actually it's interested in projects's develop applying Geographic Information Systems (GIS); therefore it was decided to tie together a predegree student in enterprise practice to make the research of development tools for applications SIG.

During the enterprise practice, some development tools studied of which two were selected to deepen in their study, these were Mapservr and Mapobjects. On each one of them tests by means of the development of small applications to observe their graphical and queries functionalities were made; this way the significance that have these components were observed

In the enterprise practice last month, the subject related to the SIG was left alongside, and it was begun to work in the development of Customer Service module initial phase for aqueduct companies, which are project component that is developed in TYT Ltda. The Customer Service module initial phase was based on RUP methodology and the activities that were made correspond to business modeling and requirements. For it, TYT Ltda's Engineers perform conference about the module, took the reading and analysis of documents related to module and aqueduct companies, Web sites of companies of this type were visited, and participated in conference them with experts in the subject.

\* Graduation Work

\*\* Physical – Mechanical Engineering's Faculty  
System Engineering and Computer science school. UIS  
Director. José Cárcamo Sepúlveda  
Tutor: Ing. Oscar González Pimentel.

## **DEDICATORIA**

A DIOS todo poderoso por darme la vida y colocar en mi camino a todas las personas que con su ayuda hicieron posible este sueño.

A Luz Stella Barajas Cáceres, mi madre, por su amor, colaboración, paciencia, confianza, apoyo y porque con su esfuerzo y dedicación hizo posible que lograra culminar con éxito esta gran etapa de mi vida.

A Natalia mi abuela que siempre estuvo ahí para apoyarme y darme ánimo en los momentos difíciles por los que pase.

## **AGRADECIMIENTOS**

A Oscar González Pimentel por darme la oportunidad y confianza de realizar la práctica empresarial en su empresa. A todos los socios de TyT por compartir su experiencia y conocimiento.

A Diana Jimena, por su amor, ternura, comprensión, colaboración y porque con su apoyo logre sacar adelante mi carrera.

A mi papá, mi abuela ana, a mis hermanos que siempre estuvieron ahí con sus consejos.

A Julio Cesar, Linderman, Jason y todos mis compañeros de la universidad por su colaboración durante la carrera y compartir momentos agradables.

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	12
<b>1 DESCRIPCIÓN DE LA PRACTICA EMPRESARIAL.....</b>	<b>14</b>
1.1 DESCRIPCION DE LA EMPRESA.....	14
1.1.1 Nombre de la empresa.....	14
1.1.1 Quiénes somos .....	14
1.1.2 Misión de la empresa .....	15
1.1.3 Visión de la empresa.....	15
1.1.4 Valores corporativos.....	15
1.2 DESCRIPCIÓN DEL PROYECTO.....	16
1.2.1 Identificación del problema.....	16
1.2.2 Objetivos .....	17
1.2.2.1 General.....	17
1.2.2.2 Específicos .....	17
1.2.3 Justificación.....	18
<b>2 MARCO TEÓRICO - CONCEPTUAL.....</b>	<b>19</b>
2.1 EL SOFTWARE LIBRE .....	19
2.1.1 Libertades del Software Libre.....	20
2.1.2 Tipos de Licencias.....	20
2.2 SISTEMAS DE INFORMACIÓN GEOGRÁFICA .....	21
2.3 HERRAMIENTAS DE DESARROLLO SIG .....	22
2.3.1 GEOSERVER.....	22
2.3.2 MAPLAB.....	22
2.3.3 SAGA .....	23
2.3.4 MAPSERVER.....	24
2.3.4.1 El CGI de MapServer.....	26
2.3.4.2 jBOX .....	28

2.3.4.3	Componentes de una aplicación MapServer .....	29
2.3.4.3.1	El archivo de inicialización.....	29
2.3.4.3.2	El Mapfile.....	30
2.3.4.3.3	El Template File .....	32
2.3.4.3.4	El Conjunto de Datos SIG .....	33
2.3.5	MAP OBJECTS .....	34
2.3.5.1	MapObjects Windows Edition .....	34
2.3.5.1.1	Principales características.....	35
2.4	REVISIÓN DE LA METODOLOGÍA DE DESARROLLO RUP Y EL LENGUAJE UML.....	35
2.5	ESTUDIO DE LA HERRAMIENTA CASE DE ORACLE.....	38
<b>3</b>	<b>METODOLOGIA .....</b>	<b>42</b>
<b>4</b>	<b>APLICACIONES PARA EL DESARROLLO DE HERRAMIENTAS SIG ...</b>	<b>43</b>
4.1	APLICACIÓN CON MAPSERVER .....	43
4.1.1	Visualización de capas con MapServer.....	43
4.1.2	Manejo de facilidades gráficas .....	47
4.1.2.1	Implementación del applet jBox.....	47
4.1.2.2	Implementación del Zoom In.....	48
4.1.2.3	Implementación del Zoom Out.....	48
4.1.2.4	Implementación del Pan .....	48
4.1.3	Manejo de atributos en la visualización de una mapa .....	49
4.1.3.1	El mapa de referencia.....	49
4.1.3.2	La Leyenda.....	50
4.1.3.3	La escala del mapa.....	52
4.1.3.4	Las etiquetas de un mapa.....	53
4.1.3.5	Simbología para los mapas de puntos.....	54
4.1.4	Consultas de información geográfica con MapServer .....	55
4.1.4.1	Consultas por selección espacial.....	57
4.1.4.2	Consultas por atributos.....	57
4.2	APLICACIÓN CON MAPOBJECTS.....	58
4.2.1	Manejo de facilidades gráficas .....	59

4.2.1.1	Función para la implementación del zoom In.....	59
4.2.1.2	Función para la implementación del Zoom Out .....	59
4.2.1.3	Función para la implementación del pan .....	60
4.2.1.4	Función para la implementación del Full Extent .....	60
4.2.1.5	Manejo de etiquetas .....	61
4.2.2	Consultas de información geográfica con MapObjects .....	61
4.2.2.1	El objeto Recordset .....	61
4.2.3	Inserción de puntos en una capa vectorial .....	64
<b>5</b>	<b>ACTIVIDADES REALIZADAS EN LA PRIMERA FASE DE DESARROLLO</b>	
	<b>DEL MÓDULO DE SERVICIO AL CLIENTE.....</b>	<b>66</b>
5.1	PRESENTACIÓN INTRODUCTORIA SOBRE EL FUNCIONAMIENTO	
	DEL MÓDULO SERVICIO AL CLIENTE .....	68
5.2	REALIZACIÓN DE UNA LISTA DE CASOS DE USO PARA EL	
	MÓDULO DE SERVICIO AL CLIENTE .....	69
	CONCLUSIONES.....	70
	RECOMENDACIONES .....	72
	REFERENCIAS BIBLIOGRAFICAS .....	73
	ANEXO A .....	75
	ANEXO B .....	80
	ANEXO C .....	87

## LISTA DE FIGURAS

<b>Ilustración 1: Interfaz gráfica de saga .....</b>	<b>24</b>
<b>Ilustración 2: Vista general de RUP .....</b>	<b>37</b>
<b>Ilustración 3: Repositorio de información de Oracle Designer .....</b>	<b>39</b>
<b>Ilustración 4: Flujo de información de análisis a diseño .....</b>	<b>40</b>
<b>Ilustración 5: Visualización de capas .....</b>	<b>44</b>
<b>Ilustración 6: Ejemplo de una estructura jerárquica de MapFile.....</b>	<b>46</b>
<b>Ilustración 7: Mapa de Referencia .....</b>	<b>50</b>
<b>Ilustración 8: Leyenda del mapa .....</b>	<b>51</b>
<b>Ilustración 9: Escala del Mapa .....</b>	<b>53</b>
<b>Ilustración 10: Labels.....</b>	<b>54</b>
<b>Ilustración 11: Creación de un símbolo .....</b>	<b>55</b>
<b>Ilustración 12: Zoom In .....</b>	<b>59</b>
<b>Ilustración 13: Pan .....</b>	<b>60</b>
<b>Ilustración 14: Etiquetas .....</b>	<b>61</b>
<b>Ilustración 15: Objeto Recordset .....</b>	<b>62</b>
<b>Ilustración 16: Objeto Fields .....</b>	<b>63</b>
<b>Ilustración 17: Fases de la metodología RUP .....</b>	<b>67</b>

## INTRODUCCIÓN

En la década de los setenta, con el desarrollo de la tecnología informática, aparecieron una serie de tecnologías cuya finalidad era gestionar datos espaciales georreferenciados<sup>1</sup>. Al inicio se necesitaba un potente instrumental para poder trabajar con ellos; pero poco a poco se fueron desarrollando mejoras técnicas que han ido simplificando y popularizando la utilización de este tipo de programas. Algunos autores han llegado a afirmar que “los Sistemas de Información Geográfica son el paso adelante más importante desde la invención del mapa” (CHORLEY, 1987). No se sabe si realmente este nuevo avance es tan crucial pero, sin duda, se trata de una interesante y útil herramienta que facilita la compilación, análisis y divulgación de los datos geográficos.

El desarrollo de los SIG ha corrido paralelo al progreso del *hardware* y del *software* informático. Los avances en la tecnología de los ordenadores personales (PC) se han visto correspondidos con unos Sistemas de Información Geográfica más potentes y fáciles de manejar (CASSETARI, 1993).

TyT Ltda es una empresa dedicada al desarrollo de software y a la incursión de nuevas tecnologías en el área de la información y las telecomunicaciones; por tal motivo ha querido incursionar en el campo de los Sistemas de Información Geográfica; de ahí nace la necesidad de tener un estudiante en práctica que profundice en los temas relacionados con este campo.

Actualmente existen en el mercado diversas herramientas para el desarrollo de aplicaciones SIG ya sean herramientas de tipo comercial o herramientas de software libre las cuales son muy importantes a la hora de pensar en el desarrollo

---

<sup>1</sup> Posicionamiento en el que se define la localización de un objeto espacial en un sistema de coordenadas determinado o procedente de una proyección cartográfica (Universal Transversal de Mercator, Lambert, etc.).

de aplicaciones basadas en sistemas de información geográfica debido a la robustez con la que cuentan dichas herramientas.

## **1 DESCRIPCIÓN DE LA PRACTICA EMPRESARIAL**

### **1.1 DESCRIPCION DE LA EMPRESA**

#### **1.1.1 Nombre de la empresa**

Tecnologías de Información y Telecomunicaciones TyT Ltda

#### **1.1.1 Quiénes somos**

En el entorno económico actual, altamente exigente y turbulento, el futuro de las empresas no está sólo ligado a la calidad y la eficiencia en costes, factores necesarios pero no suficientes, sino también, y principalmente, en la búsqueda constante de una diferenciación que aporte valor a los Clientes, Accionistas, y la Sociedad.

La innovación, y en especial la innovación tecnológica, es la herramienta para ello. En este sentido la innovación y la aplicación e implementación de la tecnología ha dejado de ser una opción para pasar a ser una necesidad. No encarar el reto de la innovación significaría condenar la empresa al estancamiento y, en el peor de los casos, al declive. Concientes de este nuevo paradigma, TyT Ltda. ha estructurado un nuevo concepto empresarial basado en el diseño, implementación , ejecución , monitoreo, seguimiento, movilidad, reestructuración y adaptabilidad de soluciones empresariales soportadas en las tecnologías y sistemas de información bajo una estructura dinámica e integral que permita la satisfacción de las necesidades globales y específicas de nuestros clientes.

Nuestra historia es el presente, somos una compañía de base tecnológica que nace del espíritu emprendedor de un grupo interdisciplinario de personas con amplios conocimientos y experiencia en Sectores y actividades como las tecnologías de información, la gestión tecnológica y la consultoría estratégica

empresarial que durante años aportaron y agregaron valor a las empresas en donde prestaron sus servicios, adquiriendo experiencia y las competencias necesarias para transformar deseos, visiones y proyectos individuales en un pensamiento corporativo basado en la dimensionalidad de tres pilares fundamentales: Talento Humano ( socios , empleados, aliados), Clientes y la Sociedad, permitiendo estructurar productos y servicios que generen valor a nuestros clientes, prácticas que generen riqueza a nuestro Talento Humano y principios que contribuyan a construir una sociedad más abierta y con mejor calidad de vida.

### **1.1.2 Misión de la empresa**

Apoyar las estrategias de negocio y el modelo organizacional de nuestros clientes, suministrando soluciones integrales en tecnologías de información, basados en la experiencia y conocimiento de nuestro talento humano, generando así valor para los Accionistas.

### **1.1.3 Visión de la empresa**

Ser para el 2007 un proveedor de soluciones en Tecnologías de información con proyección nacional internacional convirtiéndonos en aliados estratégicos para nuestros clientes con respuestas dinámicas, ágiles e innovadoras incrementando su productividad y competitividad.

### **1.1.4 Valores corporativos**

TyT Ltda da a conocer sus Valores Corporativos para asegurar que todos los ejecutivos, empleados y clientes, comparten los valores que nos llevan a demarcar la excelencia en nuestra actividad empresarial:

1. El compromiso con los clientes, actuar con iniciativa y disciplina para ayudarlos a lograr sus objetivos empresariales y sociales.

2. El conocimiento, tomar decisiones acertadas para crear nuevas ventajas empresariales y abrir oportunidades específicas de negocios que permiten demarcar el liderazgo empresarial.
3. La innovación, aceptar la variedad y las diferencias para de esta forma interpretarlas y apropiarlas con agilidad para sobre esta tomar acciones encaminadas al logro de objetivos.
4. La responsabilidad, actuar de forma coherente con la economía, el desarrollo empresarial, la tecnología, la sociedad y la política, para de esta forma enmarcar la operación en una estructura logística que beneficie a todos los actores que hacen parte de nuestro entorno.
5. La determinación, actuar con pasión, confianza, iniciativa y agilidad para lograr en el momento oportuno los resultados empresariales tanto de nuestros clientes como de nuestra empresa.

## **1.2 DESCRIPCIÓN DEL PROYECTO**

### **1.2.1 Identificación del problema**

TYT es una empresa desarrolladora de software que está interesada en la generación de proyectos aplicando sistemas de información geográfica, a un bajo costo pero de buena calidad, debido a esto desea investigar a fondo el software libre para SIG que existe actualmente.

Para comenzar a hacer un desarrollo y ofrecer determinado producto es necesario conocer los alcances y limitaciones que tienen las herramientas que se van a utilizar. El software libre puede tener alguna clase de limitaciones, de igual manera se hace necesario el estudio de algún componente de desarrollo SIG de tipo comercial, que permita hacer una comparación entre las herramientas de software libre y las que se encuentran bajo alguna licencia privada.

Una herramienta SIG debe tener funcionalidades básicas, que, además de la visualización, permita conocer y utilizar la información geográfica y atributos asociados a los mapas; de esta manera se plantea la posibilidad de investigar cómo se realizan las consultas sobre las capas que contienen los mapas vectoriales.

Antes de iniciar el desarrollo de la práctica, se cuenta con conocimientos previos sobre conceptos de SIG, los tipos de mapas que se pueden manejar y algunas herramientas de desarrollo que fueron estudiadas durante el transcurso de la carrera.

## **1.2.2 Objetivos**

### **1.2.2.1 General**

Revisar el estado del arte de los componentes de software libre existentes en el mercado para el manejo de Sistemas de Información Geográfica, indicando su forma de utilización, alcances y limitaciones.

### **1.2.2.2 Específicos**

1. Investigar que componentes de software libre son utilizados en los sistemas de información geográfica en la arquitectura Web y en ambiente Windows.
2. Investigar las facilidades de consultas geográficas, manejo espacial que puedan prestar estos componentes en la realización de proyectos SIG.
3. Estudiar los alcances más importantes de los componentes de software que se investiguen en el desarrollo de proyectos SIG.
4. Estructurar requerimientos ya levantados para adecuarlos a la metodología de desarrollo de software para el módulo de Servicio al Cliente.

5. Hacer pruebas y documentar las funcionalidades encontradas en los componentes de software, para dejar una base sólida en la realización de futuros proyectos SIG en TyT Ltda.

### **1.2.3 Justificación**

En la actualidad existen diversos componentes para el desarrollo de sistemas de información geográfica, los cuales están tomando cada vez más importancia para las empresas<sup>2</sup> en la realización de aplicaciones que hagan uso de datos espaciales, por eso es necesario hacer un estudio de dichos componentes en cuanto a funcionalidad y calidad.

El uso de componentes de software libre permite a la empresa ofrecer productos económicamente más competentes dentro del mercado de los Sistemas de Información Geográfica.

Por tal motivo Tecnologías de la Información y Telecomunicaciones optó por vincular a un estudiante en modalidad de práctica empresarial para que cumpliera con esta labor de profundización sobre el tema y utilizar toda la información recopilada en futuros proyectos basados en los sistemas de información geográfica.

---

<sup>2</sup> Empresas desarrolladoras de software.

