

SIGUIS: HERRAMIENTA SOFTWARE PARA LA CREACIÓN, EDICIÓN Y  
VISUALIZACIÓN DE MAPAS DIGITALES

CARLOS ALBERTO CUTA DURÁN  
JHON ALEXANDER SILVA ARISMENDI

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA  
2006

SIGUIS: HERRAMIENTA SOFTWARE PARA LA CREACIÓN, EDICIÓN Y  
VISUALIZACIÓN DE MAPAS DIGITALES

CARLOS ALBERTO CUTA DURÁN  
JHON ALEXANDER SILVA ARISMENDI

Trabajo de grado para optar por el título de  
Ingeniero de Sistemas

Director  
Ing. ENRIQUE SARMIENTO MORENO

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA  
2006

*Le dedico este trabajo a Dios, que  
es mi guía y quién me da las  
energías para trabajar cada día.*

*A mis padres y mi familia que me  
apoyaron siempre y es gracias a  
ellos que culminaré con éxito este  
proceso educativo.*

*A mis profesores que me  
enseñaron todo lo necesario para  
lograr mi objetivo.*

*A mis amigos y compañeros que  
hicieron fáciles las jornadas de  
estudio y que me acompañaron  
siempre que los necesité.*

*Carlos Alberto Cuta Durán*

*A DIOS por ser mi amigo fiel y  
guía en cada paso que doy en la  
vida.*

*A mi FAMILIA quienes con  
su amor, comprensión y sacrificio  
me han apoyado y colaborado a  
través de los años, logrando  
inculcarme el carácter para poder  
afrentar todos los obstáculos sin  
mirar las circunstancias.*

*A todas las personas a las que he  
llamado "AMIGO", por  
compartir momentos alegres, pero  
sobre todo los difíciles en donde se  
prueba la verdadera amistad.*

*Jhen Alexander Silva  
Arismendi*

## **TITULO. SIGUIS: HERRAMIENTA SOFTWARE PARA LA CREACIÓN, EDICIÓN Y VISUALIZACIÓN DE MAPAS DIGITALES\***

### **Autores:**

CARLOS ALBERTO CUTA DURÁN  
JHON ALEXANDER SILVA ARISMENDI\*\*

**Palabras Clave:** Mapas digitales, sistemas de información geográfica, bases de datos espaciales, formatos de mapas vectoriales.

### **Resumen**

El objetivo de este proyecto fue la creación de un programa software de uso propio de la Escuela de Ingeniería de Sistemas e Informática (EISI), del grupo de investigación en comunicación educativa (GEMA) y de la Universidad Industrial de Santander (UIS), para la administración de la información geográfica contenida en mapas digitales.

Para obtener los resultados esperados con este proyecto, se estudiaron diferentes formatos de representación de mapas digitales, y se seleccionaron los más comunes, o los que más aplicaciones futuras pueden tener. Para futuras actualizaciones de este programa, se recomienda la inclusión de las rutinas de manejo de nuevos formatos de mapas digitales.

El programa obtenido permite visualizar, editar y crear mapas digitales que cumplan con el formato Shapefile o con el formato PostGIS, y administrar la información alfanumérica asociada con ellos.

La metodología aplicada para el desarrollo de este proyecto fue el Modelo de Programación Extrema, mediante el cual se divide el programa completo en pequeñas versiones y se trabaja para alcanzar los objetivos planteados en cada versión, sin pensar en las siguientes, con esto, el desarrollo del software se realiza de forma más simple y rápida.

---

\* Trabajo de grado

\*\* Facultad de Ingenierías Físico-Mecánicas  
Escuela de Ingeniería de Sistemas e Informática  
Director Ing. Enrique Sarmiento Moreno

**TITLE. SIGUIS: SOFTWARE TOOL FOR THE CREATION, EDITION AND VISUALIZATION OF DIGITAL MAPS\***

**Authors:**

CARLOS ALBERTO CUTA DURÁN  
JHON ALEXANDER SILVA ARISMENDI\*\*

**Key Words:** Digital maps, geographic information systems, special databases, vectorial maps format.

**Summary**

The objective of this project was the creation of a software program for the use of the Systems Engineering and Computer Science School, the investigation group in educational communication and the Industrial University of Santander, for the administration of the geographic information contained in digital maps.

To obtain the expected results with this project, different formats of representation of digital maps were studied, and the more common were selected, or those that more future applications may have. For future upgrades on this program, the inclusion of the handling routines of new digital maps formats is recommended.

The obtained program allows to visualize, to edit and to create digital maps that fulfill the Shapefile format or the PostGIS format, and to administer the alphanumeric information associated with them.

The applied methodology for the development of this project was the Model of Extreme Programming, by means of which the complete program is divided in small versions and the work is applied to reach the objectives outlined in each version, without thinking on the following ones, with that, the development of the software is carried out in a simpler and quicker way.

---

\* Degree work

\*\* Physical-Mechanical Engineering Faculty  
Computer Science and Systems Engineering School  
Director Eng. Enrique Sarmiento Moreno

## CONTENIDO

	pág.
INTRODUCCIÓN	12
1. CONTEXTO	13
1.1 DESCRIPCIÓN DEL PROBLEMA	13
1.2 OBJETIVO GENERAL	14
1.3 OBJETIVOS ESPECÍFICOS	14
1.4 VIABILIDAD	14
1.4.1 Financiera	14
1.4.2 Técnica	14
1.4.3 Social	15
2. MARCO TEÓRICO	16
2.1 CARTOGRAFÍA	16
2.2 SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)	16
2.3 MAPAS DIGITALES	18
2.3.1 Ventajas de los mapas digitales	19
2.3.2 Aplicaciones de los mapas digitales	19
2.4 FORMATOS DE REPRESENTACIÓN DE MAPAS DIGITALES	20
2.4.1 Representación vectorial	20
2.4.2 Representación raster	20
2.5 FORMATOS MÁS UTILIZADOS PARA MAPAS VECTORIALES	21
2.5.1 Formato Shapefile	21

2.5.2	PostGIS	22
2.5.3	GML	24
2.5.4	Spatial Data Transfer Standard (SDTS)	26
3.	METODOLOGÍA	27
3.1	METODOLOGÍA DEL PROYECTO	27
3.1.1	Planeación del proyecto	29
3.1.2	Diseño	29
3.1.3	Programación	29
3.1.4	Pruebas	30
4.	SOFTWARE UTILIZADO EN EL DESARROLLO	31
4.1	SISTEMA OPERATIVO	31
4.1.1	Sistema operativo Windows	31
4.2	LENGUAJES Y AMBIENTES DE PROGRAMACIÓN EMPLEADOS	32
4.2.1	Visual Basic.Net	32
4.2.2	SQL	32
5.	DESARROLLO	34
5.1	VISUALIZADOR	34
5.1.1	Requisitos	34
5.1.2	Planeación	34
5.1.3	Diseño	35
5.1.4	Pruebas	35
5.1.5	Resultados esperados	36
5.1.6	Resultados obtenidos	36
5.2	CREADOR	37

5.2.1	Requisitos	37
5.2.2	Planeación	37
5.2.3	Diseño	37
5.2.4	Pruebas	38
5.2.5	Resultados esperados	38
5.2.6	Resultados obtenidos	38
5.3	EDITOR	39
5.3.1	Requisitos	39
5.3.2	Planeación	39
5.3.3	Diseño	39
5.3.4	Pruebas	40
5.3.5	Resultados esperados	40
5.3.6	Resultados obtenidos	40
5.4	POSTGIS	41
5.4.1	Requisitos	41
5.4.2	Planeación	41
5.4.3	Diseño	41
5.4.4	Pruebas	42
5.4.5	Resultados esperados	42
5.4.6	Resultados obtenidos	43
6.	CONCLUSIONES Y RECOMENDACIONES	44
	BIBLIOGRAFÍA	46
	ANEXOS	48
	GLOSARIO	111

## LISTA DE ANEXOS

	pág.
ANEXO A. Descripción técnica ESRI Shapefile	48
ANEXO B. dBASE: Definición técnica	60
ANEXO C. Consultas SQL	64
ANEXO D. Estructura del programa	74
ANEXO E. Manual de instalación de SIGUIS	75
ANEXO F. Manual de usuario de SIGUIS	90

## INTRODUCCIÓN

En este documento se exponen las principales características del proyecto de grado orientado al desarrollo de un programa software para la manipulación de mapas digitales y el marco teórico necesario para su creación.

En el primer capítulo de este documento, se plantea el problema, así como los objetivos que se pretenden alcanzar con este trabajo.

En el segundo capítulo se presenta la teoría relacionada con el tema del trabajo. Se incluyen contenidos sobre la cartografía, los sistemas de información geográfica, los mapas digitales y los principales formatos de representación de éstos.

En el tercer capítulo se explica la metodología empleada en la elaboración de este proyecto: La programación extrema, cuya utilización en el desarrollo de software se está extendiendo en los últimos tiempos por su fácil aplicación y sus buenos resultados.

En el cuarto capítulo se mencionan las herramientas de desarrollo y de soporte a la aplicación utilizadas en la creación del software obtenido.

El quinto capítulo muestra, a manera de comentario, el proceso de desarrollo, la división del proyecto general en etapas, el diseño de cada una de estas etapas, los objetivos planteados y las pruebas realizadas al finalizarlas.

Las conclusiones y recomendaciones se generan como parte de los resultados del proyecto y de las actividades que se deben realizar para la correcta continuación y aplicación de este programa.

Los anexos de este documento incluyen las especificaciones técnicas de los formatos utilizados: El shapefile de ESRI y el formato de almacenamiento de base de datos dBASE, un tutorial de SQL para el manejo de consultas, el manual de instalación del software SIGUIS y sus requisitos técnicos, así como el manual de utilización del programa.

## 1. CONTEXTO

### 1.1 DESCRIPCIÓN DEL PROBLEMA

A través de este proyecto se pretende desarrollar una herramienta software que cumpla con las expectativas del usuario interesado en explorar mediante la cartografía digital, el globo terráqueo y todo punto que quiera localizar en el espacio.

Es de conocimiento público que este campo de la tecnología necesita de mucho más avance y que los ingenieros de sistemas pueden ejercitar sus capacidades para dar un gran aporte al estudio geográfico.

En el mercado de software existen herramientas que realizan esta labor, pero se espera que la Escuela de Ingeniería de Sistemas e Informática (EISI) deba crear sus propias herramientas para un mejor uso, para poder adaptarla a sus necesidades propias presentes o futuras, y para apropiarnos de una tecnología que cada día es más requerida; además que algunas de estas herramientas son muy costosas y las que son gratuitas pueden llegar a ser difíciles de manejar y de adaptar a los propósitos de la EISI o del grupo de investigación.

Desde la antigüedad el hombre ha mostrado interés por el saber geográfico y con herramientas rudimentarias ha logrado aportar avances para su estudio. Hoy en día los ingenieros de sistemas cuentan con innumerables herramientas que nos permiten desarrollar un software capaz de organizar datos geográficos y visualizarlos como mapas digitales mejorando las aplicaciones ya existentes y haciendo que su utilización sea mucho más interactiva con el usuario.

Es necesario para su creación elaborar software abierto que permita ser modificado con el avance científico, además es necesaria una herramienta que sea capaz de representar las variaciones geográficas de un modo que pueda ser comprendido con rapidez por diferentes tipos de personas.

Se espera que este trabajo sea de gran apoyo para el grupo de investigación GEMA, en donde puede servir como herramienta para la elaboración de otros proyectos en esta rama de investigación, y que se pueda utilizar en entidades donde se requieran los mapas digitales para su objetivo funcional.

## 1.2 OBJETIVO GENERAL

Diseñar y desarrollar una herramienta software para la creación, edición y visualización de mapas digitales, utilizando una metodología propicia para su creación y un lenguaje de programación adecuado; para un posterior uso en el grupo de investigación GEMA de la Escuela de Ingeniería de Sistemas e Informática (EISI) para las prácticas sobre el tema y en el posible desarrollo de nuevos proyectos.