## 2 MARCO TEÓRICO - CONCEPTUAL

Actualmente se encuentran en el mercado diversas herramientas y componentes<sup>3</sup> para desarrollar aplicaciones de Sistemas de Información Geográfica, muchas de estas de software libre y otras con licencia comercial, siendo las primeras el objetivo primordial de este estudio, pero sin dejar al lado la posibilidad de analizar alguna herramienta o componente de carácter comercial.

Durante el transcurso de la práctica empresarial se revisó qué herramientas o componentes de desarrollo de aplicaciones SIG existen actualmente en el mercado que tengan las funcionalidades básicas de manejo gráfico de mapas y consultas de información sobre los mapas. Se encontró varias herramientas de este tipo, sin embargo no todas cumplían las expectativas necesarias para profundizar sobre ellas.

De este estudio preliminar se obtuvo dos herramientas que contaban con las funcionalidades necesarias como herramientas de desarrollo para la realización de un proyecto SIG; estas herramientas son MapServer la cual sirve para el desarrollo de aplicaciones SIG en ambiente Web y MapObjects que sirve para el desarrollo de aplicaciones en ambiente Windows.

### 2.1 EL SOFTWARE LIBRE

Software libre es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente. Análogamente, el software gratuito (denominado usualmente Freeware) incluye en algunas ocasiones el código fuente; sin embargo, este tipo de software *no es libre* en el mismo sentido

---

<sup>3</sup> Las herramientas se utilizan para el desarrollo de aplicaciones y hacen uso de los componentes para agregar nuevas funcionalidades a las mismas.

que el software libre, al menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

### **2.1.1 Libertades del Software Libre**

De acuerdo con tal definición, el software es "libre" si garantiza las siguientes libertades:

- "libertad 0", ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.)
- "libertad 1", estudiar y modificar el programa (para lo cual es necesario poder acceder al código fuente)
- "libertad 2", copiar el programa de manera que se pueda ayudar al vecino o a cualquiera
- "libertad 3", mejorar el programa, y hacer públicas las mejoras, de forma que se beneficie toda la comunidad.

### **2.1.2 Tipos de Licencias**

Existen distintas variantes del concepto de software libre o grupos de licencias, por ejemplo:

- Las libertades definidas anteriormente están protegidas por licencias de software libre, de las cuales una de las más utilizadas es la *Licencia Pública General GNU* (GPL). El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que no sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL.

- Software bajo el dominio público, sobre el cual el autor ha abandonado sus derechos de autor. El software bajo el dominio público, puesto que carece de protección de copyright alguna, puede ser incorporado libremente tanto en software libre como no libre.
- Licencias estilo BSD<sup>4</sup>, llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles.

MapServer cumple con las cuatro libertades del software libre y según el tipo de licencias se puede catalogar dentro de la Licencia Pública General (GPL).

## 2.2 SISTEMAS DE INFORMACIÓN GEOGRÁFICA

Un Sistema de Información Geográfica es una **tecnología de manejo de información geográfica** formada por equipos electrónicos (**hardware**) programados adecuadamente (**software**) que permiten manejar una serie de **datos** espaciales (información geográfica) y realizar análisis complejos con éstos siguiendo los criterios impuestos por el equipo científico (**personal**).<sup>5</sup>

No se debe confundir la definición de un sistema de información geográfica con la de una herramienta SIG ya que la herramienta SIG es solo una parte de todo el sistema la cual nos ayuda a realizar el análisis de los datos espaciales.

---

<sup>4</sup> Berkeley Software Distribution

<sup>5</sup> Definición tomada de la página [www.gabrielortiz.com](http://www.gabrielortiz.com)

De acá en adelante se analizará a Mapserver y MapObjects que sirven para el desarrollo de herramientas o aplicaciones SIG.

## **2.3 HERRAMIENTAS DE DESARROLLO SIG**

- En esta sección se mencionarán las herramientas y componentes que fueron objeto de estudio en la primera fase de la práctica empresarial.

### **2.3.1 GEOSERVER**

Es una herramienta desarrollada en java para implementar un servidor completamente funcional con capacidades WFS-T<sup>6</sup> y servicio WMS<sup>7</sup>.

Con GeoServer se puede publicar mapas e imágenes; los usuarios pueden actualizar, borrar y modificar la información geográfica.

Sus principales características son:

- Completamente compatible a las especificaciones WMS y WFS.
- Contiene soporte para bases de datos como Postgres, ArcSDE, DB2 y Oracle.
- Genera mapas Web con formatos de salida como jpeg, gif, png, svg y KML.
- Utiliza otros recursos de código abierto en gran medida GeoTools que son un conjunto de herramientas la cuales sirven para desarrollar aplicaciones espaciales.

### **2.3.2 MAPLAB**

MapLab es un conjunto de herramientas que permiten crear y administrar aplicaciones cartográficas basadas en MapServer; se maneja por medio de una

---

<sup>6</sup> Web Feature Server – Transactional: Servicio que permite obtener y modificar datos geográficos de tipo vectorial a través de un mecanismo de transacciones.

<sup>7</sup> Web Map Service: Servicio que ofrece mapas a través de internet. A través de este servicio dichos datos no pueden ser modificados.

interfaz web. MapLab está compuesto de tres herramientas: MapEdit, MapBrowser y GMapFactory.

**MapEdit:** es una herramienta que sirve para la administración y edición de mapfiles. También permite dar un control completo sobre todos los aspectos del mapfile, provee validación para asegurar que su estructura sea correcta, ofrece una interfaz para modificar sus objetos y da una vista previa navegable del mapa.

**MapBrowser:** es una herramienta para la selección visual de datos espaciales. También puede ser utilizada para especificar el tamaño del mapa y su proyección.

**GMapFactory:** una herramienta para la rápida creación de aplicaciones cartográficas.

MapLab es fundamentalmente una herramienta de administración que facilita la integración de fuentes remotas y locales de datos geográficos, preparar sus condiciones de simbología y presentación cartográfica e incluso la gestión de las propias aplicaciones web residentes en servidores remotos.

Aunque la licencia es de código abierto con todos los permisos para modificar, publicar, etc., el *copyright* del software y el otorgamiento de la licencia corren a cargo de la compañía DM Solutions Group Inc., desarrolladora principal de MapLab.

### 2.3.3 SAGA

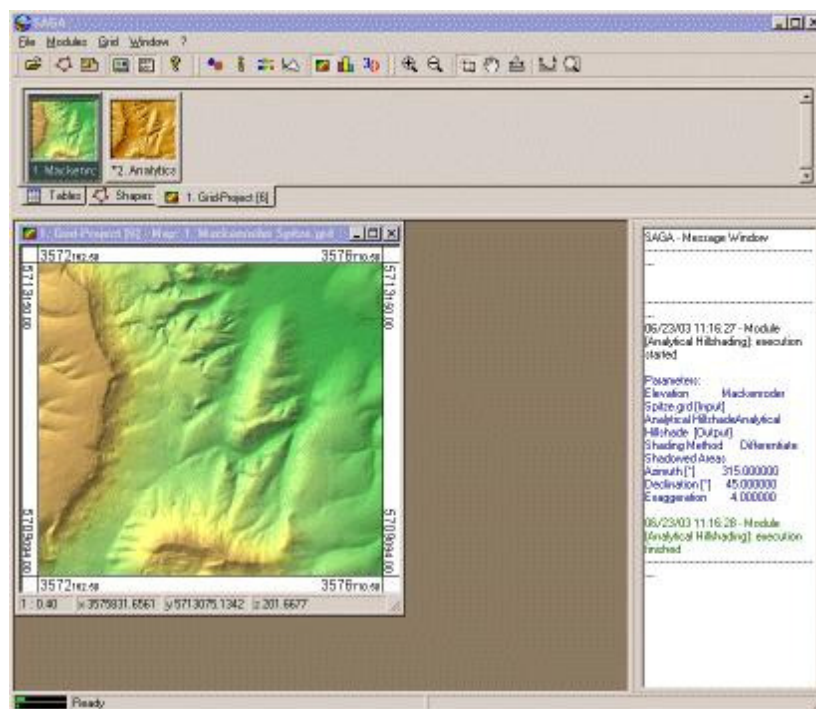
SAGA<sup>8</sup> es una aplicación SIG que brinda tanto opciones de programación como su propia interfaz gráfica para visualización. Fue desarrollado por la Universidad de Göttingen (Alemania).

---

<sup>8</sup> System for Automated Geoscientific Analyses

Entre sus características más importantes están:

- Permite el desarrollo de módulos propios o modificación de su API para personalizarla de acuerdo a la necesidad del usuario, pues la aplicación está hecha en C++.
- Posee alto desempeño en el manejo de mapas de tipo Raster.
- Soporta datos vectoriales, aunque no es su fuerte ya que está orientado al modelo de datos de tipo Raster.
- Permite reclasificar imágenes de acuerdo a características específicas, por ejemplo, si desea dar color según el grado de temperatura de la zona.



**Ilustración 1: Interfaz gráfica de saga**

### 2.3.4 MAPSERVER

MapServer es un componente de desarrollo para la construcción de aplicaciones SIG Web interactivas que permitan la visualización y consulta de información

geográfica en forma de mapas, es de carácter libre, de código abierto y distribuido bajo licencia GPL. Fue desarrollada por la Universidad de Minnesota para el proyecto ForNet en cooperación con la NASA y el Departamento de Recursos Naturales de Minnesota. Actualmente el proyecto de MapServer es utilizado por el proyecto Terrasip un proyecto patrocinado por la NASA.

Su modo de funcionamiento está basado en la generación de lado del servidor Web de imágenes estáticas (JPEG, GIF, PNG, etc.) como respuesta al proceso de las peticiones realizadas por los clientes; es programable a través de PHP, Perl, Python y Java. Estas imágenes son referenciadas posteriormente dentro de la interfaz de usuario que se le muestra al cliente (código html).

Como consecuencia de los trabajos de estandarización en materia SIG de OGC (Open Geospatial Consortium), se ha avanzado mucho en el terreno de las aplicaciones de WebMapping (herramientas que permiten construir aplicaciones Web interactivas para la visualización y consulta de información geográfica en forma de mapas).

En este sentido, MapServer cuenta con extensiones que le permiten escuchar peticiones de mapas formuladas según el protocolo WMS (Web Map Service) de OGC y de enviar respuestas según dicho protocolo. También existen extensiones para MapServer que le permiten proporcionar información cartográfica vectorial<sup>9</sup> según el estándar WFS (Web Feature Server).

Además, cualquier desarrollador puede crear sus propias aplicaciones utilizando el modelo de objetos de MapServer gracias a la biblioteca de componentes MapScript. El API de MapScript puede ser utilizada por lenguajes de scripting

---

<sup>9</sup> Mapas de formato vectorial. La información gráfica se representa internamente por medio de segmentos orientados de rectas o vectores. De este modo un mapa queda reducido a una serie de pares ordenados de coordenadas, utilizados para representar puntos, líneas y superficies.

como PHP, Perl, Python, o incluso desde Java si se desarrollan los conectores JNI<sup>10</sup> necesarios para acceder al API en C de MapScript.

Una parte importante del paquete de MapServer es el CGI MapServ que es la parte de la herramienta encargada de procesar las entradas de los usuarios, la creación directa de las imágenes (mapas) y los resultados de las consultas. El programa acepta entradas por cualquiera de los métodos "GET" O "POST y puede ser utilizado como motor de imágenes.

MapServer funciona en diversas plataformas como son Linux y en la mayor parte de las versiones de Unix, en Windows, Mac OS X, Solaris.

Durante la realización de este estudio se trabajó con la versión 4.4 pero actualmente está disponible la versión 4.8.3 en la página de MapServer<sup>11</sup>; además se utilizó el componente JBox para mapserver, un applet de java el cual se tratará más adelante.

#### **2.3.4.1 El CGI de MapServer**

Una parte importante del paquete de MapServer es el CGI<sup>12</sup> MapServ que es la parte encargada de procesar las entradas de los usuarios, la creación directa de las imágenes (mapas) y los resultados de las consultas.

El CGI de MapServer utiliza los siguientes recursos:

- Un servidor http como Apache o Internet Information Server.
- Software MapServer.
- Un archivo de inicialización que active la primera vista de la aplicación MapServer (opcional).

---

<sup>10</sup> Java Native Interface

<sup>11</sup> <http://mapserver.gis.umn.edu/>

<sup>12</sup> CGI es una norma para establecer comunicación entre un servidor Web y un programa, de tal modo que este último pueda interactuar con internet

- Un archivo que contiene las características del mapa que se van a mostrar en la aplicación, dentro de esas características se encuentran: las capas que conforman el mapa, su extensión, el mapa de referencia, la escala entre otros; este archivo es llamado Mapfile.
- Una plantilla html (Template File) que se usa como interfaz de la aplicación y permite ver los resultados de los queries que se hagan sobre los mapas.
- Una fuente de datos de tipo espacial.<sup>13</sup>

### Instalación del CGI de MapServer

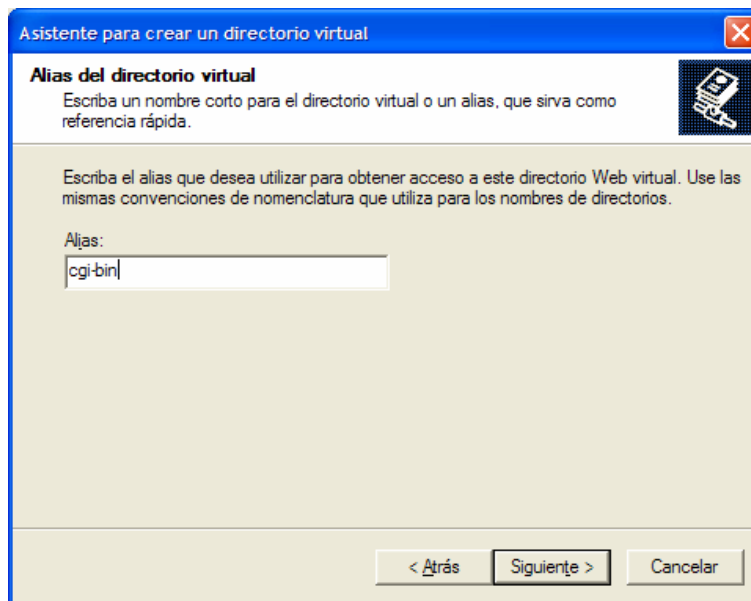
Para el funcionamiento de MapServer se necesita tener previamente instalado un servidor web en el computador; mapserver puede trabajar sobre Apache o sobre Internet Information Server (IIS).

A continuación se explicará la instalación sobre IIS que fue el servidor sobre el cual se trabajó:

1. Copiar la carpeta “cgi-bin” del paquete de MapServer en la raíz del directorio de publicaciones Web: “C:\inetpub\wwwroot”.
2. En el panel de control ir a herramientas administrativas y seleccionar la utilidad **Servicios de Internet Information Server**.
3. Agregar un nuevo directorio virtual en el Sitio Web Predeterminado, colocándole como alias “cgi-bin”, luego continuar con los pasos que solicita el asistente asignando los permisos de:
  - Leer
  - Ejecutar secuencias de comandos
  - Ejecutar aplicaciones ISAPI o CGI
4. Finalmente, reiniciar el servicio del Internet Information Server.

---

<sup>13</sup> Los tipos de datos espaciales se caracterizan porque incluyen la posición, las relaciones espaciales y las propiedades espaciales.



#### 2.3.4.2 jBOX

Como se mencionó anteriormente se utilizó el componente jBox para una mejor visualización de los mapas, el cual es un applet desarrollado en java; de código abierto que permite implementar las funcionalidades básicas para el manejo de la parte gráfica de los mapas como son hacer zoom in, zoom out, mover el mapa, seleccionar un conjunto de elementos por medio de un recuadro, medir distancias entre dos a más puntos.

jBox trabaja con formatos de imágenes JPEG y GIF y con imágenes PNG si se tiene instalado Sun's JRE 1.4.x instalado.

#### Instalación del jBox

La instalación del componente jBox se realiza de la siguiente manera:

1. Se descarga el archivo "jBox.zip" de la página de MapServer.
2. Se crea una carpeta llamada "java" en la raíz del directorio de publicaciones Web: "C:\inetpub\wwwroot\" y dentro de esta carpeta se crea otra carpeta con el nombre jBox.

3. Por último se extrae en la carpeta “jBox” el contenido del archivo “jBox.zip”

### 2.3.4.3 Componentes de una aplicación MapServer

#### 2.3.4.3.1 El archivo de inicialización

El archivo de inicialización se usa para enviar una consulta inicial al servidor http que retorna un resultado del servidor de mapas. MapServer inicialmente está sin estado, este es iniciado y ejecuta una consulta cada vez que esta es recibida, el archivo de inicialización es el encargado de pasar una variedad de parámetros a la aplicación.

Este archivo es por lo general un archivo html, cuya extensión es .htm o .html. Alternativamente se puede construir un hiperlink al servidor MapServer. Este puede pasar los parámetros básicos requeridos por la aplicación MapServer.

Una manera de construir el archivo de inicialización es la siguiente:

Dentro del formulario se declara el parámetro “*action*” que indica el programa del servidor que va a tratar los datos, el “*method*” que indica el método según se van a transferir las variables al programa declarado en el parámetro action.

```
<form action="/cgi-bin/mapserv.exe" method="get" onsubmit="submit_form()">
```

Se declara la ruta donde se encuentra el mapfile y se almacena en la variable map.

```
<input type="hidden" name="map" value="C:\Inetpub\wwwroot\proyectomar\hidro.map">
```

Se declara cada una de las capas que se encuentran definidas en el mapfile

```
<input type="hidden" name="layer1" value="Predios">  
<input type="hidden" name="layer2" value="Malla">  
<input type="hidden" name="layer3" value="Sitios">
```

Se define el parámetro zoomsize el cual va a indicar el tamaño del zoom

```
<input type="hidden" name="zoomsize" value="2">
```

Se define también el parámetro zoomdir que indica el tipo de zoom que se va a realizar.

```
<input type="hidden" name="zoomdir" value="0">
```

La variable program la cual hace el llamado al CGI

```
<input type="hidden" name="program" value="\cgi-bin\mapserv.exe">
```

Los parámetros map\_web\_imageurl y map\_web\_imagepath nos indican el directorio en el cual se van a guardar las imágenes temporales.

```
<input type="hidden" name="map_web_imagepath" value="tmp\">
```

```
<input type="hidden" name="map_web_imageurl" value="/proyectomar/tmp/">
```

Y por último el botón que envía los datos

```
<input id="Submit1" type="submit" value="Ver Mapa" name="Submit1">
```

### **2.3.4.3.2 El Mapfile**

El mapfile es el corazón de MapServer, acá se definen los datos a ser usados por la aplicación, las capas que se van a mostrar y la visualización de las consultas que se realicen sobre los mapas. El mapfile tiene una extensión .map.

#### **Características del Mapfile**

- Si se van a utilizar palabras reservadas de MapServer en la definición de atributos deben colocarse entre comillas dobles.

- El mapfile permite definir hasta 50 layers, esta cantidad puede ser cambiada editando el archivo map.h cambiando la propiedad MS\_MAXLAYERS.
- El objeto mapfile tiene una estructura jerárquica cuya organización tiene en la cima de su estructura al objeto MAP y por debajo de éste objetos como WEB, REFERENCE, LEGEND, SCALEBAR y LAYER.
- Los comentarios se colocan precedidos del signo número (#).
- En el mapfile se declaran los objetos que van a ser utilizados en la aplicación, comenzando con el nombre del objeto y terminando con la palabra END y dentro de este bloque se definen todas las características o atributos que se le vayan a definir al objeto colocando el nombre del atributo y al frente su valor y también pueden ser declarados otros objetos dentro de este. Veremos a continuación los objetos más relevantes del Mapfile.

## **Principales objetos manejados en el Mapfile**

### **MAP**

Este es el objeto principal y el primero que se define en el Mapfile, dentro de él se definen las características principales y todos los objetos que se vayan a utilizar en el mapa.

### **WEB**

Dentro de este objeto se define cómo va a operarse la interfaz Web y las referencias a las páginas html.

### **REFERENCE**

Este objeto define la creación del mapa de referencia; en este mapa se enmarca la parte visualizada del mapa principal.

## **LEGEND**

Este objeto permite crear una leyenda que será la encargada de activar o desactivar las capas o layers que se deseen visualizar en la aplicación.

## **SCALEBAR**

Su objetivo es permitir la creación de una barra de escala para dar información de la escala de los mapas presentados.

## **LAYER**

Este es el objeto más utilizado en el Mapfile; en él se declaran las capas que van a ser usadas para componer el mapa. Las capas se visualizan en el orden que se declaren dentro del mapfile.

## **LABEL**

Este objeto se utiliza para definir una etiqueta que es utilizada para colocarle nombre a un atributo en el mapa.

### **2.3.4.3.3 El Template File**

El archivo Template File controla cómo las salidas de mapas y las leyendas de MapServer se deben presentar en la página html; éste opera como cualquier otro archivo html a excepción de que contiene etiquetas que pueden ser modificadas en el CGI de MapServer. El Template File permite al autor colocar la posición de presentación del mapa, la leyenda y el mapa de referencia; además determina que opciones están disponibles para que el usuario interactúe con la aplicación MapServer (browse, Query, zoom, etc.).

Para producir el documento html que se envía al browser MapServer usa palabras clave en el archivo Template y las reemplaza con información que se encuentra en la fuente de datos SIG. Cuando un Template File es usado para crear un archivo html este es almacenado generalmente con extensión .html.

#### 2.3.4.3.4 El Conjunto de Datos SIG

El CGI MapServer usa archivos de formatos shapefile el cual es el formato propio de la casa ESRI<sup>14</sup> para almacenar información y atributos para un conjunto de elementos geográficos de tipo vectorial. La geometría de un elemento es almacenada como una forma que comprende un conjunto de coordenadas de vectores (punto, línea, polígono).

El formato ESRI está conformado por tres archivos con el mismo nombre pero con extensión diferente, por ejemplo:

- Ciudades.dbf
- Ciudades.shp
- Ciudades.shx

El archivo \*.dbf es un archivo dBase que almacena la información geográfica de los atributos asociados al mapa.

El archivo \*.shp es el archivo que almacena la geometría del elemento (información sobre la forma y la localización).

El archivo \*.shx es el archivo que almacena el índice de la geometría de las entidades y une el archivo \*.shp con el \*.shx.

También utiliza formato raster dependiendo de cómo MapServer es compilado; por defecto soporta archivos geoTiff y archivos Tiff. El conjunto de datos SIG puede ser ubicado en un directorio, el cual es referenciado en el Mapfile.

---

<sup>14</sup> Es la compañía del Grupo EP dedicada a la distribución y la promoción de los Sistemas de Información Geográfica (SIG).

### **2.3.5 MAP OBJECTS**

Otro componente que se estudió en la práctica empresarial fue MapObjects de la casa ESRI, a diferencia de MapServer, MapObjects no es de carácter libre, sin embargo TyT Ltda. propuso su estudio debido a que este componente se vio como una alternativa para el desarrollo de un proyecto que en ese momento se iba a comenzar.

Se trabajó con una versión de evaluación de 3 meses de MapObjects Windows Edition para su análisis y revisión de su funcionalidad.

#### **2.3.5.1 MapObjects Windows Edition**

Es un Control ActiveX compuesto por objetos, que permiten añadir funcionalidad GIS a aplicaciones basadas en entorno Windows existentes, o desarrollar aplicaciones propias personalizadas.

MapObjects proporciona herramientas que facilitan el desarrollo de proyectos SIG, brindando las funcionalidades básicas de este tipo de aplicaciones tales como:

- Añadir componentes de mapa para ampliar la funcionalidad de aplicaciones existentes.
- Desarrollar aplicaciones ligeras de visualización de datos.
- Dar solución a problemas concretos como toma decisiones en mercadeo, manejo de redes de acueducto, control del transporte sobre vías; todo esto por medio de la realización de aplicaciones completas.
- Desarrollar aplicaciones sencillas de consulta que faciliten el acceso a datos generados por aplicaciones GIS más sofisticadas.

### **2.3.5.1.1 Principales características**

Las características principales que posee MapObjects Windows Edition son las siguientes:

- Está basado en tecnología Windows estándar.
  
- Soporta numerosos formatos de datos:
  - Formatos vectoriales GIS: Shapefile, Cobertura, Geodatabase multiusuario (ArcSDE), VPF (Vector Product Format).
  - Formatos CAD: DXF, DGN, DWG.
  
  - Numerosos formatos raster como GRID, GeoTiff o MrSid.
  
- El desarrollo de aplicaciones con este componente puede hacerse utilizando lenguajes como Visual Basic 6 o .NET, esto es una gran ventaja si se tiene en cuenta el número de usuarios que desarrollan bajo estos lenguajes ya que se garantiza un amplio respaldo en cuanto al manejo de sus funciones, variables, procedimientos y demás elementos que componen estos lenguajes,

Además de MapObjects Windows Edition existe la versión MapObjects LT que es una versión ligera, económica de MapObjects Windows, y la versión MapObjects Java que es un conjunto de componentes que permiten construir aplicaciones SIG multiplataforma<sup>15</sup>.

## **2.4 REVISIÓN DE LA METODOLOGÍA DE DESARROLLO RUP Y EL LENGUAJE UML**

La finalidad de una metodología de desarrollo es llevar a cabo el proceso de forma ordenada de manera que el producto sea eficiente, además debe ser una guía para todas las personas que participan en él.

---

<sup>15</sup> Puede funcionar sobre los sistemas operativos Windows y Linux.

La metodología RUP<sup>16</sup> es uno de los procesos más generales que existen actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software.

Las principales características principales de la metodología RUP son:

### **Manejo por casos de uso**

Desarrolla una guía que permite que el sistema establezca la interacción que va a tener tanto con los usuarios como con otros sistemas, durante todos los procesos de diseño, implementación y pruebas.

### **Centrado en arquitectura**

La arquitectura está conformada por cada una de las partes del sistema, es decir, el conjunto de sistemas operativos, plataformas, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.

### **Iterativo e incremental**

A veces es necesario dividir cada parte del desarrollo en ciclos, cada uno de los cuales produce pequeñas versiones del software, buscando que al final de cada ciclo se puedan observar resultados y fallas en el desarrollo. De igual forma, se busca que cada nueva versión sea mejor que la anterior.

En otras palabras, RUP se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, y basándose en el lenguaje UML (*Unified Modeling Language*) como herramienta principal.

---

<sup>16</sup> Llamada así por sus siglas en inglés Rational Unified Process

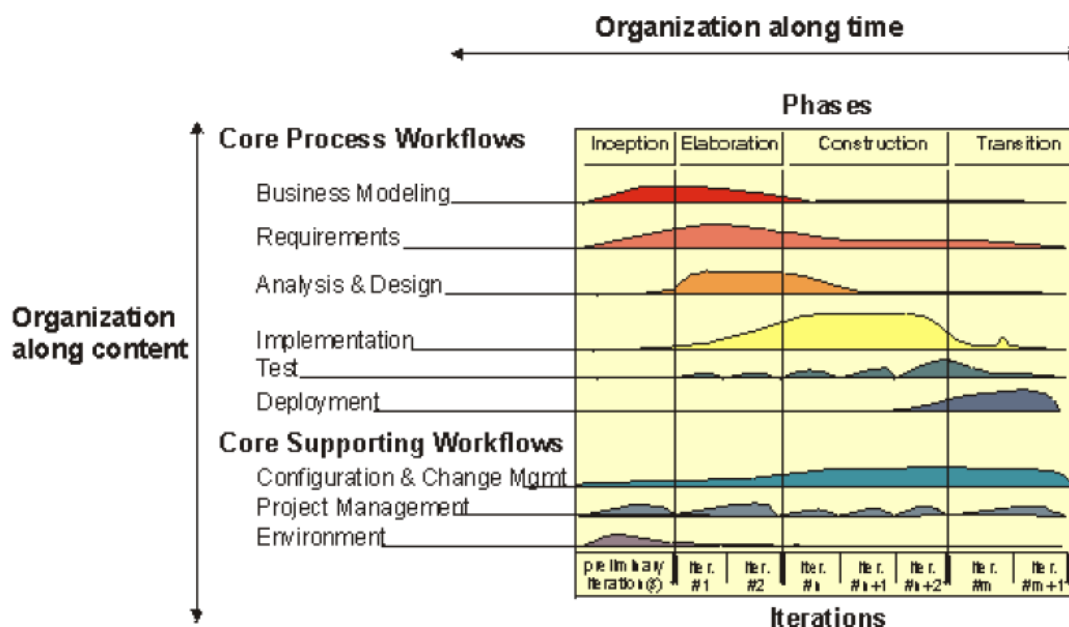
Un proyecto realizado con esta metodología se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición. En cada fase se ejecutarán una o varias iteraciones y dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas.

**Fase de Inicio.** Se establece la finalidad y los alcances que pueda tener el proyecto y se comienza con la identificación de los casos de uso.

**Fase de Elaboración.** Se plantea la arquitectura del producto, y se capturan la mayor parte de los requisitos funcionales.

**Fase de Construcción.** Se obtiene la parte operacional del producto y por lo tanto una nueva versión.

**Fase de Transición.** Se hace la entrega del producto una vez realizadas las pruebas pertinentes, ya sea como producto final o para pasar a generar una nueva versión.



**Ilustración 2: Vista general de RUP**

## **El lenguaje unificado de modelado**

UML es un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Las funciones de UML son:

**Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

**Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.

**Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.

**Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura re-visión.

## **2.5 ESTUDIO DE LA HERRAMIENTA CASE DE ORACLE**

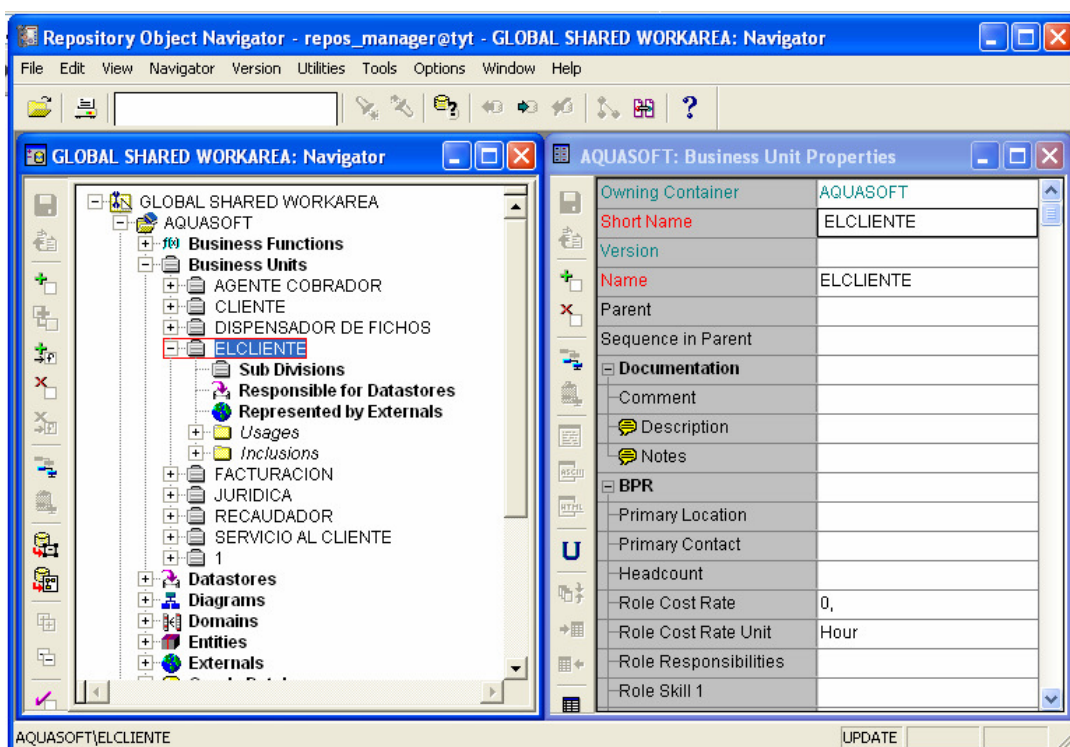
Para el desarrollo del módulo de servicio al cliente, TYT Ltda se apoya en la herramienta CASE de Oracle, Oracle Designer, por esta razón se recibió una charla sobre su funcionalidad y forma de usarla.

Una herramienta CASE se constituye por un conjunto de programas y ayudas que dan asistencia al equipo de desarrollo durante todos los pasos del ciclo de vida de desarrollo de un software.

Oracle Designer es un conjunto de herramientas para guardar las definiciones que necesita el usuario y automatizar la construcción rápida de aplicaciones cliente/servidor gráficas. Integrado con Oracle Developer, Oracle Designer provee una solución para desarrollar sistemas de información ahorrando tiempo por la manera como fluye la información a través de sus utilidades.

Si es cierto que el uso de esta herramienta trae muchas ventajas y facilidades, también requiere de mucho cuidado en el manejo de la información, debido a que cualquier cambio repercute en la totalidad del proyecto.

La información referente a un proyecto se almacena en el Repositorio, y es tomada como un objeto con sus propiedades, de manera cada elemento que se incluya tendrá una descripción y así, cualquier integrante del equipo de desarrollo y directivos podrán saber a qué se está haciendo referencia.