## 1.3 OBJETIVOS ESPECÍFICOS

- Seleccionar los formatos de mapas digitales de tipo vectorial más apropiados para su manipulación.
- Desarrollar la herramienta teniendo como base las etapas propias de la metodología seleccionada.
- Realizar una interfaz de usuario sencilla de manejar y que posea los componentes suficientes para cumplir las tareas propuestas: Visualizar mapas, crear mapas, editar mapas, consultar información asociada con los mapas, generar versiones imprimibles (imágenes) de los mapas.
- Realizar pruebas\* a la herramienta, comparándola con otras disponibles en el mercado.

## 1.4 VIABILIDAD

1.4.1 Financiera. El desarrollo del proyecto SIGUIS es viable económicamente, debido a que se cuenta con el software, el hardware y las personas necesarias para su desarrollo. Además, se considera que el manejo eficiente de la información es uno de los pilares de las empresas y en este sentido este proyecto puede colaborar ampliamente por lo que su beneficio puede ser muy grande.

1.4.2 Técnica. En este momento, se cuenta con los conocimientos necesarios en análisis de sistemas, en programación y en herramientas SIG. En otros lugares del mundo se han llevado a cabo proyectos de este mismo estilo, por lo que con ayuda de Internet u otro tipo de comunicación se puede acceder a información pertinente que colabore con su desarrollo. Los formatos que se utilizarán son estándares internacionales, por lo que su forma de uso está completamente

---

\* Las pruebas consisten en verificar su costo y usabilidad, que incluye a la efectividad, la eficiencia y la satisfacción con la cual usuarios específicos pueden alcanzar metas específicas en ambientes particulares, en comparación con otro software de propósitos similares.

documentada en libros, revistas y en las páginas de las empresas que los administran. Los sistemas de información geográfica han tenido un crecimiento importante en los últimos tiempos lo cual permite obtener literatura diversa sobre el tema.

1.4.3 Social. El desarrollo de este proyecto no afecta el medio ambiente, y en cambio es útil para tener la fortaleza que requiere manejar este tema con seriedad.

## 2. MARCO TEÓRICO

### 2.1 CARTOGRAFÍA

Se entiende cartografía como el arte y la ciencia de trazar mapas, pero no solamente se limita a esto, sino también comprende el conjunto de técnicas para la adecuada elaboración de los mapas, los cuales son considerados un medio de comunicación de la humanidad, de aquí la importancia de realizar trabajos en los cuales se muestre la información eficientemente.

En la antigüedad estas técnicas consistían en buscar y seleccionar la información sobre diferentes aspectos de la geografía y poder comprimir estos estudios en un único grupo de datos consistente y preciso, con lo cual se esperaba representar la información lo más fielmente posible, para esto se requería una gran destreza manual y poseer conocimientos de diseño gráfico para poder sintetizar la información en un solo documento legible. No existe un estándar para trazar mapas, más bien cada mapa era diseñado según las herramientas del cartógrafo, la finalidad del mapa y la base de conocimientos.

A medida que surgían los avances tecnológicos, tales como el invento de la imprenta y del computador, la cartografía sufrió cambios radicales entre los que se destacan la realización de los mapas a partir de bases de datos, empleando el computador como un instrumento para controlar la cantidad y la calidad de los datos, para su posterior manipulación y visualización. Con esto se obtuvo la personalización de los mapas en donde el usuario selecciona la región que quiere contemplar, y sobre ésta el puede ejercer su trabajo, para su posterior impresión. También surge el concepto de mapas digitales, en los que dichos mapas solamente se pueden contemplar en la pantalla. Pero uno de los mayores avances es que los programas y los datos para realizar este tipo de mapas son cada vez más accesibles al público, por esto hoy en día se pueden observar gran cantidad de mapas, y estos mapas los realizan a menudo personas que no tienen un mayor conocimiento cartográfico.

### 2.2 SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)

Un sistema de información geográfica, es un sistema de información especializado en el manejo y análisis de la información geográfica (geoespacial) es decir de las abstracciones o representaciones de la realidad geográfica.

Los sistemas de información geográfica (SIG) son hoy en día una herramienta de uso masivo. Existe una gran cantidad de software orientado a facilitar el manejo

de los datos vectoriales sobre representaciones digitales, aunque el costo de estos es bastante elevado. Por otra parte, también es posible encontrar librerías y aplicaciones orientadas al manejo de datos geográficos de forma gratuita. Cada vez son más los recursos de los cuales se dispone para emprender un desarrollo de estas características. Java y sus seguidores, desarrollan librerías libres cada vez mejores, conforme el lenguaje crece.

El objetivo primordial que se espera de un SIG, es que nos permita almacenar, recuperar, analizar y desplegar información geográfica, es decir, disponen de métodos y propiedades destinadas a proporcionar la información descriptiva obtenida desde el mapa en cuestión.

Un SIG puede dividirse en tres partes:

- Herramientas para la entrada y manipulación de información geográfica.
- Un sistema de administración de base de datos (DBMS).
- Herramientas que soportan consultas, análisis y visualización de elementos geográficos.

Un SIG posee las siguientes funciones:

- Entrada de datos: debe poseer la capacidad de recibir datos de diferentes maneras:
  - Digitalización directa: mapas en papel.
  - Entrada de coordenadas en archivos digitales.
  - Teledetección.
- Output: Permite la representación grafica en papel o en formato digital. Esta es una función importante en un SIG pero no es la prioridad.
- Manipulación de datos:
  - Funciones para la transformación matemática de coordenadas: proyecciones cartográficas, transformaciones de datum y funciones para transformaciones geométricas: rotación, reducción y ampliación de la escala de tamaño.
  - Reformateo: Integración, generalización, depuración.
  - Importación y exportación de datos.
- Análisis: Este es el componente mas importante dentro de un SIG, consiste en usar la información disponible para producir nueva información, para lo cual debe poseer las siguientes funciones analíticas:

- Selección geográfica: Búsquedas simples o complejas tanto en el aspecto geográfico como en la base de datos.
- Proximidad: Determinar los objetos que están cerca a otros basándose en las distancias.
- Rutas óptimas: Selección de rutas con la menor distancia o resistencia posible.

Existen dos tipos de variables geográficas:

- Datos discretos: se constituye por elementos separados unos de otros o individualmente distintos, su representación es de forma vectorial.
- Datos continuos: se representan mediante la agrupación de celdas que tratan de imitar este tipo de datos, en donde cada celda tiene un valor separado, un ejemplo de este tipo de variable es la elevación, los programas raster trabajan este tipo de datos.

La base de datos asociada al SIG esta compuesta con datos que pueden ser del siguiente tipo:

- Numérico: Binarios, fechas, enteros, reales.
- Texto: cadenas de caracteres que se usan para describir características nominales o de cualidades de los registros de la base de datos.
- Objetos: archivos, imágenes, etc.

### 2.3 MAPAS DIGITALES

La realización de los mapas surge como consecuencia de satisfacer la necesidad humana que resulta de tratar de resolver la siguiente pregunta: ¿Donde?, por tal motivo hoy existen diversos tipos de mapas: climáticos, biogeográficos, urbanos, políticos, geológicos, y se componen de diferentes elementos para su entendimiento.

Un mapa es una representación de un área geográfica, que suele ser generalmente una porción de la superficie de la Tierra, dibujada o impresa en una superficie plana. La finalidad de la elaboración de los mapas desde sus inicios hasta hoy no ha cambiado, solamente ha variado la tecnología para elaborarlos.

Los hombres han usado mapas desde la más remota antigüedad, y probablemente ya los hacían en épocas prehistóricas. Es posible que incluso algunos dibujos encontrados en cuevas y refugios, con un significado desconocido hasta el momento, sean croquis de los territorios donde vivían y cazaban.

El progreso de la aviación y la fotogrametría y la invención de los aparatos de restitución supuso el paso del mapa antiguo al mapa moderno. Hoy en día podemos hablar de la cartografía automática en la que se establecen mapas con técnicas y medios informáticos totalmente asistidos por computadoras. Con esta aplicación de las tecnologías computacionales se han creado herramientas que han mejorado la introducción de los datos geográficos, cartográficos o socioeconómicos en un ordenador, lo cual ha puesto de manifiesto que el acceso a este tipo de información no es sólo en papel. Estos son los llamados mapas digitales.

### 2.3.1 Ventajas de los mapas digitales

- Un mapa queda reducido a una serie de pares ordenados de coordenadas, utilizados para representar puntos, líneas y superficies, con lo cual su manejo puede ser realizado más rápidamente con un computador, con el fin de obtener mejores resultados, por ejemplo, en la búsqueda de ubicaciones específicas.
- En un mapa digital se puede adjuntar a cualquier sitio geográfico un muy variado tipo de información y en mayores cantidades, sin los límites que impone el papel para los mapas impresos.
- La escala de estos mapas no es fija, y se pueden realizar tan amplios y detallados como se requiera.
- Los mapas digitales pueden perdurar un mayor tiempo, ya que su almacenamiento no es tan costoso, no se desgastan por su uso y pueden ser modificados sin necesidad de ser generados nuevamente.
- Entre sus aplicaciones se encuentran muchos sistemas informáticos, en los cuales los mapas impresos no tienen cabida: Sistemas GPS, Internet, cálculo del área, perímetro y/o volumen de sitios espaciales, análisis y búsqueda más rápida de información.

### 2.3.2 Aplicaciones de los mapas digitales

Los mapas digitales tienen múltiples aplicaciones en diferentes sectores tanto de la productividad de las empresas, como en ubicación espacial. Algunas de estas son:

- Inventario de recursos naturales.
- Inventario y mantenimiento de infraestructura: Eléctrica, Agua, etc.

- GeoMercadeo: Investigación de mercado según la geografía censal, datos estadísticos de encuestas, etc.
- Seguridad pública: Análisis de patrones delictivos, manejo y respuesta de emergencias, rutas más cortas de atención.
- Distribución de fondos: Fondos de emergencia por desastres, educación, etc.
- Sistemas de ubicación GPS y de dirección de automóviles.

## 2.4 FORMATOS DE REPRESENTACIÓN DE MAPAS DIGITALES

En general, la forma en que se guarda la información que representa los datos geográficos posee dos grandes divisiones:

### 2.4.1 Representación vectorial

La información gráfica en este tipo de formato 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.

Las principales características de este tipo de mapas son que permite una fácil y rápida manipulación, su nitidez no depende del método mediante el cual son creados, ocupan poco espacio de almacenamiento en disco duro y se les puede adjuntar una gran cantidad de información temática o alfanumérica.

Existen diferentes formatos en los que se pueden obtener este tipo de mapas. El tipo de formato depende de la entidad desarrolladora; algunos de estos formatos son mucho más extendidos y conocidos que otros.

Algunos programas que utilizan estos mapas son los SIG vectoriales como ESRI ArcInfo, Manifold System y otros.

### 2.4.2 Representación raster

Este tipo de representación se basa en la unidad fundamental llamada celda o píxel, los cuales definen toda una capa de información. Los mapas en este formato se obtienen cuando se digitaliza un mapa o una fotografía, o cuando se obtienen imágenes digitales capturadas por satélites. En ambos casos se obtiene un archivo digital con esta información.

Este formato para representar mapas consiste en una imagen, y como tal, su escala es fija y su nitidez depende de la resolución de la imagen; imágenes con alta resolución, las cuales son ideales para realizar análisis de información geográfica, ocupan un gran espacio en el disco duro

Esta representación es utilizada por programas tales como Idrisi, MapCalc, ArcInfoGRID y otros en la categoría de procesadores de imágenes tales como ERDAS y otros. Algunos programas pueden combinar ambos modelos como lo hace Idrisi, ARC/INFO, y Manifold.