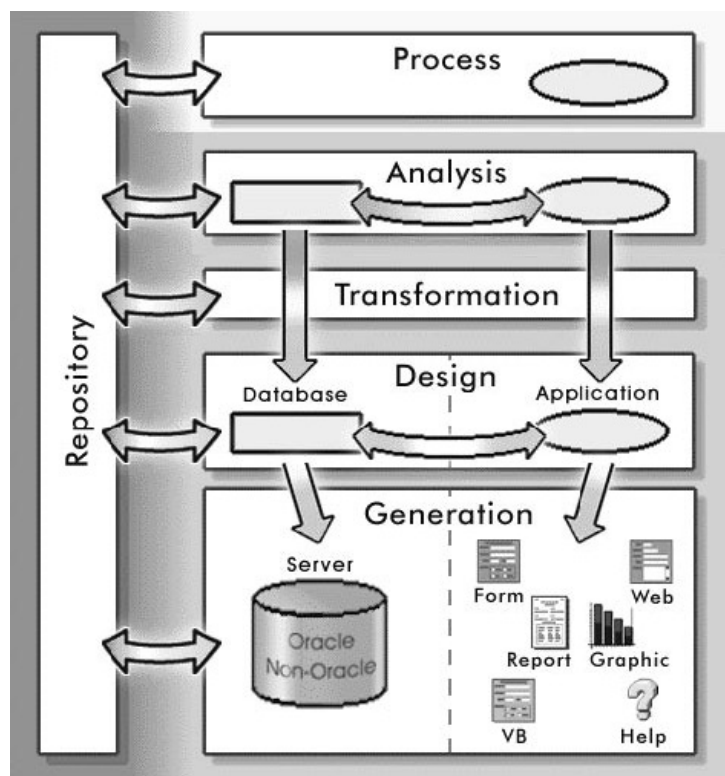
**Ilustración 3: Repositorio de información de Oracle Designer**

Aunque sólo algunas propiedades son obligatorias, puede ocurrir que otras sean de importancia en alguna de las etapas del desarrollo, por ello, se recomienda llenar aquellas que no se entiendan a simple vista o que tienen una importancia significativa.

Una vez almacenada la información en el repositorio, ésta puede ser vista gracias un conjunto de herramientas y utilidades, por medio de las cuales fluye la información.

En el caso del desarrollo del módulo de Servicio al Cliente, ésta herramienta CASE sólo se usa en las fases de análisis y diseño.

En el siguiente diagrama se ilustra la manera como fluye la información desde el análisis hacia el diseño.



**Ilustración 4: Flujo de información de análisis a diseño**

Los datos se transforman mediante la utilidad Database Design Transformer DDT, en donde se transforman las entidades en tablas y los atributos en columnas. Aquí también se establecen las relaciones de muchos a muchos creando una nueva tabla con una llave primaria compuesta por las llaves primarias de las dos entidades involucradas en la relación.

### 3 METODOLOGIA

Para cumplir los objetivos propuestos en la práctica empresarial se utilizó una metodología basada en cinco fases:

1. **Identificación de los temas:** se determina claramente el contenido que se va a estudiar durante el desarrollo de la práctica empresarial.
2. **Búsqueda referente de los temas propuestos:** haciendo uso de medios como libros e Internet se busca información para enterarse de lo que ya existe en el mercado y esté relacionado con los temas propuestos.
3. **Extracción de la información importante de estos temas:** se determina qué elementos hacen un aporte significativo a cada uno de los temas propuestos.
4. **Pruebas:** en esta fase de la práctica se realizan las pruebas pertinentes sobre la funcionalidad de los componentes de desarrollo.
5. **Documentación:** la documentación se realiza durante todo el tiempo de la práctica para que se pueda disponer del conocimiento obtenido en el momento que se requiera.

## **4 APLICACIONES PARA EL DESARROLLO DE HERRAMIENTAS SIG**

Debido a que la empresa estaba a punto de comenzar el desarrollo de una aplicación SIG en ambiente Web, con Visual Studio .NET y un componente que permitiera la visualización y navegación sobre el mapa, edición y consultas de información geográfica, se profundizó en un componente en especial, Mapserver, por varias razones:

- Es el que más desarrollo ha tenido en comparación con los otros que se estudiaron.
- Estaba dentro de las expectativas de la empresa el uso de este componente.
- Las aplicaciones que se desarrollan con Mapserver tiene gran calidad y robustez.<sup>17</sup>
- Se puede conectar con bases de datos como Postgres y Oracle.

### **4.1 APLICACIÓN CON MAPSERVER**

Para probar las funcionalidades con las que cuenta MapServer se realizó una aplicación web para visualizar mapas y hacer consultas sobre ellos. En esta aplicación se implementaron funciones de navegación sobre un mapa (zoom in, zoom out, pan), consultas por selección espacial y por atributos.

La forma como se desarrolló esta aplicación se explica de manera detallada en las siguientes secciones.

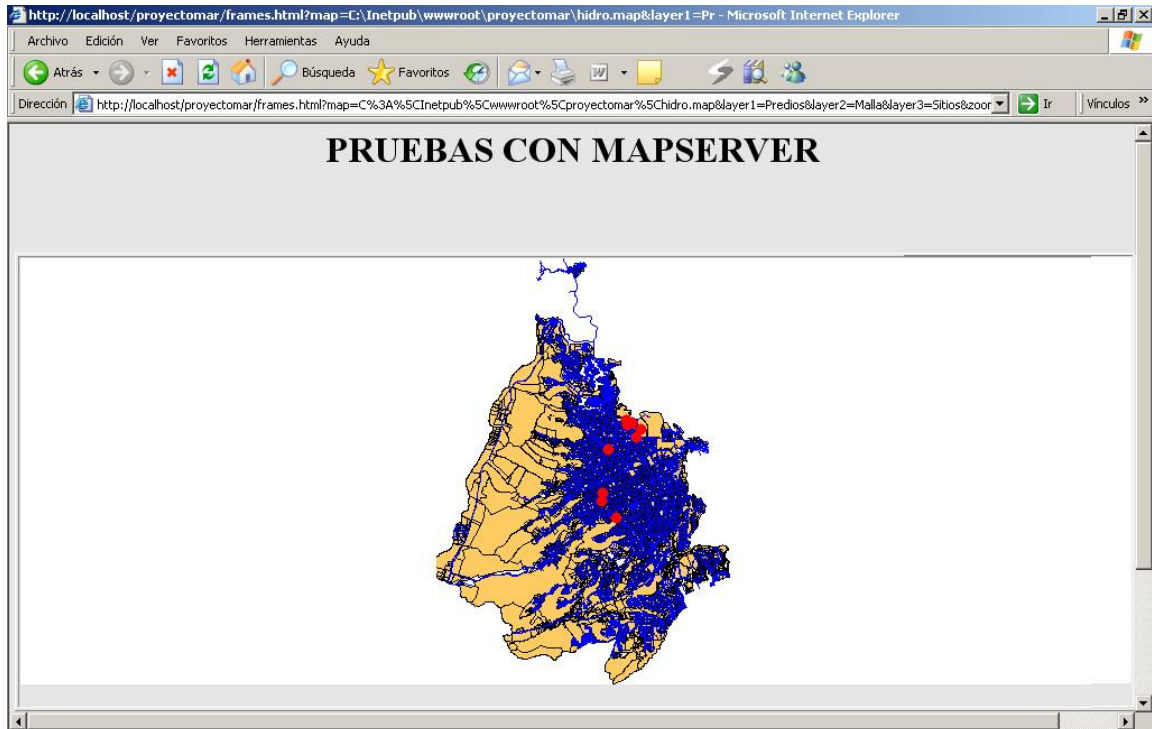
#### **4.1.1 Visualización de capas con MapServer**

Para la visualización de las capas se utiliza el Template File en el cual se muestra la presentación de los datos enviados por el archivo de inicialización y los objetos declarados en el Mapfile. La siguiente figura muestra el mapa de la ciudad de

---

<sup>17</sup> Puede consultar ejemplo de estas aplicaciones en <http://mapserver.gis.umn.edu/gallery>

Bucaramanga formado por tres capas (predios, malla vial y algunos sitios de interés) definidas en el Mapfile



**Ilustración 5: Visualización de capas**

El Mapfile para este primer ejemplo es el siguiente:

```
MAP                                     # Especificación del objeto MAP
  NAME Bucaramanga                       # Nombre del objeto map
  STATUS on
  SIZE 700 380                            # Tamaño de la imagen
  EXTENT 99940 73681 108806 85912       # Extensión georeferenciada del
                                          mapa
  UNITS DD                                # Definición de las unidades de
                                          medida
  IMAGECOLOR 255 255 255                 # Color del fondo de la imagen

WEB                                     # Definición del objeto WEB
  TEMPLATE "templates/principal.html"
  IMAGEPATH "tmp/"                       # Ruta para guardar las imágenes
  IMAGEURL "tmp/"                         temporales
  FOOTER "templates/query_footer.html"
  END

LAYER                                   # Definición del objeto LAYER
  NAME "Predios"                          # Nombre del layer
```

```

STATUS ON
DATA "data/2111u_terreno_del_predio" # Directorio donde está el archivo
TYPE POLYGON # Tipo de formato vectorial
CLASS # Definición del objeto class
    NAME "Predios"
    COLOR 255 204 99
    OUTLINECOLOR 0 0 0
END # Fin de la definición de CLASS
END # Fin de la definición de LAYER

LAYER # Capa de la malla vial
    NAME "Malla" # Nombre de la capa
    STATUS ON # Estado ON para activarla
    DATA "data/malla_buc" # Ruta del archivo
    TYPE LINE # Tipo de mapa vectorial
    CLASS # Inicio objeto clase
        NAME "malla" # Nombre clase
        COLOR 0 0 255 # Color de la capa
    END # Fin del objeto class
END # Fin del objeto layer

LAYER # Capa de la capa sitios
    NAME "Sitios"
    STATUS ON # Estado ON para activarla
    DATA "data/Puntos_buc"
    TYPE POINT # Tipo de mapa vectorial PUNTO
    CLASS
        NAME "Sitios"
        COLOR 0 0 0
        STYLE # Inicio del objeto STYLE
            SYMBOL "circulo" # Símbolo del punto en la capa
            COLOR 255 0 0 # Color del punto
            SIZE 10 # Tamaño del punto
            MINSIZE 1
            MAXSIZE 100
        END # Fin del objeto Style
    END
END # Fin del objeto MAP

```

El anterior es el mecanismo de configuración básica que se usa para MapServer. El Mapfile está compuesto por “objetos” y cada objeto posee características propias y otros objetos. Cada definición de un objeto comienza con el nombre del objeto y termina con la palabra “END”.

Se observa que el Mapfile tiene una estructura como la que se muestra en la siguiente figura:

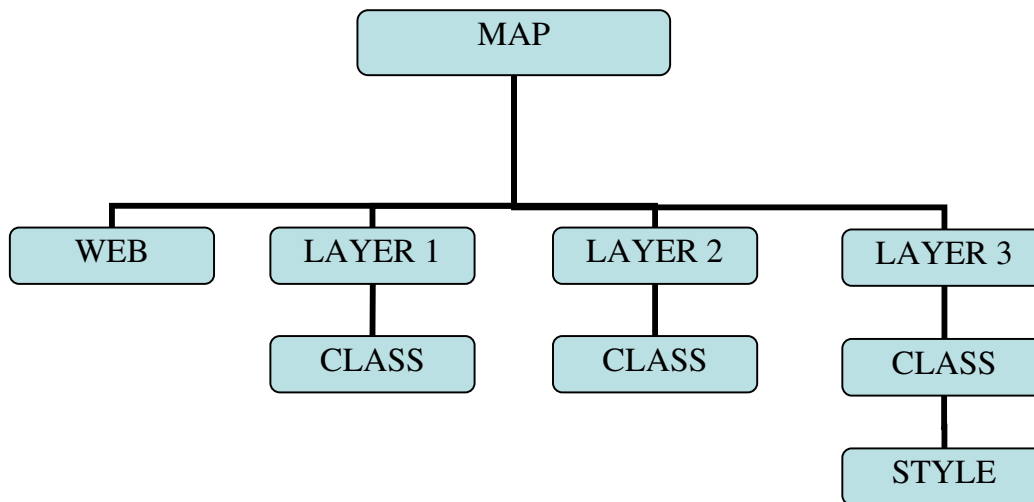


Ilustración 6: Ejemplo de una estructura jerárquica de MapFile

Estos son algunos parámetros definidos dentro del objeto MAP:

**EXTENT:** este parámetro especifica las dimensiones de salida del mapa. Este necesita estar en las mismas unidades de los datos. Para extraer los valores de la extensión se puede usar ArcGis o cualquier otro software GIS.

**SIZE:** especifica el tamaño de la imagen (el mapa) que MapServer va a generar en píxeles. En el mapfile de ejemplo se está generando una imagen de 700 píxeles de ancho por 380 de alto.

**IMAGECOLOR:** define el color de fondo de la imagen. Los valores son RGB.

Algunos de los parámetros del objeto LAYER son:

**LAYER:** marca el inicio de objeto dentro del MAP, permite definir las capas que van a formar el mapa. Se declaran tantos objetos layer como capas se van a mostrar en la aplicación.

**DATA:** en este parámetro se define la ruta a donde se encuentra el archivo shapefile.

**TYPE:** se utiliza para definir el tipo de dato de la capa, se utiliza POLYGON, LINE O POINT para el formato vectorial.

**STATUS:** este parámetro permite colocar el layer visible en el estado ON o si se prefiere ocultar se usa el estado OFF.

#### 4.1.2 Manejo de facilidades gráficas

Dentro de cualquier herramienta SIG se encuentran funciones básicas para el manejo de la parte gráfica de los layers; estas funciones permiten interactuar con el mapa de manera que se pueda hacer un acercamiento al punto donde se haga clic sobre el mapa (Zoom In), un alejamiento (Zoom Out) o un desplazamiento del mapa a la posición deseada (Pan).

Estas funciones se implementan en el Template File enviando algunos parámetros al CGI MapServ. Mediante el applet jBox y programación con javascript se puede dar a estas funciones características avanzadas, por ejemplo realizar zoom a un área enmarcada por un recuadro.

MapServer realiza estas operaciones en modo BROWSE; MapServer maneja dos modos de trabajo sobre el mapa, el modo browse permite al usuario manejar funcionalidades gráficas sobre el mapa y el modo query en el cual se realizan las consultas de información geográfica; este último será analizado más adelante.

##### 4.1.2.1 Implementación del applet jBox

El jBox se incluye para que MapServer muestre la imagen generada del mapa en él y se define en el código html del Template File de la siguiente manera:

```
<applet
  codebase="/java/jBox"
  archive= "jBox.jar"
  code="jBox.class"
  width="[mapwidth]"
  height="[mapheight]"
  name="jBox"
  MAYSRIPT VIEWASTEXT>
```

```

    <param name="image" value="http://[host][img]">
    <PARAM name="box" value="false">
    <PARAM name="thickness" value="2">
    <PARAM name="color" value="green">
    <PARAM NAME="jitter" VALUE="1">
    <PARAM name="verbose" value="true">
</applet>

```

Además para que el jBox funcione correctamente se debe implementar las siguientes funciones en javascript, las cuales van a enviar los parámetros IMGBOX y IMGXY al CGI:

```

<SCRIPT LANGUAGE="JavaScript">
function setbox_handler(name, minx, miny, maxx, maxy)
{
    document.mapserv.imgbox.value = minx + " " + miny + " " + maxx + " " +
maxy;
    document.mapserv.imgxy.value = minx + " " + miny;
    document.mapserv.submit();
}

function seterror_handler(message)
{
    alert(message);
}
</SCRIPT>

```

#### 4.1.2.2 Implementación del Zoom In

La función del Zoom In se define de la siguiente manera:

```

<input type="radio" name="test" onClick="document.jBox.boxon();">

```

#### 4.1.2.3 Implementación del Zoom Out

En la implementación del Zoom Out el nombre del parámetro es "zoomdir" y su valor de -1. Este valor que se envía al CGI permite definir que tipo de zoom se va a realizar.

```

<input type="radio" name="zoomdir" value="-1" onClick="document.jBox.boxoff();">

```

#### 4.1.2.4 Implementación del Pan

La función del Pan (arrastrar la imagen al lugar deseado) se define así:

```
<input type="radio" name="test" onClick="document.jBox.dragon();">
```

Note que para el Zoom In no es necesario colocarle el nombre de “zoomdir” al parámetro para indicar el tipo de zoom a realizar ya que de esto que se encarga la función del jBox “document.jBox.boxon()” la cual realiza el zoom por medio de un recuadro sobre el mapa.

### 4.1.3 Manejo de atributos en la visualización de una mapa

Los atributos para la visualización de los mapas son aquellos que ayudan a personalizar nuestra aplicación, además pueden dar información acerca del mapa y presentar algunas características de ellos.

Entre los atributos más importantes para la visualización de mapas con MapServer están:

- El mapa de referencia
- La leyenda
- La escala
- Las etiquetas
- La simbología

#### 4.1.3.1 El mapa de referencia

Esta es una imagen que se incluye en la aplicación para mostrar qué parte del mapa está siendo visualizada; además da mayor control a la aplicación, pues al hacer clic sobre un punto de la imagen, este es enmarcado por un recuadro y visualizado en el mapa.

MapServer utiliza el objeto REFERENCE definido en el Mapfile de la siguiente manera:

```
REFERENCE
  IMAGE "data/reference1.jpg"      # Localización de la imagen
  SIZE 169 228                    # Tamaño de la imagen
  EXTENT 99940 73681 108806 85912
```

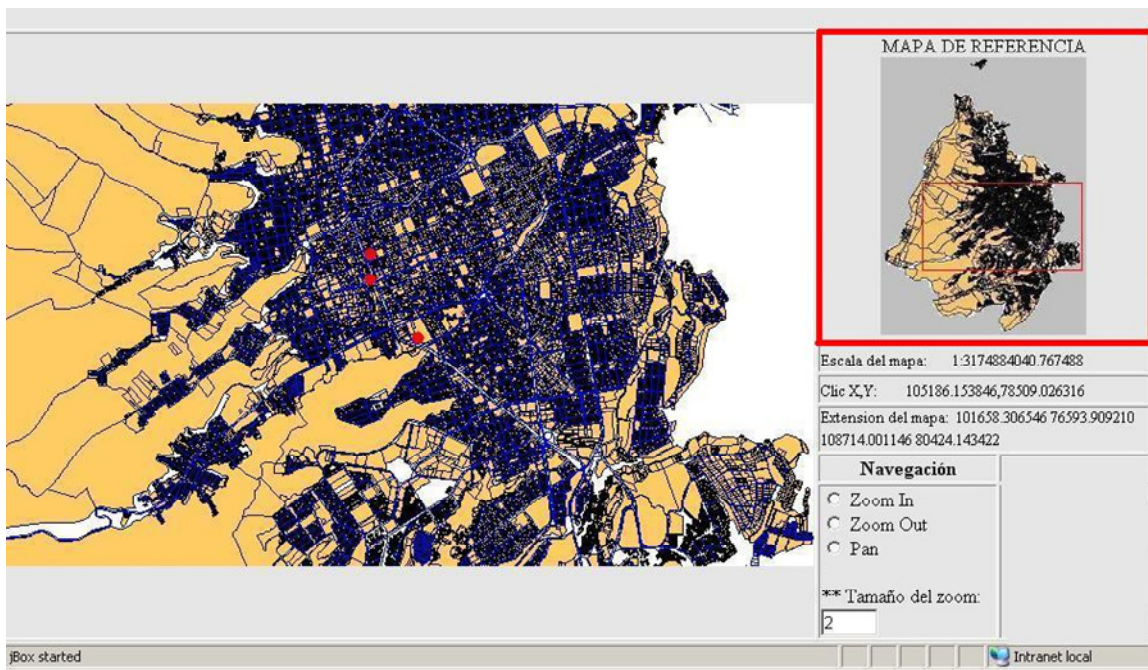
```
STATUS ON
MINBOXSIZE 10
MAXBOXSIZE 150
COLOR -1 -1 -1
OUTLINECOLOR 255 0 0
MARKER 'star'
MARKERSIZE 8
END
```

# Color del cuadro del área seleccionada  
# Símbolo cuando la selección sea muy pequeña

Esta imagen se invoca desde el Template File con la siguiente instrucción:

```
<input type="image" name="ref" SRC="[ref]" width=169 height=228 border=1>
```

La siguiente figura muestra el mapa de referencia en la aplicación enmarcando en un recuadro la parte del mapa que se está visualizando.



**Ilustración 7: Mapa de Referencia**

#### 4.1.3.2 La Leyenda

La leyenda es utilizada para permitir la activación o desactivación de las capas visualizadas en la aplicación, con ésta se puede escoger qué capa será mostrada;

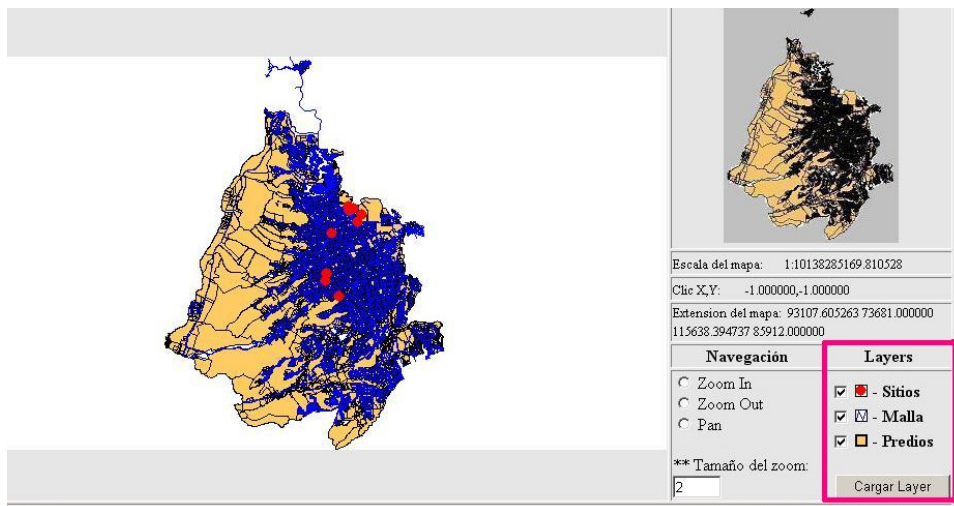
además en la leyenda se muestra la convención de cada una de las capas en el mapa.

Para la leyenda MapServer utiliza el objeto **Legend** definido en el mapfile desde donde es llamado por el Template File para mostrarlo en la aplicación. El objeto Legend cuenta con sus propias características o atributos para su correcto funcionamiento dentro de la aplicación como son el estado (On/Off), el tamaño, la posición, etc.

La declaración del objeto Legend dentro del MapFile se declara de la siguiente manera:

```
LEGEND
  KEYSIZE 10 10
  LABEL
    TYPE bitmap
    SIZE medium
    COLOR 255 0 0
    BACKGROUNDCOLOR 142 42 214
  END
  TEMPLATE "templates/legend.html"
  STATUS ON
END
```

La siguiente figura indica la leyenda implementada en la aplicación realizada en esta práctica.



**Ilustración 8: Leyenda del mapa**

### 4.1.3.3 La escala del mapa

La escala es una característica del mapa que se muestra en la aplicación con una barra de medida y representa la relación entre la distancia de dos puntos situados en la tierra y la distancia que se corresponden con ellos en el mapa, medidos siempre con unidades de la misma especie. Mapserver cuenta con el objeto **Scalebar** para colocarle la escala a un mapa, este objeto contiene sus propios atributos entre los cuales están: unidades, color, tamaño, tipo de fuente, el número de intervalos, la posición. La barra de escala generada por el CGI varía automáticamente conforme el usuario interactúa con el mapa y aplica opciones de Zoom In u Out.

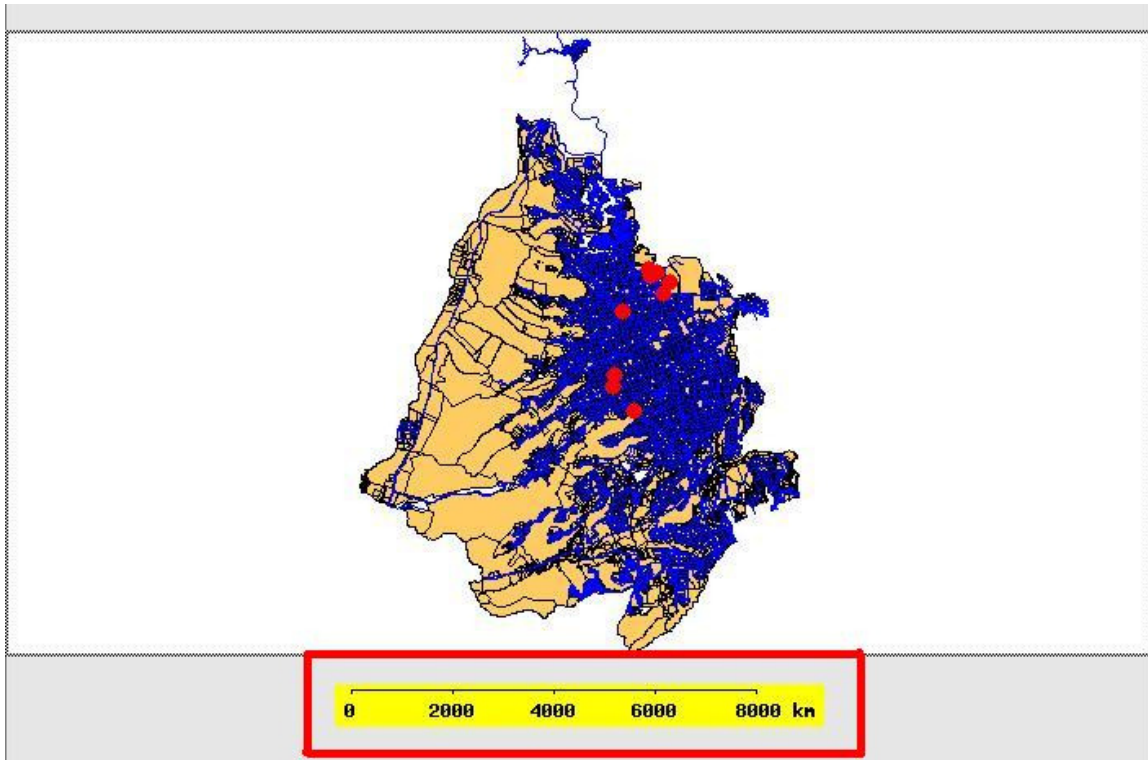
El objeto Scalebar se define en el archivo .map de la siguiente forma:

```
SCALEBAR
  IMAGECOLOR 255 255 0
  LABEL
    COLOR 0 0 0          # Color de la fuente de la escala
    SIZE medium         # Tamaño de la fuente de la escala
    FONT verdanab      # Fuente de la escala
  END
  STYLE 1
  SIZE 300 2           # Tamaño de la escala
  COLOR 0 0 0
  UNITS kilometers    # Unidades de medida de la escala
  INTERVALS 4         # Intervalos a mostrar en la escala
  TRANSPARENT false
  STATUS ON
END
```

Para que la escala se muestre dentro de la aplicación se debe llamar desde el Template File con la siguiente instrucción:

```
< img src = "[scalebar]" >
```

En la figura se aprecia la escala correspondiente del mapa visualizado.



**Ilustración 9: Escala del Mapa**

#### **4.1.3.4 Las etiquetas de un mapa**

Las etiquetas son aquellos que nos permiten rotular un punto, una línea o un polígono, es decir colocarle el nombre a un objeto en el mapa.

Para poder visualizar cualquier característica de un layer en el mapa (por ejemplo el nombre) se debe declarar el atributo LABELITEM dentro del objeto LAYER del que se desee mostrar dicha característica.

MapServer cuenta con el objeto Label el cual tiene una serie de atributos o características para su manejo y funcionamiento entre las cuales están: el ángulo, la fuente, el color, el tamaño, la posición, etc., que nos permite personalizar la etiqueta que se va a mostrar en el mapa.

La forma como quedaría en el Mapfile sería:

```

LAYER # Definición del objeto LABEL
  NAME "Sitios"
  GROUP "MAPAS_VECTORIALES"
  STATUS on
  DATA "data/Puntos_buc"
  TYPE point
  LABELITEM "Nombre" # nombre del campo que se visualiza
                        en la etiqueta
  CLASS
    NAME "puntos"
    COLOR 0 0 0
    STYLE
      SYMBOL "circulo"
      COLOR 255 0 0
      SIZE 10
      MINSIZE 1
      MAXSIZE 100
    END
  LABEL
    COLOR 255 0 0 # Color de la etiqueta en el mapa
    POSITION cl # Posición de la etiqueta
    FONT verdana # Fuente de la etiqueta
    OUTLINECOLOR 255 255 255
  END # Fin del objeto label
END

```

En la figura se muestran algunos puntos etiquetados en el mapa con sus respectivos nombres (labels) los cuales nos ayudan a identificarlos.

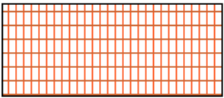


**Ilustración 10: Labels**

#### 4.1.3.5 Simbología para los mapas de puntos

SYMBOL es un objeto que permite escoger la forma del elemento punto de un mapa vectorial para ser visualizado en la aplicación. MapServer utiliza un archivo de símbolos de extensión .sym el cual contiene un conjunto de símbolos con características que definen su forma.

El siguiente es un ejemplo de cómo definir un símbolo en el archivo .sym y la manera de declararlo en el mapfile:

Características del objeto SYMBOL	Objeto CLASS en el Mapfile
<pre> SYMBOL   NAME "line-vertical"   TYPE vector   POINTS     0.5 0     0.5 1   END END SYMBOL   NAME "line-horizontal"   TYPE vector   POINTS     0 0.5     1 0.5   END END </pre>	<pre> CLASS   STYLE     SYMBOL "line-vertical"     SIZE 8     COLOR 255 102 51     OUTLINECOLOR 0 0 0   END   STYLE     SYMBOL "line-horizontal"     SIZE 8     COLOR 204 102 51     OUTLINECOLOR 0 0 0   END END </pre>
	

**Ilustración 11: Creación de un símbolo**

De esta forma podemos crear nuestros propios símbolos y personalizar nuestra aplicación.

#### **4.1.4 Consultas de información geográfica con MapServer**

Dentro de cualquier aplicación SIG es de vital importancia el manejo de la información geográfica, por medio de esta se pueden tomar decisiones muy importantes para la entidad, ya que esta permite obtener respuesta sobre: Donde está algo?, Qué está ocurriendo en algún lugar?, Qué se encuentra en un lugar?; por todo esto la información geográfica se convierte en el ser de cualquier Sistema de Información Geográfica, por eso el manejo de la misma se vuelve algo muy relevante a la hora de desarrollar una aplicación SIG.

Los tipos de consultas más importante dentro de una aplicación SIG son la consulta por selección espacial en la cual se realiza la consulta seleccionando uno

o varios objetos sobre el mapa y la consulta por atributos en la cual se realiza la consulta pasándole parámetros a la misma.

MapServer permite realizar este tipo de consultas teniendo en cuenta las variables necesarias para enviar al CGI, la declaración del objeto QUERY dentro del archivo Mapfile (.map) y su implementación.

Para ésto hay que tener claro que MapServer trabaja en dos tipos de modos diferentes que son el tipo BROWSE en cual se realizan todas las operaciones de visualización gráfica como son el zoom In, zoom Out, Pan y el otro es el modo de los QUERY's en el cual MapServer interpreta que se pueden realizar las consultas.

Para poder mostrar el resultado de las consultas es necesario declarar en el Mapfile los parámetros HEADER y TEMPLATE los cuales contienen las rutas de las plantillas html donde se van a mostrar los resultados de las consultas realizadas. La llamada a las plantillas desde el Mapfile se hace de la siguiente manera:

```
LAYER
  NAME "Sitios"
  GROUP "MAPAS_VECTORIALES"
  STATUS ON
  DATA "data/Puntos_buc"
  TYPE POINT
  LABELITEM "Nombre"
HEADER "templates/sitios_query_header.html"
TEMPLATE "templates/sitios_query_body.html"
  CLASS
    NAME "puntos"
    COLOR 0 0 0
  STYLE
    SYMBOL "circulo"
    COLOR 255 0 0
    SIZE 10
    MINSIZE 1
    MAXSIZE 100
  END
END
END
```

En el parámetro HEADER se especifica la ruta del archivo plantilla el cual contiene el encabezado de la tabla donde se va a mostrar los resultados de la consulta, mientras que en el parámetro TEMPLATE se especifica la ruta de la plantilla que contiene los datos del resultado de la consulta

#### **4.1.4.1 Consultas por selección espacial**

Para las consultas por selección espacial MapServer utiliza los modos QUERY Y NQUERY, con el modo QUERY se realizan consultas espaciales sobre el mapa haciendo clic sobre este y con el modo NQUERY se realizan consultas espaciales de varios elementos con selección por recuadro o igual que el QUERY por medio de un clic sobre el mapa.

Las consultas con NQUERY son consultas múltiples esto quiere decir que si se tienen varios shapes (capas) activos sobre el applet al hacer clic o hacer una selección sobre este la consulta se hará sobre las capas que estén activas en ese momento.

Así que lo único que se requiere para poder realizar una consulta de este tipo es enviarle al CGI el modo en el cual se va a realizar la consulta; para el modo QUERY se especifica de la siguiente manera:

```
<select name="mode" value="query" >
```

Y para el modo NQUERY solo hay que agregarle la función del jBox "document.jBox.boxon();" para que nos permita hacer consultas seleccionando varios puntos enmarcados por un recuadro:

```
<select name="mode" value="nquery" onClick="document.jBox.boxon();" >
```

#### **4.1.4.2 Consultas por atributos**

Otro tipo de consultas que maneja MapServer son las consultas por atributos para estas consultas MapServer utiliza los modos ITEMQUERY y el modo

ITEMNQUERY los cuales trabajan con dos nuevos parámetros que fueron incluidos en esta versión de MapServer llamados Qstring y Qitem; el Qitem es el parámetro que contiene el nombre del campo de la tabla del archivo dbf sobre la cual se va a hacer la consulta y el Qstring es el encargado de armar la cadena de búsqueda.