## 2.5 FORMATOS MÁS UTILIZADOS PARA MAPAS VECTORIALES

Existen una gran cantidad de formatos diferentes utilizados para presentar los mapas digitales en formato vectorial, incluyendo algunos creados por cada empresa para utilizar en su interior, sin embargo, para el análisis de este proyecto y de los formatos que se deberían incluir en el programa se utilizaron los más comunes en el mercado, los que más posibilidades futuras tienen y los que cumplen con estándares internacionales.

### 2.5.1 Formato Shapefile

El formato Shapefile es un formato propietario de ESRI<sup>\*</sup>, en el cual cada mapa está completamente definido por tres archivos binarios diferentes, que poseen el mismo nombre, pero diferente extensión: .SHP, .SHX y .DBF.

El primer archivo describe la geometría, contiene los vértices del objeto geográfico representado, el segundo sirve para indicar la ubicación de los elementos en el primer archivo, y el tercero es un fichero de base de datos que contiene los atributos de cada elemento geográfico que se encuentre definido en los otros dos; además puede haber otros ficheros con índices. La estructura de campos de estos ficheros se encuentra en la cabecera de los mismos. Un mapa puede contener tantos objetos geográficos como desee, pero de un mismo tipo de objeto. Los diferentes tipos de objetos geográficos que soportan los shapefile son puntos, líneas, polígonos y multipuntos.

ESRI introdujo el shapefile para proveer a los usuarios de información geográfica un medio simple y efectivo para propagar información geoespacial, como una alternativa al formato de exportación E00. Como resultado, el shapefile se esta

---

<sup>\*</sup> Environmental Systems Research Institute, Inc. (ESRI) es una de las empresas líderes en la creación de software para el manejo de datos geográficos. Entre sus productos se cuentan ARC/INFO, ArcView, ArcGIS. Estos programas se encuentran entre los más utilizados por las empresas y por tanto su formato de distribución (shapefile) se encuentra entre los más comunes para representar la información geográfica.

convirtiéndose en el líder como el estándar para el intercambio de datos geoespaciales y para aplicaciones SIG de escritorio. El formato shapefile se encuentra publicado abiertamente, y está basado en una estructura de datos geoespaciales no propietaria (libre). Una copia de la descripción técnica del formato shapefile puede ser encontrada en Internet en la dirección: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf> \*\*

### 2.5.2 PostGIS

PostGIS es una extensión del sistema de base de datos PostgreSQL, con la cual permite que objetos SIG (Sistemas de Información Geográfica) sean almacenados en la base de datos.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) de código abierto, liberado bajo la licencia BSD (Distribución de Software Berkeley). Es una alternativa a otros sistemas de bases de datos de código abierto como MySQL, Firebird y MaxDB, así como a sistemas comerciales como Oracle o DB2.

Los sistemas de mantenimiento de Bases de Datos relacionales tradicionales (DBMS) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los tipos posibles de datos incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas. PostgreSQL ofrece un valor agregado sustancial, al incorporar conceptos adicionales básicos de tal forma que los usuarios pueden extender fácilmente el sistema con clases, herencia, tipos propios y funciones. Además de otras características que aportan potencia y flexibilidad adicional que colocan a PostgreSQL como la Base de Datos de código abierto más avanzado del mundo.

Entre sus principales características, PostgreSQL ofrece:

- **DBMS Objeto-Relacional:** PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multiusuario, transactions, optimización de consultas, herencia, y arrays.
- **Write Ahead Logging (WAL):** La característica de PostgreSQL conocida como Write Ahead Logging incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de

---

\*\* Véase el anexo A para obtener una traducción libre de la descripción del formato shapefile.

datos se caiga existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos.

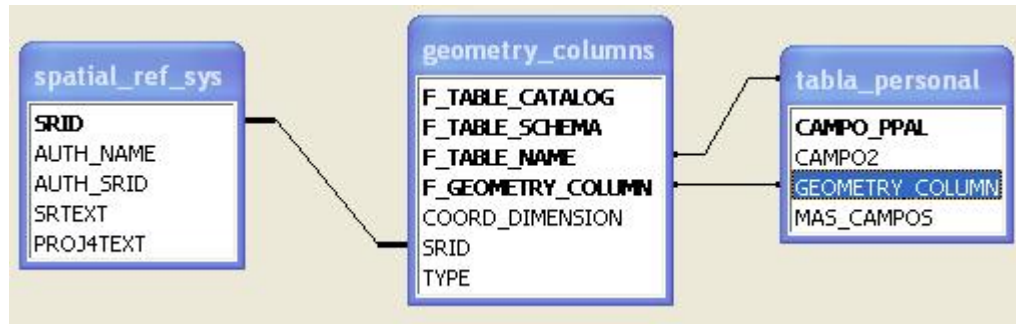
- **Lenguajes Procedurales:** PostgreSQL posee soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- **API Flexible:** La flexibilidad de PostgreSQL ha permitido proporcionar soporte al desarrollo del RDMBS, mediante interfaces entre las que se incluyen Pascal, Pitón, Perl, PHP, ODBC, Java, C, C++.
- **Integridad Referencial:** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos en la base de datos.

PostGIS permite acceder a datos espaciales dentro de una base de datos relacional de PostgreSQL, haciendo que el manejador de base de datos acepte tipos de datos diferentes a los incluidos por defecto y funciones propias para su manejo. Estos tipos de datos son: point, line, polygon, multipoint, multiline, multipolygon y geometrycollections, los cuales son definidos por el OpenGIS Consortium (<http://www.opengis.org>). De esta forma, en una consulta SQL regular, se permite la inclusión de estos nuevos tipos de datos y de algunas de sus funciones.

Para realizar la inserción de los datos geométricos dentro de una tabla perteneciente a una base de datos relacional, PostgreSQL agrega dos tablas de metadatos en las bases de datos que se desea que posean esta característica. Estas tablas cumplen los estándares de OpenGIS, y en una de ellas (geometry\_columns) se almacena la información de las columnas geométricas de cada tabla (tipo de objeto geográfico, nombre de la tabla, etc.) y en la otra (spatial\_ref\_sys) todos los sistemas de referencia espacial soportados (con información sobre sus parámetros, nombres y otras características).

En la figura siguiente, se muestra el modelo entidad/relación correspondiente a una base de datos que soporta los tipos de datos espaciales. En esta figura se muestran las tablas agregadas por PostGIS y la relación con las tablas creadas por los usuarios.

Figura 1. Modelo de tablas de PostGIS



La principal ventaja de este formato es que proporciona una relación directa entre los datos espaciales y los datos alfanuméricos asociados a ellos, a diferencia del formato shapefile en el cual la única relación existente es la posición.

PostGIS es desarrollado por Refrations Research Inc, como un proyecto de investigación de tecnología para bases de datos espaciales. Para más información sobre el proyecto y la herramienta, se puede visitar la página Web <http://postgis.refrations.net>.

### 2.5.3 GML

Es un modelo XML para información geográfica, aspecto importante para cargar y/o transportar información independientemente de la plataforma. Con este formato es posible incorporar información sobre la fuente de datos; el modelo de datos ofrece un conjunto de elementos geográficos básicos: punto, línea, polígono, con información asociada como: sistema de referencia, identificación, y más; además permite definir otras geometrías según las necesidades que se presenten.

En los SIG podemos encontrar varias formas de generar y almacenar los datos, los cuales pueden encontrarse en distintos tipos de base de datos, páginas html, etc., y pueden seguir o no un estándar. Por eso es necesario que a la hora de compartir datos, las empresas establezcan formatos en los cuales brindarán la información o esperaran recibirla, y el que la recibe deberá convertirla al formato utilizado por su sistema, es aquí donde es útil la utilización de estándares ya que reduce los costos y facilita la automatización de dicha tarea.

Entre los formatos GML y shapefile existen similitudes de sus características principales, por tanto, existe la posibilidad de aplicar cierto conjunto de transformaciones para poder convertir un formato en otro y así poder aprovechar las ventajas que ofrece cada tipo de mapa según la necesidad presente, pero cabe resaltar que en algunas transformaciones de GML a Shapefile se presenta pérdida de la información, por tanto en esos tipos de geometría se deberá estudiar

las necesidades del proyecto, y así poder decidir cual de los formatos es el más adecuado para el trabajo, ya que si se desea transportar los datos vía Internet lo sugerido sería utilizar GML, pero si se quiere una integridad entre distintas herramientas lo aconsejable sería el shapefile, ya que este es un formato muy utilizado y se puede transformar a GML sin pérdida de información.

La representación de los elementos se basa en un modelo XML que posee la siguiente estructura:

- Puntos:

```
<complexType name="PointType">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <sequence>
        <choice>
          <element ref="gml:coord"/>
          <element ref="gml:coordinates"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

- Multipunto:

```
<complexType name="MultiPointType">
  <complexContent>
    <restriction base="gml:GeometryCollectionType">
      <sequence>
        <element name="pointMember" maxOccurs="unbounded">
          <complexType>
            <sequence> <element ref="gml:Point"/> </sequence>
          </complexType>
        </element>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

- Líneas:

```
<complexType name="MultiLineStringType">
  <complexContent>
    <restriction base="gml:GeometryCollectionType">
      <sequence>
        <element name="lineStringMember" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element ref="gml:LineString"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

```
</sequence>
</restriction>
</complexContent>
</complexType>
```

- Polígono:

```
<complexType name="PolygonType">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <sequence>
        <element name="outerBoundaryIs">
          <complexType> <sequence> <element ref="gml:LinearRing"/>
        </sequence>
      </complexType>
    </element>
    <element name="innerBoundaryIs" minOccurs="0"
      maxOccurs="unbounded">
      <complexType> <sequence> <element ref="gml:LinearRing"/>
    </sequence>
    </complexType>
  </element>
</sequence>
</extension>
</complexContent>
</complexType>
```

#### 2.5.4 Spatial Data Transfer Standard (SDTS)

El SDTS es un estándar federal de Estados Unidos, diseñado para apoyar la transferencia de diferentes tipos de datos geográficos y cartográficos. El estándar define la estructura y el contenido de los datos espaciales para asistir la transferencia de datos entre diferentes bases de datos. SDTS es también conocido como el Estándar Federal de Procesamiento de Información (FIPS por sus siglas en inglés). Más información puede ser obtenida en la página Web: <http://mcmcweb.er.usgs.gov/sdts>.

### 3. METODOLOGÍA

#### 3.1 METODOLOGÍA DEL PROYECTO

La metodología que se utilizó en el desarrollo del proyecto SIGUIS, es el Modelo de programación extrema. La programación extrema nace como nueva disciplina de desarrollo de software, y ha causado un gran revuelo entre el colectivo de programadores del mundo. La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código; para algunos no es más que aplicar una pura lógica.

Entre sus principales características se encuentran:

- Esta orientada a quien produce y usa el software.
- Reduce el costo del cambio en todas las etapas del ciclo de vida del sistema.
- Combina las que han demostrado ser las mejores practicas para desarrollar software y las lleva al extremo.

Por lo tanto la programación extrema puede resumirse en los siguientes puntos:

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes de que sean necesarias.

Una de las cosas que los programadores deben tener muy claro es que en el ciclo de vida del desarrollo de un proyecto software los cambios van a aparecer, cambiarán los requisitos, las reglas de negocio, el personal, la tecnología, todo va a cambiar. Por tanto el problema no es el cambio en si, ya que este va a suceder, sino la incapacidad de enfrentar estos cambios.

Como en otra cualquier actividad humana se necesitan valores para desarrollar el trabajo establecido y conseguir los planteamientos iniciales, por lo tanto en esta

metodología se plantea los siguientes valores como esenciales para el grupo de trabajo:

- Comunicación.
- Sencillez.
- Retroalimentación.
- Valentía.

Las prácticas de la programación extrema traducen estos valores en actividades que el programador debe desarrollar diariamente, en esta metodología dichas prácticas son llevadas al extremo para conseguir más que la suma de sus partes:

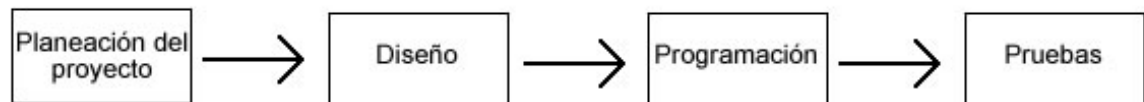
- Retroalimentación a escala fina: En esta etapa se hace un desarrollo guiado por pruebas, donde los clientes y los programadores negocian el alcance de una iteración del proyecto y el estimado para alcanzarla. Por lo tanto se sugiere la presencia continua del cliente y la programación en pares.
- Proceso continuo en lugar de por lotes: Se debe hacer una integración continua del código en donde llegado el caso se tendrá que hacer una refabricación sin piedad, es decir, modificaciones en donde no se altere la comportamiento externo pero se mejore su estructura interna, y realizar liberaciones pequeñas.
- Entendimiento compartido: Se debe hacer el diseño de la forma más simple, pero que alcance las necesidades primarias, se recomienda establecer las convenciones del código para facilitar su uso por cualquier integrante del grupo de trabajo.
- Bienestar del programador: Se debe establecer un horario de trabajo de 40 horas semanales, ya que esta demostrado que los programadores cansados son menos productivos y más propenso a errores.

Se escogió esta metodología puesto que sus características se acoplan al proyecto, ya que esta expone una retroalimentación a escala fina de cada versión liberada en las cuales es primordial el desarrollo de pruebas de unidad y aceptación. Además cada versión debe ser lo más simple posible con lo cual se busca que cualquier integrante del grupo pueda entender la intención de lo que se esta haciendo, aspecto importante para el proyecto ya que uno de los objetivos es que pueda servir de apoyo para futuros proyectos del grupo, con lo cual no habría una dependencia a cierto grupo de desarrolladores. Para cumplir con esto, se

plantea la programación en pares\* con lo cual se espera una mejor funcionalidad en la ejecución del proyecto y una mejor y más continua comunicación, con lo que se podrán acordar los aspectos a cubrir en las respectivas versiones y la manera más eficaz de poder lograrlos.

Esta metodología se divide en las siguientes fases para cada versión:

Figura 2. Modelo de programación extrema



### 3.1.1 Planeación del proyecto

En esta etapa se definirán las actividades que realizará el sistema. Es en donde se deben identificar los problemas, proponer soluciones y definir aquellos puntos a los que se le debe dar mayor importancia.

Para cada versión se deben definir su alcance, las pruebas a realizarse, establecer un plan de liberación de manera específica, en el cual se establecerá el tiempo de entrega para poder hacer la respectiva retroalimentación.

### 3.1.2 Diseño

Se debe plantear un diseño basado en la simplicidad y la facilidad de modificaciones, para esto se propone usar metáforas del sistema las cuales ayudaran al equipo de trabajo a entender mejor la finalidad de la versión correspondiente, también se deben diseñar las pruebas unitarias, las cuales deben ser creadas junto con el código ya que lo que plantea esta metodología es codificar para poder superar todas las pruebas acordadas. Es importante establecer que el diseño debe ser revisado y va a estar expuesto a cambios continuos a medida que se añadan nuevas funcionalidades al sistema.

### 3.1.3 Programación

En esta etapa se recomienda el trabajo en parejas para propiciar un intercambio de ideas. También se debe codificar para cumplir los objetivos trazados para la

---

\* La **Programación en Pares** requiere que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Se basa en la idea de que "Si se pone a dos personas juntas, se obtendrán más del doble de maneras de resolver un problema." Sin embargo, no se descarta el trabajo por separado.

versión, se debe seguir los estándares establecidos por el grupo y se dejara para el final la optimización del código.

#### 3.1.4 Pruebas

En el periodo de pruebas de aceptación se definirán las entradas del sistema y los resultados correspondientes a dichas entradas. Se recomienda automatizar las pruebas para que el sistema pueda ser probado varias veces.

## 4. SOFTWARE UTILIZADO EN EL DESARROLLO

### 4.1 SISTEMA OPERATIVO

El dominio y conocimiento del sistema operativo proporciona al usuario un enorme poder sobre todos los recursos del computador, debido a que todos los programas que se desarrollan para los computadores se basan en las características propias del sistema operativo; conocer del mismo resulta de gran ayuda al momento de usar aplicaciones comerciales y desarrollar programas propios.

El sistema operativo es el software básico que ejerce el control sobre el computador, y es el encargado de coordinar y manipular el uso del hardware entre diferentes programas de aplicación y los diferentes usuarios, además de gestionar los errores de hardware y la pérdida de datos. Los sistemas operativos realizan dos funciones diferentes: El primero es proveer una máquina virtual, es decir, un ambiente en el cual el usuario pueda ejecutar programas de manera conveniente, protegiéndolo de los detalles y complejidades del hardware. Con esta técnica se emplea espacio en el disco duro para simular la memoria adicional necesaria, sin embargo, el acceso al disco duro requiere más tiempo que el acceso a la memoria principal, por lo que el funcionamiento del ordenador resulta más lento; el segundo es administrar eficientemente los recursos del computador.

El Sistema Operativo es también un programa que actúa como intermediario entre el hardware y el usuario de un computador, y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas. El objetivo principal de un Sistema Operativo es lograr que el sistema de computación se use de manera cómoda y amigable, y el objetivo secundario es que el hardware del computador se emplee de manera eficiente.

#### 4.1.1 Sistema operativo Windows

Microsoft Windows es el nombre de una familia de sistemas operativos no libres desarrollados por la empresa de software Microsoft Corporation. Todos ellos tienen en común el estar basados en una interfaz gráfica de usuario fundamentada en el paradigma de ventanas. Las versiones de Windows que existen hasta el momento se basan en dos líneas separadas de desarrollo que finalmente convergen en una sola con la llegada de Windows XP.

Entre las principales características se puede destacar el hecho de que ofrece un ambiente de trabajo altamente gráfico e intuitivo, todos los programas y aplicaciones desarrolladas para funcionar bajo este ambiente tienen muchos

puntos en común lo cual permite la fácil adaptación de dichas aplicaciones, permite ejecutar múltiples aplicaciones al mismo tiempo y se puede intercambiar información entre ellas, su carácter gráfico lo hace ideal para programas de diseño gráfico y aplicaciones multimedia.

## 4.2 LENGUAJES Y AMBIENTES DE PROGRAMACIÓN EMPLEADOS

### 4.2.1 Visual Basic.Net

Visual Basic.Net es un ambiente de Desarrollo Rápido de Aplicaciones (RAD: Rapid Application Development) que funciona bajo el sistema operativo Windows XP. Los sistemas RAD están orientados a facilitar la productividad en el desarrollo de software.

Visual Basic.Net es un lenguaje que puede utilizarse de igual manera como lenguaje de script en el desarrollo de paginas ASP.NET, como en la creación de interfaces de usuario, componentes y servicios Web, con lo cual brinda un gran soporte para el campo de desarrollo de software.

Posee un ambiente visual de desarrollo para aplicaciones controlados por intervenciones o eventos de usuario sobre interfaces gráficas, lo cual hace que las aplicaciones desarrolladas puedan ser amigables para el usuario. Proporciona una jerarquía muy extensa de Clases de Objetos reutilizables y extensibles, con recursos de computación visuales y de procesamiento; además, permite un rápido desarrollo de aplicaciones que soporten Bases de Datos y posee una documentación abundante y una ayuda de fácil acceso.

### 4.2.2 SQL

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft Jet. SQL se utiliza para crear objetos QueryDef, como el argumento de origen del método OpenRecordSet y como la propiedad RecordSource del control de datos. También se puede utilizar con el método Execute para crear y manipular directamente las bases de datos Jet y crear consultas SQL de paso a través para manipular bases de datos remotas cliente - servidor.

#### Componentes del SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Existen tres tipos de comandos SQL:

- Los DDL (*Data Definition Language*) que permiten crear y definir nuevas bases de datos, campos e índices.
- Los DML (*Data Manipulation Language*) que permiten generar consultas para ordenar, filtrar, extraer, insertar o eliminar datos de la base de datos.
- Los DCL (*Data control Language*) que permiten administrar los permisos y los usuarios de las bases de datos.

## 5. DESARROLLO

Para el desarrollo de este proyecto, y acorde con la metodología seleccionada, se dividió el proceso completo en cuatro (4) etapas continuas, las cuales son:

1. Visualizador de capas en formato shapefile.
2. Creador de capas con formato shapefile.
3. Editor de capas con formato shapefile.
4. Cumplir las etapas anteriores para el formato PostGIS.

Cada etapa consiste en plantear los requisitos que se deben cumplir al finalizar la etapa, realizar el diseño de la etapa, efectuando modificaciones al diseño de la etapa anterior, codificar la etapa, plantear y ejecutar pruebas al programa obtenido en la etapa y analizar los resultados.

### 5.1 VISUALIZADOR

#### 5.1.1 Requisitos

- Realizar la correcta visualización de los distintos estilos de mapas digitales en formato shapefile.
- Acceder a los datos alfanuméricos asociados con la capa abierta.
- Crear las operaciones de visualización más comunes (zoom, centrar, exportar, etc.) y realizarlas de forma correcta.
- Permitir modificar los datos alfanuméricos asociados con la capa seleccionada como lo realiza un manejador de base de datos.

#### 5.1.2 Planeación

En este ciclo del proyecto se estableció como el objetivo primario a alcanzar la correcta visualización de los mapas digitales en formato shapefile. Para alcanzar dicha meta se necesitó de una etapa previa, en la cual se realizó un estudio profundo a las principales características de dicho formato, el cual se compone de 3 archivos binarios que son el .shp, .shx y .dbf; por lo tanto, la primera actividad realizada fue la interpretación de los datos binarios guardados en cada uno de estos archivos dándole prioridad al .archivo shp, que es en el cual se encuentran almacenados los datos referentes a la geometría de cada elemento del mapa. Una

vez lograda dicha actividad se procedió a desarrollar los algoritmos de visualización de los datos extraídos de este archivo; para esto se utilizaron todas las herramientas ofrecidas por el lenguaje de programación escogido.

Una vez conseguida la correcta visualización de los elementos del mapa se procedió al tratamiento del archivo .dbf en el cual están almacenados los datos asociados a cada elemento del mapa y al cual se le aplicarían todas las operaciones básicas de manejo de una base de datos tales como la inserción, eliminación y modificación de los datos. En una prueba realizada a este tipo de archivos, se observó la incompatibilidad de este formato con los manejadores de datos básicos incluidos en el entorno de programación, debido a la ausencia de una llave principal para la tabla de datos, por lo tanto se procedió a realizar un estudio más detallado de las características del archivo .dbf\* para crear un manejador propio de este formato de datos.

### 5.1.3 Diseño

Se planteo un diseño simple y sencillo, ya que la prioridad en esta fase era obtener un programa que pueda mostrar los mapas digitales en formatos shapefile, utilizando una codificación eficaz y de fácil entendimiento para cualquier programador.

Se decidió que era mejor realizar el acceso a los archivos de la capa una sola vez, y representar los datos en un formato interno de vectores de puntos y marcadores, utilizados para diferenciar unos elementos de otros. Para esto, el diseño inicial incluyó una clase que posee todos los atributos y métodos necesarios para la visualización de los mapas en este formato. Además, dentro de cada visualizador, el cual es una clase que posee una ventana de salida, se incluyen variables de control de mapa (zoom actual, numero de capas, etc.) y los objetos que pertenecen a la clase capa (uno por cada capa abierta).

Para iniciar la codificación de esta etapa, se establecieron los estándares de programación a utilizar durante todo el proyecto.

### 5.1.4 Pruebas

- Abrir capas de puntos con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile y comparar los resultados.

---

\* Véase el anexo B para observar las características y la descripción técnica del formato de archivo dBase (\*.dbf)

- Abrir capas de líneas con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile y comparar los resultados.
- Abrir capas de polígonos con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile y comparar los resultados.
- Abrir una capa de multipuntos con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile y comparar los resultados.
- Acceder a los datos asociados a las capas abiertas, y modificar su diseño y su contenido, y comprobar que el archivo de datos creado es compatible con el formato de dBASE, abriendo este archivo utilizando otra aplicación que sirva para ver capas con formato shapefile y sus datos asociados.

#### 5.1.5 Resultados esperados

- El resultado de la visualización de las capas debe ser el mismo tanto en el programa SIGUIS como en la otra aplicación seleccionada.
- El archivo de datos producido al modificar la tabla de datos desde el programa SIGUIS debe poder ser accedido utilizando cualquier otra aplicación creada para ese fin, y los datos deben ser los mismos.

#### 5.1.6 Resultados obtenidos

- El resultado de abrir las capas desde el visualizador de SIGUIS y desde otra aplicación es el mismo, por lo que se entiende que esta etapa cumple con su objetivo principal.
- El archivo de datos creado utilizando el programa SIGUIS al modificar los datos asociados a una capa, pudo ser abierto desde otra aplicación para este propósito, por lo que este archivo cumple con su formato, y se comprobó que los datos son los mismos.

## 5.2 CREADOR

### 5.2.1 Requisitos

- Proporcionar la posibilidad de crear capas en formato shapefile con los diferentes estilos soportados por la aplicación.
- Dibujar capas utilizando el estilo seleccionado por el usuario de la aplicación, obteniendo las ubicaciones de los elementos mediante el apuntador de la pantalla (Mouse).
- Guardar la capa dibujada utilizando el formato shapefile.
- Agregar los menús y elementos necesarios para un fácil acceso a la función de crear capas.

### 5.2.2 Planeación

El objetivo de esta etapa consistió en agregar la posibilidad de crear mapas en formato shapefile además de visualizarlos. Para esto, se debió revisar continuamente la definición técnica del formato shapefile, hasta conocer completamente todos los elementos que componen cada uno de los archivos. Con esta información, se procedió a crear archivos que cumplieran este formato, agregando elementos de ubicaciones fijas, para verificar que el formato se cumpliera correctamente, antes de pasar a obtener la ubicación de los elementos mediante el Mouse.

Una vez comprobada la consistencia entre el formato shapefile y los archivos creados con la aplicación, se procedió a codificar los manejadores de los eventos del Mouse, de tal forma que los datos a almacenar en los archivos fueran obtenidos de forma interactiva con el usuario. Para concluir esta etapa, se realizaron las pruebas cuyo objetivo era comparar los mapas dibujados en pantalla, con los guardados en los archivos.

### 5.2.3 Diseño

En esta etapa no se realizaron cambios en el diseño de la clase capa, ya que el proceso de creación de capas es realizado desde el visualizador. Para el desarrollo de esta etapa, se agregaron nuevas variables de control a la clase visualizador que se encargan de verificar la correcta creación de los archivos y de guardar temporalmente los datos obtenidos de la pantalla.

#### 5.2.4 Pruebas

- Crear una capa de puntos y abrirla con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile.
- Crear una capa de líneas y abrirla con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile.
- Crear una capa de polígonos y abrirla con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile.
- Crear una capa de multipuntos y abrirla con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile.

#### 5.2.5 Resultados esperados

- Las capas creadas, deben poder ser abiertas con ambas aplicaciones y los resultados de su visualización deben ser los mismos.
- Todos los archivos creados con la nueva capa deben cumplir con su propio formato y por tanto debe ser posible accederlos desde las dos aplicaciones.
- Debe ser posible ejecutar todas las operaciones de visualización una vez terminada la creación de la capa.
- La cantidad de registros creados en el archivo de datos debe corresponder con la cantidad de elementos agregados al mapa.

#### 5.2.6 Resultados obtenidos

- Todas las capas creadas con SIGUIS pudieron ser abiertas con el visualizador del programa y con otra aplicación, y la ubicación de los elementos en la pantalla es la misma.
- Todas las capas creadas con SIGUIS y sus elementos adicionales (tabla de datos) pudieron ser abiertas desde SIGUIS y desde otra aplicación, por lo que cumplen con su formato.
- Los registros en el archivo de la tabla de datos se correspondían en una relación de uno-uno con los elementos creados en el mapa.

## 5.3 EDITOR

### 5.3.1 Requisitos

- Proporcionar la posibilidad de editar capas en formato shapefile sin importar su estilo de presentación, siempre y cuando sea soportado por la aplicación.
- Permitir la edición de las capas de forma similar a la utilizada para la creación de las capas, con el fin de no confundir al usuario final.
- Guardar la capa modificada cumpliendo con el formato shapefile.
- Agregar los menús y elementos necesarios para un fácil acceso a la función de edición de capas y a sus operaciones individuales.

### 5.3.2 Planeación

Se planteó como objetivo principal para esta versión del proyecto ejecutar operaciones de edición a los mapas en formato shapefile tales como eliminar elementos, agregar elementos y desplazar elementos. Una vez alcanzadas estas actividades se planteó la operación de rehacer, deshacer y cancelar la edición; todo esto trabajando sobre el archivo .shp y .shx.

Al lograr el correcto funcionamiento de la edición de los datos dentro de estos archivos, se procedió al tratamiento del archivo .dbf en el cual se debían manipular los datos según las modificaciones realizadas al mapa (nuevos elementos agregados y elementos eliminados).

### 5.3.3 Diseño

En esta etapa no se realizaron cambios en el diseño de la clase capa, ya que el proceso de edición de capas es realizado desde el visualizador. Para desarrollar esta etapa, se agregaron nuevas variables de control a la clase visualizador que se encargan de verificar la correcta creación de los archivos y guardaban temporalmente los datos obtenidos de la pantalla y los datos que se tenían antes de empezar la edición. Sin embargo, en esta etapa se realizó una depuración de las rutinas creadas a lo largo del desarrollo del proyecto, eliminando partes de código duplicado y agrupando operaciones repetidas en nuevas rutinas para cumplir con la simplicidad requerida por la metodología del proyecto. Además, se hizo un diseño gráfico preliminar de la apariencia del producto final, para buscar un entorno amigable con el usuario final.