Al igual que en las consultas por selección, el modo ITEMQUERY retorna como resultado de la consulta la primera coincidencia que encuentre en el archivo .dbf que contiene la información de los layers, mientras que el modo ITEMNQUERY retorna todas las coincidencias que se encuentren en el archivo dbf.

El parámetro Qstring que es el encargado de pasar al CGI tiene la siguiente sintaxis:

Qstring = ('[ qitem ]' operador 'parámetro de búsqueda').

En el rango de operadores permitidos por MapServer están: "in, and, or, =, >, <".

## **4.2 APLICACIÓN CON MAPOBJECTS**

Para probar las funcionalidades con las que cuenta este componente se desarrolló una pequeña aplicación con MapObjects Windows Edition y Visual Basic .Net donde se implementaron las funciones básicas que debe tener cualquier aplicación SIG como son visualización de mapas, Zoom In, Zoom Out, Pan, Full extent, además se desarrollaron algunas funciones para consultas de información sobre los mapas.

En las siguientes secciones se mostrará la aplicación realizada en la cual se implementó el manejo de las facilidades gráficas, algunas consultas de información geográfica y la inserción de puntos en un mapa vectorial.

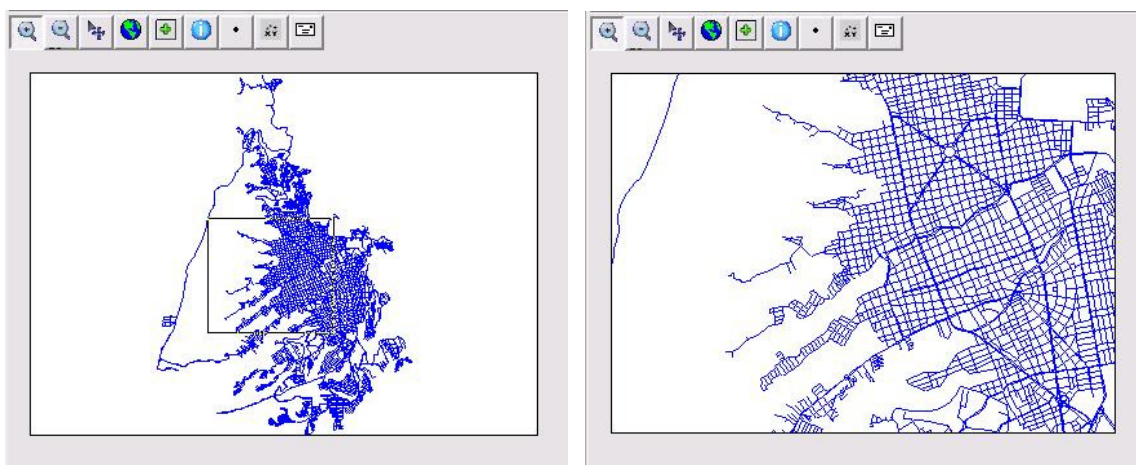
## 4.2.1 Manejo de facilidades gráficas

Las funciones para la implementación de los controles de navegación sobre el mapa son:

### 4.2.1.1 Función para la implementación del zoom In

```
AxMap1.Extent = AxMap1.TrackRectangle
```

En las figuras se observa el *zoom in* hecho sobre un área enmarcada por un recuadro sobre el mapa:



**Ilustración 12: Zoom In**

### 4.2.1.2 Función para la implementación del Zoom Out

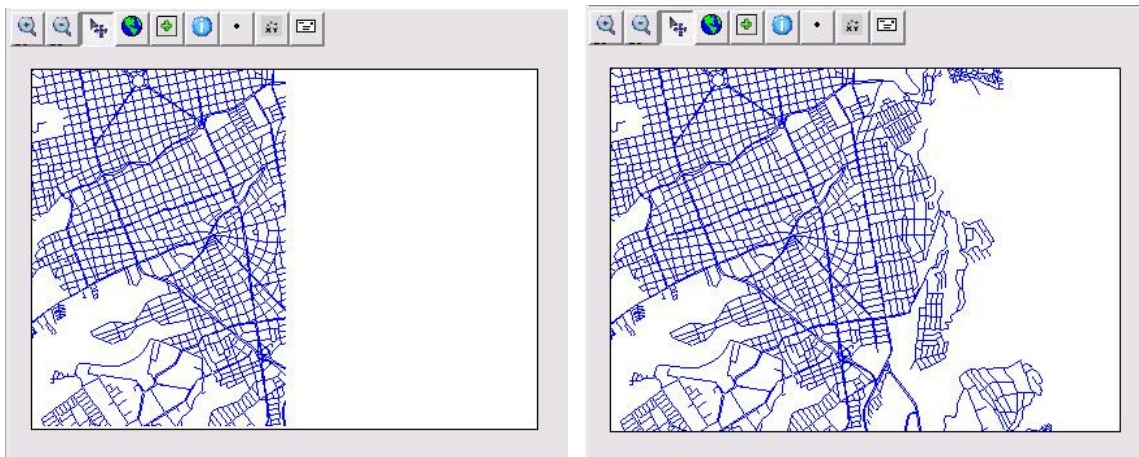
```
LOC = AxMap1.ToMapPoint(e.x, e.y)
curRectangle = AxMap1.Extent
mapWigth = AxMap1.Extent.Width
mapheight = AxMap1.Extent.Height
curRectangle.Right = LOC.X + mapWigth
curRectangle.Left = LOC.X - mapWigth
curRectangle.Top = LOC.Y + mapheight
curRectangle.Bottom = LOC.Y - mapheight
AxMap1.Extent = curRectangle
```

En MapObjects se debe realizar primero el Zoom In sobre el mapa para después poder realizar el Zoom Out y hacer llegar el mapa hasta su tamaño original, a diferencia de MapServer que se puede realizar de forma inmediata y puede hacer el mapa más pequeño que su tamaño original.

#### 4.2.1.3 Función para la implementación del pan

```
AxMap1.Pan()
```

El resultado de esta función se aprecia en las siguientes figuras:



**Ilustración 13: Pan**

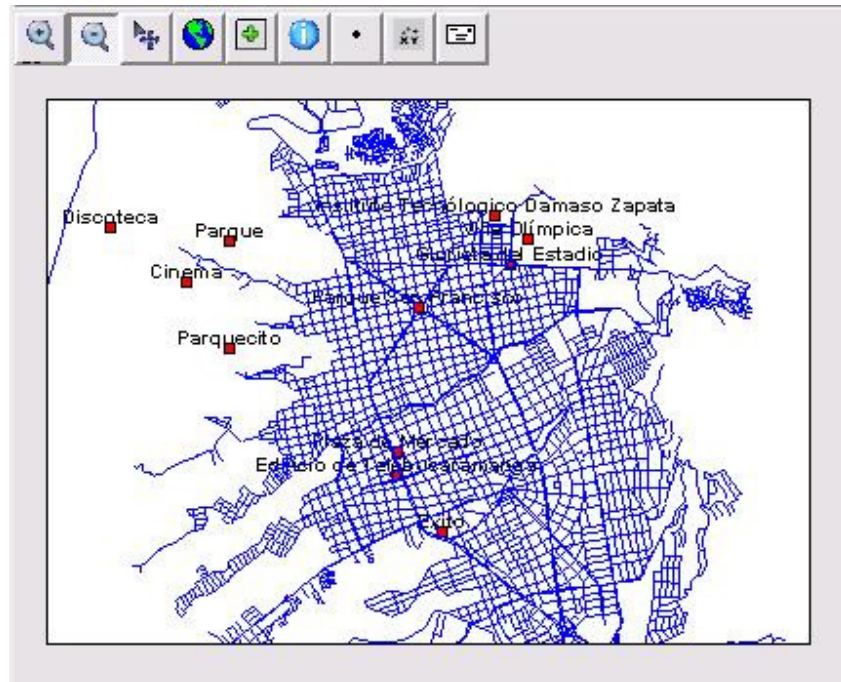
#### 4.2.1.4 Función para la implementación del Full Extent

```
AxMap1.Extent = AxMap1.FullExtent
```

La función para el Full Extent nos permite volver el mapa a su tamaño original después de haber realizado el Zoom In, Zoom Out o el Pan.

#### 4.2.1.5 Manejo de etiquetas

En MapObjects al igual que en MapServer se manejan labels para etiquetar los layers de puntos y mostrar algún tipo de característica para dar información del mismo como se muestra en la figura.



**Ilustración 14: Etiquetas**

#### 4.2.2 Consultas de información geográfica con MapObjects

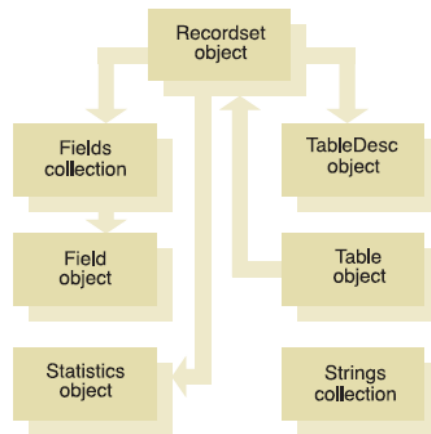
Para el manejo de las consultas MapObjects trabaja con el objeto Recordset con el cual se accede a la información asociada a los layers o shapes.

##### 4.2.2.1 El objeto Recordset

Un **Recordset** es una representación de los registros y los campos en un GeoDataset o Tabla, el Recordset es propiamente del Maplayer y el objeto Table. Un Recordset contiene todos o un subconjunto de los registros del Maplayer o de la Tabla, puede también ser creado al aplicar métodos de búsqueda sobre el

objeto Maplayer o al realizar una búsqueda por medio de una expresión en un objeto Tabla.

La estructura del objeto Recordset es la siguiente:

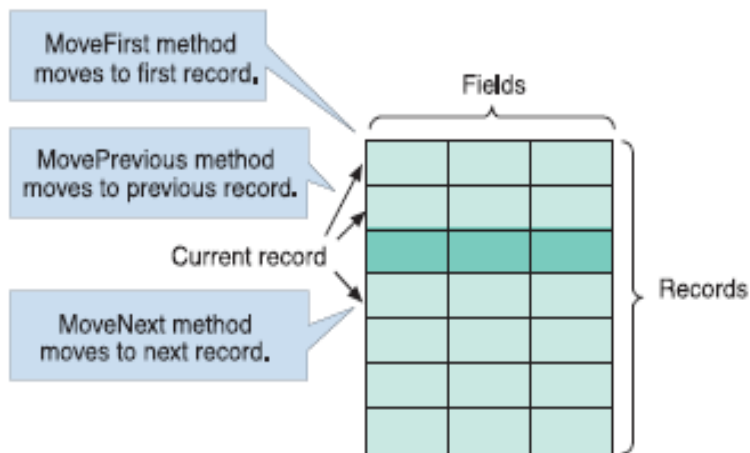


**Ilustración 15: Objeto Recordset**

El objeto Recordset contiene las propiedades Object TablaDesc y Fields Collection por medio del cual retorna una colección de objetos Field que comprende el Recordset.

Se usa el método CalculateStatistics para crear un object Statistics basado en el nombre de un Field en el Recordset. Una vez creado el Objeto Statistics podemos retornar algunos valores de estadísticas básicas de un Field.

También es posible navegar a través del Recordset con el método MoveNext, regresar al registro anterior con el método MovePrevious o también moverse al primer registro del Recordset con el método MoveFirst. Con la propiedad Count se puede saber cuántos registros contiene el Recordset y la propiedad EOF retorna verdadero si la posición en el registro actual está antes del último registro.



**Ilustración 16: Objeto Fields**

La consulta por atributos se hace por medio del método `SearchExpression(expression)` el cual tiene la siguiente sintaxis:

*Variable* = *object*.**SearchExpression**(*expression*)

Donde la sintaxis del método consta de las siguientes partes:

Parte	Descripción
<b>Variable</b>	Una variable que debe ser declarada de tipo Recordset
<b>Object</b>	Un objeto de tipo MapLayer sobre el cual se va a realizar la consulta.
<b>Expression</b>	Una cadena para armar una sentencia SQL para la búsqueda.

Para el tipo de consulta por selección de puntos se utiliza el método `SearchByDistance` el cual tiene la siguiente sintaxis:

*Variable* = *object*.**SearchByDistance**(*shape*, *tolerancia*, *expression*)

Donde la sintaxis del método consta de las siguientes partes:

<b>Parte</b>	<b>Descripción</b>
<b>Variable</b>	Una variable que debe ser declarada de tipo Recordset
<b>Object</b>	Un objeto de tipo MapLayer sobre el cual se va a realizar la consulta.
<b>Shape</b>	Una expresión que evalúa a un objeto punto, línea, polígono, rectángulo o colección de puntos.
<b>Tolerancia</b>	Indica la tolerancia con la que el método determina que características seleccionar.
<b>Expresión</b>	Una cadena para armar una sentencia SQL para la búsqueda.

#### **4.2.3 Inserción de puntos en una capa vectorial**

Se creó un procedimiento que permitiera insertar puntos, ya sea especificando sus coordenadas geográficas, o directamente haciendo clic sobre la capa vectorial de puntos.

Para este procedimiento se utilizó el objeto Recordset mencionado anteriormente y el objeto Field el cual representa una columna de datos dentro de un recordset con un tipo de datos común y un sistema común de propiedades.

El objeto Field contiene las siguientes propiedades:

- **Name:** que es la propiedad por defecto para un objeto Field.
- **Type:** esta propiedad retorna un valor representando que tipo de dato contiene el Field.
- **Value:** con esta propiedad podemos acceder al valor del campo.

Para adicionar puntos al shape se utilizaron los siguientes métodos del Objeto Recordset:

- **Updatable():** esta propiedad retorna verdadero sí el recordset puede ser editado.
- **AddNew():** este método permite crear un nuevo registro en el Recordset.
- **Update():** este método guarda el registro actual y los cambios que se hagan sobre él.
- **StopEditing():** este método termina la edición sobre el recordset.

De igual manera con estos métodos y propiedades tanto del objeto Field como del objeto Recordset se puede editar la información de un punto existente en el shape.

## **5 ACTIVIDADES REALIZADAS EN LA PRIMERA FASE DE DESARROLLO DEL MÓDULO DE SERVICIO AL CLIENTE**

Actualmente, cualquier empresa de servicios públicos (y privada) debe tener una dependencia que se encargue de hacer el contacto con los clientes o usuarios; es imprescindible, teniendo en cuenta que los clientes son parte fundamental para cualquier empresa. Por esta razón es necesario tener una oficina de Servicio al Cliente. TYT Ltda busca sistematizar todas las operaciones que en esta dependencia se realizan de manera con el fin de agilizar los diferentes trámites que se requieran debido a las solicitudes hechas por parte de los usuarios.

Los ingenieros de TyT Ltda. plantearon el desarrollo de un sistema de información para el manejo de los servicios públicos; su enfoque inicialmente se dirigió al campo de la telefonía debido a la experiencia con la que cuentan en esta área; sin embargo al darse cuenta que la competencia en los sistemas de telefonía es muy alta se decidió incursionar en el campo de los acueductos.

Uno de los módulos con los que cuenta este sistema es el de servicio al cliente; su objetivo es gestionar la atención a los requerimientos planteados por los clientes mediante las solicitudes, reclamaciones o sugerencias interpuestas sobre los servicios que presta la compañía, aportando toda la información disponible con el fin de agilizar su solución. Permite registrar y consultar la información de los clientes a través de diferentes medios o canales como: presencial, vía telefónica, vía electrónica, Internet y call center para prestar y gestionar la atención a los requerimientos presentados por los clientes bien sea mediante solicitudes, oportunidad de nuevos negocios, soporte help desk o reclamaciones presentadas.

La metodología de desarrollo que se usa en La Empresa es la RUP<sup>18</sup>, sus fases de desarrollo se muestran en la *Ilustración 17*, donde se observa resaltado mediante un rectángulo, la fase de la cual se participó.