#### 5.3.4 Pruebas

- Editar una capa de puntos y abrirla con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile.
- Editar una capa de líneas y abrirla con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile.
- Editar una capa de polígonos y abrirla con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile.
- Editar una capa de multipuntos y abrirla con el visualizador de SIGUIS y con otra aplicación utilizada para ver capas con formato shapefile.
- Verificar la consistencia en los datos alfanuméricos asociados a cada capa y las modificaciones realizadas al mapa en cuanto a las operaciones de agregar elementos y eliminar elementos.

#### 5.3.5 Resultados esperados

- Las capas editadas, deben poder ser abiertas con ambas aplicaciones y los resultados de su visualización deben ser los mismos.
- Todos los archivos modificados con la capa deben cumplir con su propio formato y por tanto debe ser posible accederlos desde las dos aplicaciones.
- Los datos asociados a los elementos del mapa eliminados deben ser borrados del archivo .shp y se debe crear la cantidad correcta de registros que corresponda a los elementos agregados al mapa.

#### 5.3.6 Resultados obtenidos

- Todas las capas editadas con SIGUIS pudieron ser abiertas con el visualizador del programa y con otra aplicación, y la ubicación de los elementos en la pantalla es la misma.
- Todas las capas editadas con SIGUIS y sus elementos adicionales (tabla de datos) pudieron ser abiertas desde SIGUIS y desde otra aplicación, por lo que cumplen con su formato.

- Los registros en el archivo de la tabla de datos se correspondían en una relación de uno-uno con los elementos en el mapa, y los datos asociados a los elementos eliminados fueron también borrados del archivo de datos.

## 5.4 POSTGIS

### 5.4.1 Requisitos

- Proporcionar la posibilidad de abrir capas en formato PostGIS para su visualización.
- Permitir la posibilidad de crear y guardar capas en formato PostGIS del mismo modo que las capas en formato shapefile.
- Permitir al usuario seleccionar que tipo de capa abrir, y la posibilidad de tener capas de diferente formato en el mismo visualizador.
- Permitir realizar las operaciones de visualización y de modificación de datos alfanuméricos de las capas abiertas con el formato PostGIS.

### 5.4.2 Planeación

En esta etapa, el objetivo planteado fue agregar todas las funcionalidades generadas durante el desarrollo del proyecto al formato PostGIS. Para hacer esto, se realizó un estudio de este formato, para verificar sus limitaciones y los elementos compatibles con los estilos del formato shapefile. Además, se pensó en nuevas operaciones que este formato puede realizar más fácilmente que el shapefile; como resultado, se realizó el diseño y desarrollo de la función de búsqueda espacial de información\*, la cual es una gran aplicación de los mapas de este formato.

### 5.4.3 Diseño

Los cambios realizados al diseño de la clase capa se hicieron con el fin de permitir la compatibilidad con el formato PostGIS, por lo tanto, se agregaron nuevos atributos a la clase para controlar los parámetros de conexión con el servidor de PostgreSQL y se realizó sobrecarga de la función de creación de capas, para hacerla compatible con un formato u otro, dependiendo de la cantidad de

---

\* Para realizar la búsqueda de información, se desarrollo un asistente de consultas, desde el cual se puede realizar cualquier consulta de selección (compatible con SQL) sobre la tabla geográfica y generar una nueva tabla con los resultados. Véase el anexo C para observar un manual del uso de la función de selección SELECT de SQL.

parámetros utilizados. Se agregó la posibilidad de editar y guardar las capas de PostGIS, y con esto se permitió la conversión de formato PostGIS a Shapefile.

El otro código modificado fue el destinado a acceder a la información alfanumérica. Para permitir esta compatibilidad, se agregaron condiciones de verificación del formato, para utilizar un proveedor de acceso a datos u otro.

Al finalizar esta etapa, se realizó el diseño gráfico final de la apariencia del producto, pensando en el fácil acceso a todas las operaciones del programa y a la nueva operación creada.

#### 5.4.4 Pruebas

- Abrir capas de diferente geometría que se encuentren en formato PostGIS.
- Comparar la visualización de una capa en formato PostGIS, con la misma capa abierta en formato shapefile.
- Crear capas de diferente geometría y guardarlas en formato PostGIS, y verificar los datos almacenados en ella mediante un auxiliar de consulta SQL.
- Realizar modificaciones a los datos alfanuméricos asociados a una capa abierta en formato PostGIS, y verificar los cambios realizados mediante un auxiliar de consulta SQL.
- Abrir diferentes capas de los dos formatos permitidos en un mismo visualizador.
- Comprobar el funcionamiento del auxiliar de consulta SQL, con la creación de una nueva capa a partir de otra ya existente, con solo los objetos espaciales que cumplan con ciertas condiciones para sus datos alfanuméricos.

#### 5.4.5 Resultados esperados

- Se debe permitir abrir las capas seleccionadas en formato PostGIS de diferentes geometrías.
- El mapa obtenido mediante la visualización de varias capas de PostGIS y de las mismas capas en formato shapefile debe ser igual.

- La tabla creada con la nueva capa debe ser vista desde la herramienta de administración de base de datos suministrada con PostGIS (PostgreSQL).
- La tabla modificada con el programa SIGUIS debe ser vista desde la herramienta de administración de base de datos suministrada con PostGIS (PostgreSQL) y los datos deben ser los mismos.
- Los mapas de diferentes formatos deben ser permitidos dentro del mismo visualizador.
- La nueva capa creada debe poseer solo los objetos geográficos que cumplan con las restricciones planteadas.

#### 5.4.6 Resultados obtenidos

- Todas las capas creadas con SIGUIS en formato PostGIS pudieron ser abiertas con el visualizador del programa.
- La superposición de dos visualizadores, una con capas en formato PostGIS y la otra con capas en formato shapefile del mismo mapa encaja perfectamente.
- Todas las capas creadas con SIGUIS pudieron ser vistas desde la herramienta de administración de base de datos PostgreSQL.
- Todas las tablas modificadas con SIGUIS pudieron ser vistas desde la herramienta de administración de base de datos PostgreSQL y los datos eran los mismos.
- La herramienta permite la apertura en un mismo visualizador de capas en los dos formatos soportados sin presentar ningún problema de rendimiento ni de errores de dibujo.
- La nueva capa creada posee solo los objetos geográficos que cumplen con las restricciones planteadas.

## 6. CONCLUSIONES Y RECOMENDACIONES

- La metodología adoptada en el desarrollo de este proyecto fue adecuada, pues con esta se permitió desarrollar el proyecto por etapas, lo cual facilitó el diseño de cada una de ellas y el cumplimiento de los objetivos planteados para cada una.
- Durante el desarrollo del proyecto se adquirieron nuevos conocimientos sobre herramientas de desarrollo de software, lenguajes de programación, diseño de sistemas, bases de datos, todas estas áreas recibidas durante el transcurso de la carrera y reforzadas con este trabajo.
- Al interactuar con personas de otras carreras en diferentes temas, en este caso sobre los SIG, el ingeniero de sistemas debe poseer los conocimientos del funcionamiento interno de las herramientas utilizadas. Por eso en este proyecto fue importante el estudio de la estructura interna de los diferentes formatos de representación vectorial de los mapas digitales.
- En la etapa de pruebas, se hizo una comparación con otros productos del mercado, con lo cual se observó que el software realizado cumple con todas las características de un producto de calidad, lo cuál demuestra que la preparación recibida en la universidad nos da las herramientas necesarias para poder competir en el mercado mundial.

Para futuros proyectos en este mismo campo:

- Se recomienda estudiar el formato GML de una forma más detallada, pues éste posee una estructura muy compleja, pero ofrece muchas ventajas aplicadas al campo del intercambio de información geográfica vía Internet. Por lo tanto sería muy útil ampliar esta herramienta con el manejo de los mapas en este formato, y poder hacer las respectivas transformaciones entre los diferentes formatos de mapas digitales.
- Se recomienda continuar el desarrollo del software SIGUIS, agregando nuevas funcionalidades, nuevos formatos de mapas y realizando las funciones que no fueron finalizadas.
- Debido al auge de los dispositivos móviles, sería muy útil desarrollar una herramienta que pudiera trabajar en esta clase de dispositivos, la cual

proveería una gran ayuda para los diferentes usuarios que ven necesario el manejo de los datos en los sistemas de información geográfica de forma móvil.

- Hacer un estudio más detallado de las posibilidades que ofrecen los sistemas de información geográfica, ya que en la escuela no se le ha dado la importancia debida y se esta dejando de aprovechar un campo en donde el ingeniero de sistemas podría dar grandes aportes para el desarrollo de diferentes aplicaciones.

## BIBLIOGRAFÍA

AGUILAR SIERRA, Alejandro. Introducción a la Programación Extrema. México: Universidad Nacional Autónoma de México, diciembre de 2002. 9 p.

BACHMANN, Eric. Xbase Data file (\*.dbf). Documento disponible en Internet: <<http://www.clicketyclick.dk/databases/xbase/format/dbf.html>>. Actualizada en marzo de 2006.

BALENA, Francesco. Programming Microsoft Visual Basic .NET – Core Reference. Washington: Microsoft Press, 2004. 1403 p.

BROWN, David L. Library and Archives Canada: Guidelines on Computer File Types, Interchange Formats and Information Standards. Canadá: Library and Archives Canada, junio de 2004.

CALERO SOLÍS, Manuel. Una explicación de la programación extrema (XP). Madrid: V Encuentro usuarios xBase, 2003. 18 p.

ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE, INC. ESRI Shapefile Technical Description. Estados Unidos de América: An ESRI white paper, Julio de 1998. 34 p.

FERRER ZARZUELA, Jorge; ROBLES MARTÍNEZ, Gregorio. Programación eXtrema y Software Libre. V Congreso Hispalinux, Octubre de 2002.

GUZMÁN, Jacqueline; MOTZ, Regina. Integración de Información Geográfica. Uruguay: Instituto de computación - Facultad de Ingeniería, Universidad de la República, 2002.

HERNÁNDEZ HERNÁNDEZ, Guillermo. Introducción a los sistemas de información geográfica. Bucaramanga: 2005.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS. Normas Colombianas para la presentación de tesis, trabajos de grado y otros trabajos de investigación. Quinta actualización. Santafé de Bogotá D.C.: ICONTEC, 2002. 34 p. NTC 1486.

PROGRAMACIÓN EXTREMA. Documento disponible en Internet: <<http://chuidiang.iespana.es/metodologia/extrema.html>>. Actualizada en Abril de 2006.

RAMÍREZ ROMERO, Luz María. Programación extrema. México: Mayo de 2003.

SANTIAGO, Iván. Fundamentos de ArcGIS versión ArcView 9.1 – Tutorial de lecturas. Puerto Rico: Área de Tecnologías de Información Gubernamental, Oficina de Gerencia y Presupuesto, Versión 1, 2005. 157 p.

WORSLEY, John; DRAKE, Joshua. PostgreSQL Práctico. Estados Unidos: Command prompt, Inc. 2001.

## **ANEXO A. Descripción técnica ESRI Shapefile \***

Este documento define el formato de datos espaciales shapefile (.shp) y describe porque los shapefile son importantes. Se mencionan las herramientas disponibles en Environmental Systems Research Institute, Inc. (ESRI), software para la creación directa de shapefiles o para convertir datos de otros formatos en shapefiles. Este documento también proporciona toda la información técnica necesaria para escribir un programa de computador que cree shapefiles sin el uso de software de ESRI®, para organizaciones que quieran escribir sus propios intérpretes de datos.

### **¿El porqué de los Shapefiles?**

Un shapefile almacena información no topológica de la geometría y los atributos para los elementos espaciales en un conjunto de datos. La geometría para un elemento es almacenada como una figura conformada por un conjunto de coordenadas vectoriales.

Dado que los shapefile no poseen el procesamiento de la estructura topológica de datos, poseen ventajas sobre otras fuentes de datos, como mayor velocidad de dibujo y la capacidad de edición. Los shapefile manejan objetos simples que se superponen o son no contiguos. Típicamente requieren menos espacio en disco duro y son más fáciles de leer y escribir.

Los shapefile pueden soportar elementos de puntos, líneas y áreas. Los elementos de área son representados como un ciclo cerrado, polígonos doblemente digitalizados. Los atributos son guardados en un archivo de formato dBase®. Cada atributo posee una relación uno-a-uno con el registro asociado de la figura.

### **Como pueden ser creados los Shapefile**

Los shapefile pueden ser creados con los siguientes cuatro métodos generales:

- **Exportación:** Los shapefile pueden ser creados mediante la exportación de cualquier fuente de datos a shapefile utilizando el software ARC/INFO®, PC ARC/INFO®, Spatial Database Engine™ (SDE™), ArcView® GIS, o BusinessMAP™

---

\* Traducción libre y solo de elementos relevantes para el proyecto. El documento completo puede ser encontrado en la dirección de internet <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

- Digitalización: Los shapefile pueden ser creados directamente mediante la digitalización de figuras utilizando la herramienta de creación de ArcView GIS.
- Programando: Utilizando el software Avenue™ (ArcView GIS), MapObjects™, ArcMacro Language (AML™) (ARC/INFO), o Simple Macro Language (SML™) (PC ARC/INFO), se pueden crear shapefile desde su propio programa.
- Mediante la creación de un programa que escriba directamente el shapefile.

El software SDE, ARC/INFO, PC ARC/INFO, Data Automation Kit (DAK™), and ArcCAD® proveen traductores shape-coverage, y ARC/INFO provee también un traductor coverage-shape. Para intercambio con otros formatos de datos, la especificación shapefile es publicada en este documento. Otras Fuentes de datos, como aquellas de los receptores de sistema de posicionamiento global (GPS) pueden ser almacenadas como shapefile o tablas de tipo X, Y.

### **Descripción técnica de los Shapefile**

Se pueden crear programas de computadores para leer o escribir shapefile utilizando la descripción técnica de esta sección.

Un shapefile de ESRI, consiste de un archivo principal, un archivo índice, y una tabla dBASE. El archivo principal es un acceso directo, con registros de tamaño variable en el cual cada registro describe una figura como una lista de sus vértices. En el archivo índice, cada registro contiene la posición del registro correspondiente dentro del archivo principal, contado desde el comienzo del archivo principal. La tabla dBASE contiene los atributos de los elementos, con un registro por cada elemento. La relación uno-a-uno entre la geometría y los atributos está basada en el número del registro. Los registros de los atributos en el archivo dBASE deben estar en el mismo orden que los registros en el archivo principal.

### **Convenciones para los nombres**

Todos los nombres de archivo deben cumplir con la convención de nombres 8.3. El archivo principal, el archivo índice y el archivo dBASE poseen el mismo prefijo. El prefijo debe empezar con un carácter alfanumérico (a-Z, 0-9), seguido por cero o hasta siete caracteres (a-Z, 0-9, \_, -). El sufijo para el archivo principal es .shp. El sufijo para el archivo índice es .shx. El sufijo para la tabla dBASE es .dbf. Todas las letras en el nombre de un archivo están en minúscula para los sistemas operativos sensibles a los nombres de archivo (case sensitive).

Ejemplos:    Archivo principal:    counties.shp  
                   Archivo índice:     counties.shx  
                   Tabla dBASE:        counties.dbf

**Tipos numéricos**

Los shapefile almacenan números enteros y de doble precisión. El resto de este documento se referirá a los siguientes tipos:

- Entero: Número entero con signo de 32 bits (4 bytes)
- Doble: Número de coma flotante de doble precisión con signo IEEE de 64 bits (8 bytes)

Los números con coma flotante deben ser valores numéricos. Los valores infinito positivo, infinito negativo, y no numérico (NaN) no están permitidos en los shapefile. Sin embargo, los shapefile soportan el concepto de valores “sin información”, pero solo son usados para medidas. Cualquier número de coma flotante menor que  $-10^{38}$  es considerado por un lector de shapefile como la representación de un valor “sin información”.

La primera sección a continuación describe la estructura general y la organización del shapefile. La segunda sección describe el contenido de los registros para cada tipo de figura soportado dentro del shapefile.

**Organización del archivo principal**

El archivo principal (.shp) posee un encabezado de tamaño fijo seguido por registros de tamaño variable. Cada registro de tamaño variable esta constituido de un encabezado de registro de tamaño fijo, seguido por un contenido de registro de tamaño variable. La figura a continuación ilustra la organización del archivo principal.

Encabezado del archivo	
Encabezado de registro	Contenido del registro
Encabezado de registro	Contenido del registro
Encabezado de registro	Contenido del registro
Encabezado de registro	Contenido del registro
Encabezado de registro	Contenido del registro
Encabezado de registro	Contenido del registro
...	
...	
Encabezado de registro	Contenido del registro

## Orden de los bytes

Todo el contenido en un shapefile puede ser dividido en dos categorías:

- Relacionado con los datos
  - Contenido de los registros en el archivo principal.
  - Campos de datos de descripción en el encabezado del archivo principal (Tipo de figura, caja limite, etc.)
- Relacionado con la administración del archivo
  - Tamaños de registro y de archivo
  - Posición de los registros, y demás.

Los enteros y números de doble precisión que constituyen los campos de descripción de datos en la cabecera del archivo (identificados más abajo) y el contenido de los registros en el archivo principal están codificados en un orden de bytes “little endian” (PC o Intel®). Los enteros y números de doble precisión que constituyen el resto del archivo y la administración del archivo están codificados en un orden de bytes “big endian” (Sun® o Motorola®)

## El encabezado del archivo principal

El encabezado del archivo principal posee 100 bytes de longitud. La tabla a continuación muestra los campos que se encuentran en el encabezado del archivo con su posición en bytes, valor, tipo, y orden de bytes. En la tabla, la posición es con respecto al inicio del archivo.

Posición	Campo	Valor	Tipo	Orden de bytes
Byte 0	Código de archivo	9994	Entero	big endian
Byte 4	Sin uso	0	Entero	big endian
Byte 8	Sin uso	0	Entero	big endian
Byte 12	Sin uso	0	Entero	big endian
Byte 16	Sin uso	0	Entero	big endian
Byte 20	Sin uso	0	Entero	big endian
Byte 24	Tamaño del archivo	Tamaño del archivo	Entero	big endian
Byte 28	Versión	1000	Entero	little endian
Byte 32	Tipo de figura	Tipo de figura	Entero	little endian
Byte 36	Caja limite	Xmin	Doble	little endian
Byte 44	Caja limite	Ymin	Doble	little endian
Byte 52	Caja limite	Xmax	Doble	little endian
Byte 60	Caja limite	Ymax	Doble	little endian
Byte 68*	Caja limite	Zmin	Doble	little endian
Byte 76*	Caja limite	Zmax	Doble	little endian
Byte 84*	Caja limite	Mmin	Doble	little endian
Byte 92*	Caja limite	Mmax	Doble	little endian

\* Sin uso, con valor 0.0, si no es de medida o de tipo Z.

El valor del tamaño del archivo es el tamaño total del archivo en palabras de 16 bits (incluyendo las 50 palabras de 16 bits que conforman el encabezado).

Todas las figuras no nulas en un shapefile deben ser del mismo tipo. Los valores para los diferentes tipos de figura son los siguientes:

Valor	Tipo de figura
0	Nula
1	Punto
3	Polilínea
5	Polígono
8	Multipunto
11	PuntoZ
13	PolilíneaZ
15	PolígonoZ
18	MultipuntoZ
21	PuntoM
23	PolilíneaM
25	PolígonoM
28	MultipuntoM
31	MultiPatch

Los tipos de figura no especificados anteriormente (2, 4, 6, etc., y hasta 33) son reservados para uso futuro. Actualmente, los shapefile están restringidos a contener el mismo tipo de figura. En el futuro, los shapefile podrían permitir contener más de un tipo de figura. Si se implementa la mezcla de tipos de figura, el campo de tipo de figura en el encabezado mostrará que el archivo es de ese tipo.

La caja límite en el encabezado del archivo principal almacena la extensión real de las figuras en el archivo: El rectángulo mínimo de frontera ortogonal a los ejes X y Y (y potencialmente el M y Z) que contiene todas las figuras. Si el shapefile está vacío (esto es, que no posee registros), los valores para Xmin, Ymin, Xmax, Ymax son indeterminados. Mmin y Mmax pueden contener un valor “sin información” (véase los tipos numéricos) para shapefile de tipo medida que no posea medidas.

### **Encabezado de los registros**

El encabezado para cada registro almacena el número del registro y el tamaño del contenido del registro. Los encabezados de los registros tienen un tamaño fijo de 8 bytes. La tabla a continuación muestra los campos en el encabezado con su

posición en bytes, valor, tipo, y orden de bytes. En la tabla, la posición es con respecto al inicio del registro.

Posición	Campo	Valor	Tipo	Orden de bytes
Byte 0	Número del registro	Número del registro	Entero	big endian
Byte 4	Tamaño del registro	Tamaño del registro	Entero	big endian

Los números de los registros empiezan en 1.

El tamaño del registro para un elemento, es el tamaño de la sección de contenido del registro, medido en palabras de 16 bits. Por lo tanto, cada registro contribuye con (4+tamaño del contenido) palabras de 16 bits al tamaño total del archivo (almacenado en el Byte 24 del encabezado del archivo).

### Contenido de los registros del archivo principal

El contenido de los registros de los shapefile consiste del tipo de figura seguido por los datos geométricos de la figura. El tamaño del contenido de los registros depende del número de partes y vértices en una figura. Por cada tipo de figura, primero se debe describir la figura y luego el formato del contenido almacenado en disco. En las siguientes tablas el campo posición es con respecto al comienzo del contenido del registro.

### Tipos de figura en el espacio X, Y

#### *Punto*

Un punto consiste de un par de coordenadas ordenadas X, Y de doble precisión.

#### Punto

```
{
    Doble X           //Coordenada X
    Doble Y           //Coordenada Y
}
```

Posición	Campo	Valor	Tipo	Cantidad	Orden de bytes
Byte 0	Tipo de figura	1	Entero	1	little endian
Byte 4	X	X	Doble	1	little endian
Byte 12	Y	Y	Doble	1	little endian

#### *Multipunto*

Un multipunto representa un conjunto de puntos, tal como sigue:

Multipunto

```
{
    Doble[4] Caja           //Caja límite
    Entero NumPuntos       //Número de puntos
    Punto[NumPuntos] Puntos //Los puntos en el conjunto
}
```

La caja límite es almacenada en el orden Xmin, Ymin, Xmax, Ymax

Posición	Campo	Valor	Tipo	Cantidad	Orden de bytes
Byte 0	Tipo de figura	8	Entero	1	little endian
Byte 4	Caja	Caja	Doble	4	little endian
Byte 36	NumPuntos	Número de puntos	Entero	1	little endian
Byte 40	Puntos	Puntos	Punto	NumPuntos	little endian

### *Polilínea*

Una polilínea es un conjunto ordenado de vértices que consiste de una o más partes. Una parte es una secuencia ordenada de dos o más puntos. Las partes pueden o no estar conectadas una a la otra. Las partes pueden o no intersectarse las unas con las otras.

Como esta especificación no prohíbe puntos consecutivos con coordenadas idénticas, el lector de shapefile debe manejar dicho caso. Por otro lado, las partes de longitud cero que puedan resultar no están permitidas.

### Polilínea