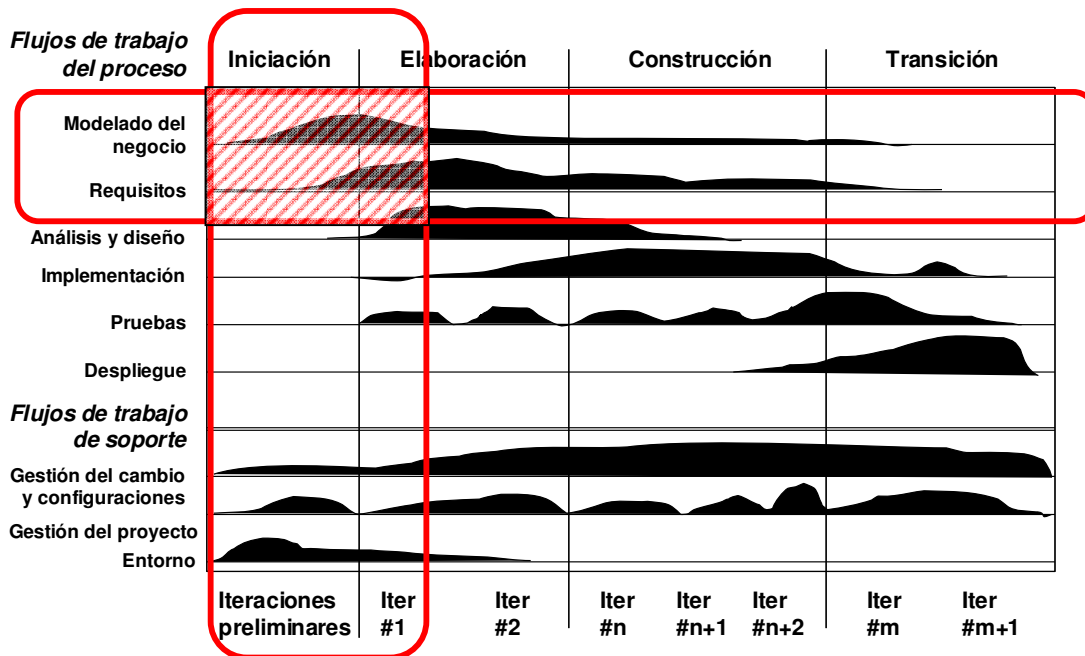


Ilustración 17: Fases de la metodología RUP

En el modelado del negocio se plasmó el conocimiento y experiencia que tienen los ingenieros de TYT Ltda. relacionado con el módulo de servicio al cliente en el campo de la telefonía para aplicarlo al sector de acueductos; esto se hizo mediante una serie de actividades que se listan a continuación:

- Estudio de la metodología de desarrollo RUP y el lenguaje UML<sup>19</sup>.
- Estudio de la Herramienta CASE de Oracle.<sup>20</sup>
- Presentación introductoria sobre el funcionamiento del módulo Servicio al Cliente.

<sup>18</sup> Rational Unified Process

<sup>19</sup> Ver sección 2.4 del marco teórico.

<sup>20</sup> Ver sección 2.5 del marco teórico.

- Elaboración de los Requerimientos generales y específicos versión 1.0.
- Revisión y análisis del documento CUSTOMER MANAGEMENT INFORMATION & BILLING SYSTEM<sup>21</sup> y el documento de procesos en una empresa prestadora del servicio de acueducto.
- Identificación de los casos de uso para el módulo de Servicio al Cliente.

Una vez realizadas las actividades anteriores, se redactó el documento de especificaciones<sup>22</sup> para posteriormente obtener la lista de casos de uso.

## **5.1 PRESENTACIÓN INTRODUCTORIA SOBRE EL FUNCIONAMIENTO DEL MÓDULO SERVICIO AL CLIENTE**

Se hizo una presentación por parte del ingeniero Herber Cala encargado de la dirección del proyecto, en donde se explicó como es el funcionamiento del módulo de servicio al cliente en una empresa de telefonía, con el fin de usar esta información para aplicarla al caso de acueductos.

Dentro de esta charla introductoria se trataron temas como: las funciones del módulo de servicio al cliente, por qué es necesario, qué actividades se realizan, qué trámite se le da a las solicitudes y qué información se maneja.

Cuando un cliente llega a cualquier empresa, la primera oficina a la que se dirige es a la de atención al cliente, si el caso es para realizar una solicitud, pedir un nuevo servicio o dar una sugerencia, es en esta misma oficina donde se atiende. Por esta razón, es importante tener la forma de administrar de manera sistematizada toda la información que se recopila en esta dependencia, tratando

---

<sup>21</sup> Documento que describe los requerimientos básicos que debe implementar los módulos de facturación, servicio al cliente y PQR para un sistema de información de acueductos. <http://www.globalwaterintel.com/>

<sup>22</sup> Ver anexo B

en lo posible, que se haga de manera rápida tanto la inclusión como la consulta de datos.

En esta charla se comentaron las funciones principales que debe tener un módulo de servicio al cliente, entre las cuales se encuentran: atender las nuevas oportunidades de negocio que existan, manejar la información referente a los diferentes productos que pueda ofrecer la empresa, tener un registro del perfil del cliente, manejo de cuentas, gestionar las órdenes de trabajo. Para conocer otras funciones de este módulo que no fueron descritas en la charla, se utilizó el documento "*CUSTOMER MANAGEMENT INFORMATION & BILLING SYSTEM*".

El sistema de información se encarga de gestionar las órdenes de trabajo con el fin de atender las diferentes solicitudes que se presenten en este módulo; ésta gestión se realiza teniendo en cuenta la zona en la que se encuentra el funcionario que la va resolver y un balanceo de cargas que consiste en asignar órdenes de trabajo según el número de tareas que tengan los funcionarios de determinada zona.

El ingeniero y el estudiante en práctica realizaron a partir de esta actividad, un documento de posibles requerimientos iniciales que se deben contemplar en el desarrollo del módulo. (Ver Anexo B).

## **5.2 REALIZACIÓN DE UNA LISTA DE CASOS DE USO PARA EL MÓDULO DE SERVICIO AL CLIENTE**

Del documento realizado de requerimientos iniciales se elaboró una lista de posibles casos de uso a implementarse en el desarrollo del módulo (ver Anexo C), la cual fue revisada y avalada por el ingeniero encargado del proyecto.

Estos casos de uso se empezarán a implementar en una etapa posterior en el desarrollo del proyecto.

## CONCLUSIONES

Después de haber concluido los seis meses de práctica empresarial en la empresa Tecnologías de Información y Telecomunicaciones se puede concluir según los objetivos propuestos que:

El desarrollo de aplicaciones basadas en sistemas de información geográfica actualmente en el mercado son muy solicitadas para el manejo de la información espacial por tal motivo se hace necesario tener un buen conocimiento en herramientas para el desarrollo de estas aplicaciones.

MapServer es la mejor alternativa existente como herramienta de desarrollo para la realización de aplicaciones Web espaciales debido que contiene un gran número de funcionalidades que permiten desarrollar una aplicación robusta y funcional, además es de carácter libre, ya que su licencia es GPL.

MapObjects es un componente robusto para el desarrollo de una aplicación SIG en entorno Windows, pues está compuesto por objetos que permiten añadir diversas funciones a nuestras aplicaciones, además se puede programar en múltiples lenguajes de programación, a pesar que no es de uso libre es una buena alternativa para el desarrollo de proyectos SIG en ambiente Windows.

Tanto MapServer como MapObjects contienen las funciones básicas para la implementación de los controles de navegación sobre la parte gráfica de los mapas (Zoom In, Zoom Out, Pan, Full Extent), las cuales son indispensables en cualquier herramienta para el desarrollo de aplicaciones SIG; además para realizar consultas de información geográfica sobre los mapas MapServer utiliza varios modos y métodos que permiten la extracción de la información espacial asociada a los atributos del mapa; de igual manera MapObjects cuenta con sus propios

objetos y métodos que permiten manipular la información espacial asociada a los mapas y realizar consultas por selección espacial o por atributos.

En cuanto a mi formación profesional el haber estudiado y trabajado la herramienta MapServer me ha servido para incursionar en el desarrollo de aplicaciones SIG, campo que cada día está tomando más importancia en el desarrollo de sistemas y que en nuestro país está comenzando a tener un gran auge, según pude observar durante el estudio que se realizó, ya que tuve la oportunidad de comunicarme con personas que están igualmente interesadas en el desarrollo con este tipo de aplicaciones.

Para TYT Ltda fue de gran valor el estudio de estos componentes, ya que usó como base para el desarrollo de uno de sus proyectos, el cual requería el manejo de mapas y en general de Sistemas de Información Geográfica.

Se conoció una nueva metodología para el desarrollo de proyectos software llamada RUP, la cual es una metodología muy bien estructurada que les facilita a los directores de proyecto y desarrolladores el trabajo en equipo y así poder obtener un producto de muy buena calidad.

La práctica empresarial fue un aporte muy valioso para mi formación como ingeniero de sistemas ya que la participación en el desarrollo de un proyecto software es una buena oportunidad de ir adquiriendo la experiencia necesaria para afrontar todos los retos que se irán presentando de aquí en adelante en cualquier empresa de desarrollo de software, además de enfrentarnos a las necesidades reales que exige el mercado en esta clase de productos.

## **RECOMENDACIONES**

Continuar con el estudio de la herramienta MapServer, pues lo que se investigó de ella en esta Práctica Empresarial es solo una parte del gran poderío que tiene esta herramienta para el desarrollo de aplicaciones espaciales Web.

Tener en cuenta el componente como MapObjects para el desarrollo ya que se pueden hacer aplicaciones muy robustas con ayuda de lenguajes de programación como Visual Basic.net.

## REFERENCIAS BIBLIOGRAFICAS

### LIBROS

[TAM02] TAMAYO TAMAYO, Mario. El proceso de la investigación científica. Limusa Noriega Editores. Cuarta Edición. Año 2002. 440 páginas.

[JAC99] JACOBSON, Ivar, GRADY, Booch y RUMBAUGH James. El Proceso unificado de desarrollo de software. Editorial Addison Wesley. Año 2000. 464 páginas.

### TOMADO DE INTERNET

[1] Santiago, Iván. Diseño Orientado a Objetos con UML. Documento pdf.

[2] VIENNEAU, Aleta. Building Applications with MapObjects, by ESRI. Documento pdf. Estados Unidos. 2001. [www.esri.com](http://www.esri.com)

### PÁGINAS WEB

[1] <http://ms.gis.umn.edu/> Sitio web de Mapserver.

[2] <http://www.esri-es.com/> Sitio web de ESRI

[3] [www.maptools.org/](http://www.maptools.org/) Sitio web de MapLab

[4] <http://geoserver.sourceforge.net/> Sitio web de Geoserver

[5] [www.saga-gis.org](http://www.saga-gis.org) Sitio web de SAGA

- [6] <http://www.gabrielortiz.com/> Sitio web del geógrafo Gabriel Ortiz
- [7] [www.rational.com](http://www.rational.com) Sitio web de la metodología RUP
- [8] Empresas públicas de Medellín: [www.eppm.com](http://www.eppm.com)
- [9] Acueducto de Bogotá: [www.acueducto.com.co](http://www.acueducto.com.co)
- [10] Acueducto del Huila [www.aguadelhuila.gov.co](http://www.aguadelhuila.gov.co)
- [11] Superintendencia de Servicios Públicos Domiciliarios:  
[www.superservicios.gov.co](http://www.superservicios.gov.co).

## ANEXO A

### **Función Javascript para manejar las consultas por atributos sobre MapServer**

```
function fqstring()
{
    var cad;
    if (document.mapserv.qlayer.checked == true)
        document.mapserv.qitem.value = "Nombre";
    cad = '(' + '"' + '[' + document.mapserv.qitem.value + ']' + '"' + 'in'
    + '"' + document.mapserv.caja.value + '"' + ')';
    alert (cad);
    document.mapserv.qstring.value = cad;
}
```

### **Funciones implementadas para mapObjects:**

#### **Identificación de un punto con MapObjects**

```
Dim l As MapObjects2.MapLayer
Dim fld As MapObjects2.Field
Dim p As MapObjects2.Point
Dim recs1 As MapObjects2.Recordset
Dim newItem As String
Dim campo As String
Dim campol As String
l = New MapObjects2.MapLayer
l = g_layer
p = AxMap1.ToMapPoint(e.x, e.y)
Label2.Text = e.x & " " & e.y
If l.shapeType = MapObjects2.ShapeTypeConstants.moShapeTypePolygon Then
    recs1 = l.SearchShape_
    (p,MapObjects2.SearchMethodConstants.moPointInPolygon, "")
    campo = "AREA"
```

```

Else
    recs1 = l.SearchByDistance(p, AxMap1.ToMapDistance(5), "")
    AxMap1.FlashShape(p, 1)
    campo = "Id"
    campol = "Nombre"
End If

If Not recs1.EOF Then
    form.Show()
    form.Focus()
    Dim it As ListViewItem
    form.ListView1.Items.Clear()
    it = Me.form.ListView1.Items.Add(recs1.Fields.Item(campo).Name)
    it.SubItems.Add(recs1.Fields.Item(campo).Value)
    it = Me.form.ListView1.Items.Add(recs1.Fields.Item(campol).Name)
    it.SubItems.Add(recs1.Fields.Item(campol).Value)
End If

```

## Agregar puntos por medio de un clic

```

Dim pt As MapObjects2.Point
pt = AxMap1.ToMapPoint(e.x, e.y)
recs = g_layer.Records
Label2.Text = recs.Count
fld = recs.Fields.Item("shape")
With AxMap1.TrackingLayer.Symbol(0)
    .SymbolType = MapObjects2.SymbolTypeConstants.moPointSymbol
    .Size = 4
    .Style = MapObjects2.MarkerStyleConstants.moTriangleMarker
    .Color = System.Convert.ToUInt32(MapObjects2.ColorConstants.moBlue)
End With
AxMap1.TrackingLayer.AddEvent(pt, 0)
If recs.Updatable Then
    recs.AddNew()
    i = i + 1
    fld.Value = AxMap1.ToMapPoint(e.x, e.y)
    fld = recs.Fields.Item("Nombre")

```

```

    fld.Value = "puntico" & " " & i
    recs.Update()
End If
recs.StopEditing()
AxMap1.Refresh()
End If

```

## Cargar un Layer

```

OpenFileDialog1.Filter = "ESRI ShapeFiles: (*.shp) | *.shp"
OpenFileDialog1.Title = "Abrir Archivo Shape"
OpenFileDialog1.ShowDialog()
If Len(OpenFileDialog1.FileName) = 0 Then Exit Sub
    Dim dc As New MapObjects2.DataConnection
    dc.Database = CurDir()
    TxtDir.Text = OpenFileDialog1.FileName
    Dim name As String = New System.IO.FileInfo(TxtDir.Text).Name
    gs = dc.FindGeoDataset(name)
    If gs Is Nothing Then Exit Sub
    g_layer = New MapObjects2.MapLayer
    g_layer.GeoDataset = gs
If g_layer.shapeType =
    MapObjects2.ShapeTypeConstants.moShapeTypePolygon Then
    TxtTipo.Text = "Polygon"
    g_layer.Symbol.Color =
    System.Convert.ToUInt32(MapObjects2.ColorConstants.moLightYellow)
ElseIf g_layer.shapeType =
    MapObjects2.ShapeTypeConstants.moShapeTypeLine Then
    TxtTipo.Text = "Line"
    g_layer.Symbol.Color =
    System.Convert.ToUInt32(MapObjects2.ColorConstants.moBlue)
Else
    TxtTipo.Text = "Point"
    g_layer.Symbol.Color =
    System.Convert.ToUInt32(MapObjects2.ColorConstants.moRed)
'habilitar las operaciones sobre el layer de puntos
query.Enabled = True
addpoint.Enabled = True

```

```

addCoord.Enabled = True
addLabel.Enabled = True
End If

Dim recs As MapObjects2.Recordset
recs = g_layer.Records
TxtNum.Text = recs.Count
AxMap1.Layers.Add(g_layer)
AxMap1.Extent = AxMap1.FullExtent

```

## Consulta sobre un layer de puntos

```

If lyr.shapeType = MapObjects2.ShapeTypeConstants.moShapeTypePoint Then
    Dim exp As String = "Nombre = '" & TextBox1.Text & "' "
    Dim recs As MapObjects2.Recordset = lyr.SearchExpression(exp)
    Dim fld As MapObjects2.Field
    Dim label As New MapObjects2.LabelRenderer
    Dim etiqueta As String
    If Not recs.EOF Then
        Dim res As MapObjects2.Point = recs.Fields.Item("shape").Value
        AxMap1.FlashShape(res, 3)
        lyr.Renderer = label
        Label2.Text = recs.Fields.Item("Nombre").Value
        AxMap1.CtlRefresh()
    End If
End If

```

## Agregar puntos por coordenadas a un layer

```

Dim point As MapObjects2.Point
Dim x, y As Double
recs = g_layer.Records
cont = recs.Count
fld = recs.Fields.Item("shape")
point = AxMap1.ToMapPoint(x, y)
With AxMap1.TrackingLayer.Symbol(0)
    .SymbolType = MapObjects2.SymbolTypeConstants.moPointSymbol
    .Size = 4

```

```
        .Style = MapObjects2.MarkerStyleConstants.moTriangleMarker
End With
AxMap1.TrackingLayer.AddEvent(point, 0)
If recs.Updatable Then
    recs.AddNew()
    fld.Value = AxMap1.ToMapPoint(x, y)
    fld = recs.Fields.Item("Nombre")
    fld.Value = TextBox4.Text
    fld = recs.Fields.Item("Id")
    fld.Value = cont
    recs.Update()
    MessageBox.Show("El punto se ha Agregado Correctamente", "Agregar
Puntos", MessageBoxButtons.OK, MessageBoxIcon.Information)
End If
recs.StopEditing()
AxMap1.Refresh()
```

## **ANEXO B**

### ***Documento de requerimientos iniciales del módulo de Servicio al Cliente***

El objeto del módulo es gestionar la atención a los requerimientos planteados por los clientes mediante las solicitudes, reclamaciones o sugerencias interpuestas sobre los servicios que presta la compañía, aportando toda la información disponible con el fin de agilizar su solución. Permite registrar y consultar la información de los clientes a través de diferentes medios o canales como: presencial, vía telefónica, vía electrónica, Internet y call center, para prestar y gestionar la atención a los requerimientos presentados por los clientes bien sea mediante solicitudes, oportunidad de nuevos negocios, soporte help desk o reclamaciones presentadas.