```
{
    Doble[4] Caja           //Caja límite
    Entero NumPartes       //Número de partes
    Entero NumPuntos       //Número total de puntos
    Entero[NumPartes] Partes //Índice al primer punto de una parte
    Punto[NumPuntos] Puntos //Puntos para todas las partes
}
```

Los campos para una polilínea son descritos a continuación:

- Caja: La caja límite para la polilínea, almacenada en el orden Xmin, Ymin, Xmax, Ymax.
- NumPartes: El número de partes en la polilínea.
- NumPuntos: El número total de puntos para todas las partes.

- Partes: Un arreglo de tamaño NumPartes. Almacena, para cada polilínea, el índice de su primer punto en el arreglo de puntos. Los índices del arreglo son con respecto a cero.
- Puntos: Un arreglo de tamaño NumPuntos. Los puntos para cada parte de la polilínea están almacenados de fin a fin. Los puntos de la parte 2 siguen a los puntos de la parte 1, y así sucesivamente. El arreglo de partes almacena el índice del punto inicial para cada parte. No existe ningún delimitador en el arreglo de puntos entre las partes.

Posición	Campo	Valor	Tipo	Cantidad	Orden de bytes
Byte 0	Tipo de figura	3	Entero	1	little endian
Byte 4	Caja	Caja	Doble	4	little endian
Byte 36	NumPartes	Número de partes	Entero	1	little endian
Byte 40	NumPuntos	Número de puntos	Entero	1	little endian
Byte 44	Partes	Partes	Entero	NumPartes	little endian
Byte X	Puntos	Puntos	Punto	NumPuntos	little endian

Nota:  $X = 44 + 4 * \text{NumPartes}$

### *Polígono*

Un polígono consiste de uno o más anillos. Un anillo es una secuencia conectada de cuatro o más puntos que forman un ciclo cerrado que no se intersecta consigo mismo. Un polígono puede contener múltiples anillos exteriores. El orden de los vértices o la orientación para un anillo indica cual lado del anillo es el interior del polígono. El vecindario a la derecha de un observador caminando a lo largo del anillo en el orden de los vértices es el vecindario adentro del polígono. Los vértices de los anillos que definen huecos en los polígonos se encuentran en la dirección en contra del sentido de las manecillas del reloj. Por tanto, los vértices de un polígono con un solo anillo siempre están en orden con el sentido de las manecillas del reloj. A los anillos de un polígono se les refiere como a sus partes.

Como esta especificación no prohíbe puntos consecutivos con coordenadas idénticas, el lector de shapefile debe manejar dicha situación. Por otro lado, las partes de longitud cero que puedan resultar no están permitidas.

La estructura del polígono es idéntica a la estructura de la polilínea:

Polígono

```
{
    Doble[4] Caja           //Caja límite
    Entero NumPartes       //Número de partes
    Entero NumPuntos       //Número total de puntos
    Entero[NumPartes] Partes //Índice al primer punto de una parte
```