Para efectos de los trámites a realizar, el sistema debe permitir la inclusión de los documentos necesarios que se requieren para realizar dicha gestión, así como incluir los pasos a seguir para tramitar solicitudes de toda índole, gestión de quejas y reclamos por tipo o causal, estudios técnicos, programación de ordenes de trabajo, gestión del soporte a los usuarios sobre los servicios contratados, gestión de daños y manejo de interfaz con los sistemas que se requiera.

#### **1. Gestión de oportunidades de negocio**

Permitir la inclusión y seguimiento de oportunidades de negocio, identificando el cliente involucrado, el servicio a contratar y la asignación del ejecutivo para su respectivo trámite. Una oportunidad de negocio, puede originarse por diversas causas: por una llamada telefónica del cliente, solicitud directa del cliente, investigación de mercadeo o por una solicitud en la web.

Las oportunidades tienen asociadas una relación de actividades que es preciso relacionar en el sistema y que se enfocan a dar a conocer los productos al cliente

potencial, brindando asesoría, configuración de la solución y programando jornadas de capacitación.

El sistema deberá llevar registro de las actividades programadas en conjunto con los clientes, generando las debidas alarmas de cumplimiento a los ejecutivos involucrados.

## **2. Gestión y atención de solicitudes**

Permitir la inclusión de solicitudes realizadas por los clientes para cualquiera de los servicios que ofrece la compañía, relacionando la documentación definida para cada una de ellas. Cada tipo de solicitud está asociado a un trámite, el cual garantiza el cumplimiento de cada uno de los pasos requeridos. Estos pasos asociados serán parametrizables y tendrán tiempos máximos de cumplimiento con el objeto de controlar la duración total del proceso.

Facilitar la revisión en línea de la documentación digitalizada de cada cliente o de cada servicio, conociendo la fecha de recepción, el tipo del documento y su respectiva imagen.

La inclusión y procesamiento de solicitudes será independiente del canal utilizado para su registro: presencial, página web, call center, etc.

El procesamiento de la solicitud será inmediato para aquellos tipos de solicitud cuyos pasos o trámite así lo permitan; en caso contrario, en donde se requiera la intervención del usuario final, será éste quien avance en el cumplimiento de las distintas actividades.

Las solicitudes estarán agrupadas por tipos, tendientes a instalar, configurar, retirar o suspender servicios. Se incluyen en este punto, el conjunto de solicitudes relacionadas con cambios de datos básicos de los clientes.

El sistema deberá contemplar la grabación de solicitudes de servicios provisionales o temporales y también que con una misma solicitud, sean requeridos múltiples servicios.

De ser necesario según su trámite, una solicitud podrá generar cobros al sistema de facturación para el cliente involucrado, los cuales deberán tenerse en cuenta en los procesos de liquidación posteriores.

A la grabación de solicitudes le serán aplicadas reglas de validación parametrizables que impidan o permitan el procesamiento de las mismas; por medio de este esquema de restricciones, será posible negar por ejemplo, la venta de nuevos servicios a usuarios morosos o la instalación de servicios en predios sancionados.

En cualquier momento, el sistema debe permitir el seguimiento a las diferentes solicitudes de los clientes conociendo el estado en que se encuentran, la dependencia o responsable que la está gestionando, el tiempo faltante de ejecución o solución. Este seguimiento podrá ser realizado por los usuarios del sistema; el mismo cliente podrá hacer esto mediante el manejo de una clave única a través de la página Web, centros de información, call center o vía telefónica.

### **3. Estudio técnico**

Para dar trámite a las solicitudes, el sistema debe permitir la inclusión de los datos relevantes de estratificación, como por ejemplo el número predial para los diferentes tipos de solicitud a los que aplique esta información; esto debe permitirse realizar de manera manual o automática mediante la interfaz con el sistema de información geográfico.

### **4. Gestión de órdenes de trabajo**

Algunas solicitudes o reclamaciones de los clientes requieren de la programación de operaciones o actividades de campo en planillas de trabajo. Varias órdenes de

trabajo pueden ser asignadas a grupos de funcionarios internos o externos encargados de su solución. Esta asignación puede ser automática (grupos de trabajo predefinidos por zona geográfica o por elementos de red) o asignación manual a criterio del usuario final.

Debe llevarse control a la gestión de actividades realizadas mediante las órdenes de trabajo, para efectos de medir la carga laboral de los grupos encargados de atender este tipo de requerimientos.

Registrar los elementos o materiales utilizados para la realización de las actividades, con el ánimo de controlar materiales consumidos y adicionalmente, que algunos de estos puedan ser cobrados a la cuenta de los clientes.

Las órdenes de trabajo podrán tener diferentes eventos de descargue, dependiendo de la labor realizada, es decir, podrán tramitarse como cumplidas o no cumplidas en cuyo caso se deben manejar causales que permitan realizar una reprogramación de estas.

#### **5. Manejo Electrónico de la documentación de los abonados.**

Permite revisar en línea la documentación digitalizada de cada cliente o de cada servicio, conociendo la fecha de recepción, el tipo del documento y su imagen; mantiene un histórico de los cambios ocurridos a cada número o cuenta asociada al servicio.

#### **6. Administración y control de clientes sancionados.**

Se debe manejar un esquema parametrizable de las diferentes causales por las que se podrá sancionar a un cliente, operando estas en las validaciones que se realizan para la grabación de las diferentes solicitudes de los clientes. Las sanciones podrán ser de tipo temporal o definitivas y estar representadas en términos monetarios o en restricciones a nuevos servicios; deben definirse

igualmente las políticas por medio de las cuales estas sanciones podrán ser eliminadas.

Estas sanciones inciden directamente en la calificación del cliente y servirán de base para la conformación de listas negras.

## **7. Ventas**

Abarca desde el tratamiento previo a la activación del servicio a contratar, hasta los datos administrativos y asociación de plan comercial, incluyendo clasificación del cliente y la posible venta de productos. Gestiona la aceptación o rechazo de la venta, en función de la comprobación de la documentación imprescindible. Así mismo tramita todas las solicitudes que el cliente formule durante su relación comercial con el operador, alteración de datos básicos, modificación de plan comercial, patrón tarifario, inscripción en nuevos servicios, etc.

## **8. Consulta de información general de los clientes.**

Maneja el concepto de la hoja de vida de los clientes definida en la segmentación de mercados, tal como manejo de información *en línea e histórica* referente a lo facturado, los recaudos, el comportamiento de cartera, patrones de consumo de los diferentes servicios, reclamaciones, daños y restricciones solicitadas.

## **9. Estudio de demanda y penetración de productos.**

Permite el manejo de datos referentes al estudio o investigación de mercados, demanda, penetración de productos y demás que se constituyen en soporte para la estructuración de nuevos servicios o rediseño de los existentes. Se manejan variables geográficas, de estratificación, posicionamiento de la competencia y demás.

## **10. Manejo de elementos adicionales al servicio.**

Permite el manejo y control de los elementos adicionales requeridos para la prestación de un servicio ya sea en arriendo, venta, préstamo o cualquier otra

modalidad manteniendo un kárdex o registro de entrada y salida actualizado para controlar la disponibilidad de los mismos. Estos elementos pueden estar ubicados en diferentes dependencias de la Empresa.

#### **11. Documentación requerida para solicitud de los diferentes servicios.**

Generación de todos los documentos necesarios para formalizar el proceso de venta: cartas, constancias, contratos, anexos, facturas, entre otros. Cada documento puede estar relacionado en las diferentes etapas del proceso de Instalación del producto o servicio. Permite asociar documentación requisito a las solicitudes dependiendo de las características especiales definidas a los clientes.

Se mantiene un registro de la documentación entregada o solicitada por los clientes.

#### **12. Instalación, suspensión y rehabilitación de servicios.**

Permite la realización de todo el proceso de venta, suspensión, activación o desactivación de los servicios desde su formalización hasta la legalización generando los cargos hacia facturación para su respectivo cobro.

#### **13. Registro de actividades postventa**

Registra el seguimiento de actividades por producto, cliente, servicio, etc.; cualquier visita, llamada, correo o contacto con el cliente es almacenado con el fin de mantener una bitácora de actividades tanto del cliente como de los diferentes servicios. Permite tabular las encuestas realizadas para medir la satisfacción de los clientes.

Maneja archivos de registro de novedades con rangos de fechas por característica del cliente

#### **14. Estructuración de cuentas de clientes flexibles con adaptación a su organización.**

Permite la estructuración de cuentas que pueden ser asociadas a un grupo de servicios, a un cliente o grupo de clientes para efectos de cobro único, estas deben permitir el borrado, modificación o inclusión de nuevos ítems sin afectar la Facturación del cliente.

**15. Manejo y control de descuentos.**

Permite el manejo controlado del otorgamiento de descuentos especiales y libranzas a los servicios ofrecidos o solicitados por los clientes. Estos deben ir asociados durante todo el proceso de venta de tal forma que se tengan en cuenta al facturar el servicio. Adicionalmente, estas deben estar soportadas con la documentación destinada para tal fin.

**16. Plantillas de Presupuesto.**

Manejo de plantillas para realizar la generación de presupuestos que permitan al cliente potencial el estudio y la posterior contratación de servicios. Estas manejan los criterios de facturación como son: tarifas, reglas de pago, descuentos por volumen, entre otros.

## ANEXO C

### *Casos de uso para el módulo de servicio al cliente*

#### **00 Requerimientos generales**

- 1 Agregar registro

#### **01 Gestión de oportunidades de negocio**

- 1 Incluir nueva oportunidad de negocio
- 2 Hacer seguimiento a las nuevas oportunidades de negocio
- 3 Parametrizar las formas de presentación de las oportunidades de negocio
- 4 Parametrizar los orígenes de las oportunidades De negocio
- 5 Parametrizar las oportunidades de negocio
- 6 Agendar actividades a realizar
- 7 Verificar cumplimiento de las actividades agendadas

#### **02 Gestión y atención de solicitudes**

- 1 Incluir una solicitud
- 2 Registro de la apertura de una cuenta
- 3 Registro del cierre de una cuenta
- 4 Solicitud de cambio de datos del cliente
- 5 Solicitud de cambio de datos de una cuenta
- 6 Solicitud de reconexión del servicio
- 7 Solicitud de desconexión del servicio
- 8 Registro de la apertura de una cuenta para constructoras
- 9 Solicitud de revisión de medidores
- 10 Solicitud de cambio de uso del servicio
- 11 Solicitud de cambio de diámetro de acometida
- 12 Solicitud de cambio o traslado de medidor
- 13 Solicitud de venta de agua en bloque
- 14 Solicitud de venta de agua en bolsa
- 15 Solicitud de desenglobe por coeficiente de facturación
- 16 Solicitud de cambio de localización
- 17 Controlar los tiempos límites para dar respuesta a la solicitud
- 18 Manejar documentación de la solicitud
- 19 Registro de solicitudes provisionales o temporales
- 20 Generar cobros al sistema de facturación
- 21 Registrar cualquier evento asociado a la solicitud
- 22 Validar la grabación de solicitudes
- 23 Hacer seguimiento a las solicitudes
- 24 Reversión de ventas de productos o instalaciones de servicio
- 25 Generar factura de cobro por ventas
- 26 Generar cotizaciones
- 27 Bitácora cambio de datos cliente
- 28 Imprimir documentos

#### **03 Estudio técnico**

- 1 Incluir datos de estratificación

- 2 Asignar datos técnicos para la prestación del servicio

#### **04 Gestión de órdenes de trabajo**

- 1 Asignación manual de órdenes de trabajo
- 2 Asignación automática de ordenes de trabajo
- 3 Ejecutar ordenes de trabajo
- 4 Anular ordenes de trabajo
- 5 Suspende ordenes de trabajo
- 6 Registrar todas las actividades realizadas a las órdenes de trabajo
- 7 Registrar y cobrar materiales utilizados en la ejecución de una orden de trabajo
- 8 Manejar tipos de descargue de órdenes de trabajo
- 9 Hacer seguimiento a las órdenes de trabajo

#### **05 Manejo electrónico de la documentación de los abonados**

- 1 Acceso a la documentación de los clientes
- 2 Consultar la documentación asociada en el registro de las solicitudes
- 3 Consultar la documentación asociada en el registro de las oportunidades de negocio

#### **06 Soporte de cyberservicios prestados a través de la Web**

- 1 Registrar sugerencias, solicitudes y pprs directamente por los clientes

#### **07 Administración y control de clientes sancionados**

- 1 Listar las causas de sanción a un cliente
- 2 Validar si el cliente se encuentra o no sancionado
- 3 Definir políticas para eliminar sanciones
- 4 Llevar una lista de clientes sancionados (lista negra)

#### **08 Ventas**

- 1 Manejar descuentos especiales y libranzas
- 2 Manejar documentación de la venta
- 3 Relacionar el origen de la venta
- 4 Parametrizar las Formas de pago
- 5 Generación masiva de varios servicios o productos
- 6 Relacionar el vendedor del servicio
- 7 Asociar la documentación requisito a la venta
- 8 Generación de comprobante de grabación de solicitud de venta
- 9 Registrar los datos del dueño, arrendatario y responsables de pago

#### **09 Manejo integral de la información referente a ejecutivos**

- 1 Agendar las diferentes actividades de los ejecutivos.
- 2 Generar información del portafolio de servicios y productos del operador

#### **10 Planes comerciales**

- 1 Crear los planes comerciales
- 2 Descuentos de los planes
- 3 Restricciones de los planes
- 4 Beneficios de los planes
- 5 Plantillas para configuración de planes
- 6 Asociar cuenta a plan comercial
- 7 Listar los planes comerciales existentes
- 8 Cambiar plan comercial

## **11 Procedimientos**

- 1 Crear procesos según el tipo de solicitudes
- 2 Manejar tiempo límites dentro de los procesos
- 3 Crear los pasos asociados al proceso de cada tipo de solicitud
- 4 Crear los programas que serán ejecutados en cada paso

## **12 Tipos de solicitudes**

- 1 Parametrizar los tipos de solicitudes

## **13 Documentación requisito**

- 1 Parametrizar el esquema de documentación requisito
- 2 Validar documentación requisito del servicio o plan

## **14 Gestión de clientes**

- 1 Creación de nuevos clientes
- 2 Manejo del esquema de características del cliente
- 3 Validar morosidad del cliente
- 4 Registrar historial de actividades del cliente
- 5 Mantener la hoja de vida de los clientes
- 6 Manejar perfil del cliente
- 7 Categorizar los clientes

## **15 Manejo de elementos adicionales al servicio**

- 1 Crear modalidades de elementos adicionales
- 2 Venta de un elemento adicional a un servicio ya instalado
- 3 Arriendo de un elemento adicional a un servicio ya instalado
- 4 Retirar elementos adicionales del servicio
- 5 Consultar la disponibilidad de los elementos adicionales

### **Estructuración de cuentas de clientes flexibles con adaptación a su**

## **16 organización**

- 1 Crear cuenta mayor
- 2 Asociar cuentas a cuenta mayor
- 3 Retiro de cuentas de la cuenta mayor

## **17 Manejo de plantillas**

- 1 Crear tipos de plantillas
- 2 Crear plantillas por tipo
- 3 Generación de documentos haciendo uso de plantillas

## **18 Manejo de predios**

- 1 Creación de predios
- 2 Asociar un cliente a un predio
- 3 Asociar varios predios a un cliente
- 4 Validar morosidad por predio
- 5 Asociar varios clientes a un predio

## **19 Rutinas de búsqueda por geocódigo**

- 1 Mantener base del conocimiento para normalizador de direcciones

- 2 Utilizar estándares de nomenclatura para los formatos de direcciones
- 3 Hacer búsquedas por geocódigo
- 4 Validar morosidad por dirección

## **20 Manejo de cuentas**

- 1 Consultar los datos básicos de las cuentas
- 2 Consultar datos detallados de las cuentas
- 3 Consultar historial de cambios de la cuenta
- 4 Consultar datos de los clientes asociados a la cuenta
- 5 Consultar datos técnicos de la cuenta
- 6 Consultar servicios asociados a la cuenta
- 7 Consultar pqrs asociadas a la cuenta
- 8 Consultar histórico de facturación asociado a una cuenta
- 9 Validar morosidad de la cuenta
- 10 Revisar estados de cuenta
- 11 Registrar cambio de datos en forma masiva
- 12 Permitir el ingreso de lecturas a cuentas ya cerradas.
- 13 Asignar múltiples responsables a una misma dirección con un único medidor
- 14 Asociar cuentas a un grupo de servicios o clientes