```
    Punto[NumPuntos] Puntos      //Puntos para todas las partes
}
```

Los campos para un polígono son descritos a continuación:

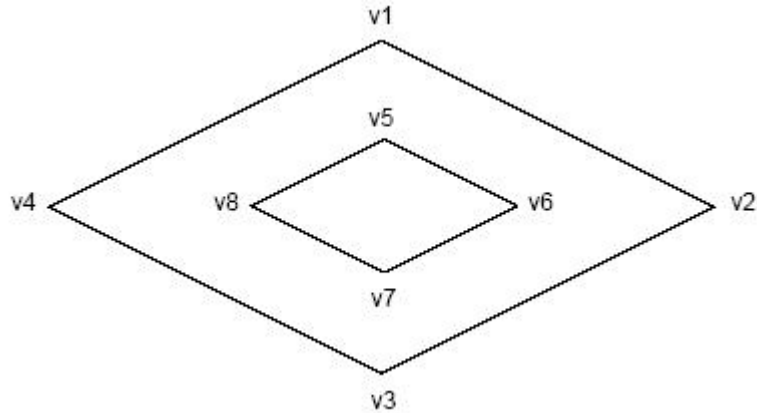
- Caja: La caja límite para la polígono, almacenada en el orden Xmin, Ymin, Xmax, Ymax.
- NumPartes: El número de anillos en el polígono.
- NumPuntos: El número total de puntos para todos los anillos.
- Partes: Un arreglo de tamaño NumPartes. Almacena, para cada anillo, el índice de su primer punto en el arreglo de puntos. Los índices del arreglo son con respecto a cero.
- Puntos: Un arreglo de tamaño NumPuntos. Los puntos para cada anillo en el polígono están almacenados de fin a fin. Los puntos del anillo 2 siguen a los puntos del anillo 1, y así sucesivamente. El arreglo de partes almacena el índice del punto inicial para cada anillo. No existe ningún delimitador en el arreglo de puntos entre los anillos.

El siguiente diagrama de ejemplo ilustra la representación de los polígonos. Esta figura muestra un polígono con un hueco y un total de ocho vértices.

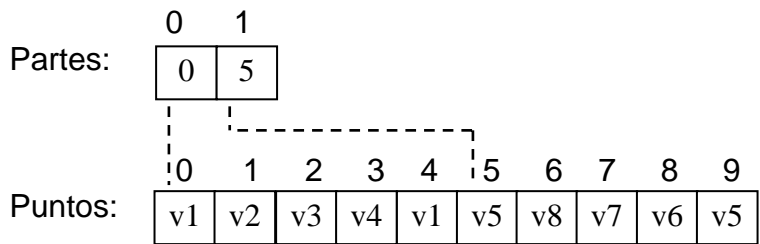
Las siguientes son notas importantes acerca de los shapefile de polígonos.

- Los anillos son cerrados (el primer y último vértice de un anillo DEBEN ser el mismo).
- El orden de los anillos en el arreglo de puntos es insignificante.
- Los polígonos almacenados en un shapefile deben ser limpios. Un polígono limpio es aquél que:
  1. No tiene intersecciones consigo mismo. Esto significa que un segmento perteneciente a un anillo no puede intersectar un segmento perteneciente a otro anillo. Los anillos de un polígono pueden tocarse entre si en los vértices, pero no entre segmentos. Segmentos colineales son considerados como intersectados.
  2. Tienen el interior del polígono en el lado “correcto” de la línea que lo define. El vecindario a la derecha de un observador caminando a lo largo del anillo en el orden de los vértices es el interior del polígono. Por lo tanto, los vértices para un polígono de un solo anillo siempre están en el orden del sentido de las manecillas del reloj. Los anillos que definen huecos en estos polígonos poseen una orientación en contra del sentido

de las manecillas del reloj. Los polígonos “sucios” son aquellos en los que los anillos que definen huecos en el polígono están también en sentido de las manecillas del reloj, lo que causa interiores superpuestos.



Para este ejemplo, NumPartes es igual a 2 y NumPuntos es igual a 10. Nótese que el orden de los puntos de la dona (hueco) del polígono están al revés a continuación.

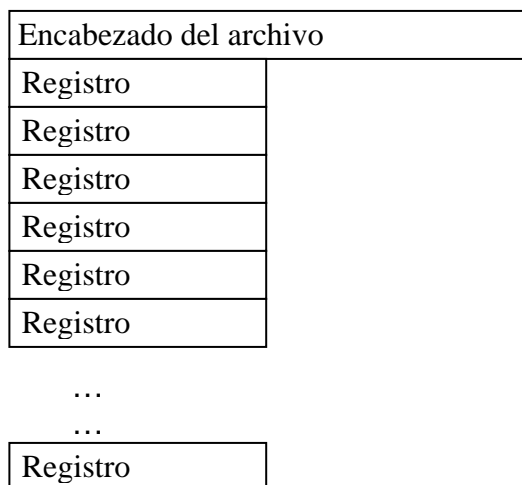


Posición	Campo	Valor	Tipo	Cantidad	Orden de bytes
Byte 0	Tipo de figura	5	Entero	1	little endian
Byte 4	Caja	Caja	Doble	4	little endian
Byte 36	NumPartes	Número de partes	Entero	1	little endian
Byte 40	NumPuntos	Número de puntos	Entero	1	little endian
Byte 44	Partes	Partes	Entero	NumPartes	little endian
Byte X	Puntos	Puntos	Punto	NumPuntos	little endian

Nota:  $X = 44 + 4 * \text{NumPartes}$

### Organización del archivo de índices

El archivo de índices (.shx) contiene un encabezado de 100 bytes seguido por registros de tamaño fijo de 8 bytes. La siguiente figura ilustra la organización del archivo de índices.



**El encabezado del archivo índice**

El encabezado del archivo es idéntico en organización al encabezado del archivo principal descrito anteriormente. El tamaño del archivo almacenado en el encabezado del archivo de índice es el tamaño total del archivo en palabras de 16 bits (las 50 palabras de 16 bits del encabezado más 4 veces el número de registros).

**Registros índices**

El i-ésimo registro en el archivo de índices almacena la posición y el tamaño del contenido del i-ésimo registro en el archivo principal. La siguiente tabla muestra los campos en el encabezado del archivo con su respectiva posición de bytes, valor, tipo y orden de bytes. En la tabla, posición es con respecto al comienzo del registro del archivo índice.

Posición	Campo	Valor	Tipo	Orden de bytes
Byte 0	Desplazamiento	Posición	Entero	little endian
Byte 4	Tamaño del contenido	Tamaño del contenido	Entero	little endian

El desplazamiento de un registro en el archivo principal es el número de palabras de 16 bits desde el comienzo del archivo principal hasta el primer byte del encabezado del registro correspondiente. De esta manera, el desplazamiento para el primer registro en el archivo principal es 50, dado el encabezado de 100 bytes.

El tamaño del contenido almacenado en el registro índice es el mismo valor almacenado en el encabezado del registro del archivo principal.

### **Organización del archivo dBASE**

El archivo dBASE (.dbf) contiene cualquier cantidad de atributos o llaves principales a los cuales se pueden unir cualquier otra tabla. Su formato es un archivo con el estándar dBASE usado por muchas aplicaciones basadas en tablas en Windows™ y DOS. Cualquier conjunto de campos puede ser presentado en la tabla. Existen tres requerimientos, que son los siguientes:

- El archivo debe tener el mismo prefijo que el archivo principal y de índice. El sufijo debe ser .dbf. (Ver el ejemplo en convenciones para los nombres).
- La tabla debe contener un registro por figura.
- El orden de los registros debe ser el mismo del archivo principal (\*.shp).
- El año en el encabezado del archivo dBASE debe ser el año desde 1900.

Para más información sobre el formato visitar la página Web de la empresa.

## **ANEXO B. dBASE: Definición técnica**

El formato de archivo dBASE es ampliamente usado para la transferencia de archivos entre bases de datos. Este formato fue creado originalmente para los programas de bases de datos dBASE. Un archivo dBASE posee una extensión .dbf, y se compone de tres partes principales: El encabezado, el descriptor de los campos y los registros.

### **Encabezado**

El encabezado es un conjunto fijo de 32 bytes que contiene información básica acerca del archivo dBASE: Su formato (versión del archivo), la fecha en la cual fue modificado, el número de registros que posee y el tamaño de cada registro, entre otros. Esta información se encuentra codificada en binario, por lo que no puede ser leída mediante un procesador de palabras, ni puede ser sobrescrita directamente.

### **Descriptor de los campos**

El descriptor de los campos es una serie de bloques de 32 bytes (seguido por un carácter de nueva línea) que contiene información acerca de los campos. Esta información consiste de: El nombre del campo, tipo de campo (numérico, fecha, etc.), tamaño y precisión (número de decimales), entre otros. Solo se reservan 10 bytes para el nombre del campo, por lo que este solo puede poseer hasta 10 letras.

### **Registros**

Los registros son una serie de bloques de tamaño fijo que contienen los datos. Los datos se encuentran codificados en ASCII, por lo que se pueden abrir desde cualquier procesador de palabras, siempre y cuando se pueda separar del encabezado y el descriptor de los campos que los preceden. No existe nada que separe un bloque de otro, por lo que su lectura desde un editor normal puede llegar a ser como la de datos sin mucho sentido.

### **Esquema de un archivo dBASE**

A continuación se muestra un esquema con todos los campos que debe poseer un archivo para cumplir con el formato dBASE III.

Byte	Descripción	Parte	
0	Versión (Firma)	<b>ENCABEZADO</b>	
1	Fecha de última actualización YYMMDD* <sup>1</sup>		
2			
3			
4			
5	Número de registros en el archivo de datos (32 bits)		
6			
7			
8			
9	Longitud del encabezado (16 bits)		
10	Longitud de cada registro (16 bits)		
11			
12	(Reservado)		
13			
14	Transacción incompleta		
15	Señal de encriptación (no/si)		
16	Registro libre de procesos (Reservado únicamente para LAN)		
17			
18			
19			
20	(Reservado para dBASE multiusuario) (dBASE III+ -)		
21			
22			
23			
24			
25			
26			
27			
28	Señal MDX (dBASE IV)		
29	Lenguaje de codificación		
30	(Reservado)		
31			
<b>Por cada campo</b>	0	Nombre del campo (ASCII) (Terminado con 00h)	<b>DESCRIPTOR DE LOS CAMPOS</b>
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	8		
	9		

<b>Por cada campo</b>	10		<b>DESCRIPTOR DE LOS CAMPOS</b>
	11	Tipo de campo (ASCII) <sup>*2</sup>	
	12	Dirección del campo (en la memoria) (dBASE III+)	
	13		
	14		
	15		
	16	Tamaño del campo <sup>*3</sup>	
	17	Cantidad de decimales <sup>*4</sup>	
	18	(Reservado para dBASE multiusuario)	
	19		
	20	ID del área de trabajo (es 01h)	
	21	(Reservado para dBASE multiusuario)	
	22		
	23	Señal para SET FIELDS	
	24	(Reservado)	
	25		
	26		
27			
28			
29			
30			
31	Señal de índice del campo		
	Finalizador (ODh)		
...	<b>BLOQUE DE REGISTROS</b>	<b>REGISTROS</b>	
...			
...			
...			
...			
...			
...			
...	Fin del archivo (1Ah)		

\*1. Formato de fechas: Las fechas en el encabezado del archivo se almacenan sin el siglo (YYMMDD) y las fechas en los registros se almacenan con el siglo (YYYYMMDD). Los intervalos válidos van desde 00h hasta FFh (de 0 a 255). Al agregar el año base 1900 se obtiene el intervalo de años 1900 - 2155.

\*2. Tipos de datos: dBASE III puede manejar los tipos de datos más comunes, y otros propios del sistema. Para el desarrollo de este trabajo, se escogieron los tipos de datos básicos de todos los sistemas de bases de datos: Numérico, texto, fecha y lógico. Estas son sus características en el sistema dBASE.

<b>Código</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
C	Texto	<254	Texto ASCII de menos de 254 caracteres de longitud.
N	Numérico	<18	Texto ASCII de hasta 18 caracteres de longitud (incluyendo el signo y el punto decimal). Valores válidos incluyen los números de 0 a 9, el signo – y el punto (.).
L	Lógico	1	Tipo booleano. Sus valores posibles son Y, y, T, t para definir TRUE, F, f, N, n para definir FALSE y ? para definir indeterminado o nulo.
D	Fecha	8	Fecha en el formato YYYYMMDD. Una fecha en el formato 0000-00-00 no es válida.

\*3. Tamaño del campo: Valor máximo 255. Cada tipo de campo tiene un valor fijo, véase el numeral anterior.

\*4. Cantidad de decimales: Para campos numéricos, máximo 15.

#### **Algunas características adicionales de los archivos dBASE**

- Cada valor de cada campo tiene su espacio reservado. Algo debe ir en ese espacio, por lo tanto, no existe algo así como un valor vacío o nulo. Esta conversión la debe realizar el programa intérprete del archivo.
- Literalmente cualquier cosa puede ir en los registros, por ejemplo caracteres en campos numéricos. La validación de los datos depende de la verificación del software sobre los tipos de datos.
- No existe información especial que restrinja el tipo de dato que puede ir en un campo, por ejemplo, no existe ninguna manera de especificar que un campo numérico solo puede contener valores positivos.

## ANEXO C. Consultas SQL

Las consultas de selección se utilizan para indicar al motor de datos que devuelva información de las bases de datos, esta información es devuelta en forma de conjunto de registros que se pueden almacenar en un objeto recordset. Este conjunto de registros es modificable.

La sintaxis básica de una consulta de selección es la siguiente:

```
SELECT [lista de campos] FROM Tabla;
```

Adicionalmente se puede especificar el orden en que se desean recuperar los registros de las tablas mediante la cláusula ORDER BY [Lista de Campos]. En donde Lista de campos representa los campos a ordenar

```
SELECT [lista de campos] FROM Tabla ORDER BY [lista de campos];
```

Incluso se puede especificar el orden de los registros: ascendente mediante la cláusula (ASC -se toma este valor por defecto) ó descendente (DESC)

### Consultas con Predicado

El predicado se incluye entre la cláusula y el primer nombre del campo a recuperar, los posibles predicados son:

Predicado	Descripción
ALL	Devuelve todos los campos de la tabla
TOP	Devuelve un determinado número de registros de la tabla
DISTINCT	Omite los registros cuyos campos seleccionados coincidan totalmente
DISTINCTROW	Omite los registros duplicados basándose en la totalidad del registro y no sólo en los campos seleccionados.

- ALL  
Si no se incluye ninguno de los predicados se asume ALL. El Motor de base de datos selecciona todos los registros que cumplen las condiciones de la instrucción SQL. No es conveniente abusar de este predicado ya que obligamos al motor de la base de datos a analizar la estructura de la tabla para averiguar los campos que contiene, es mucho más rápido indicar el listado de campos deseados.

```
SELECT ALL FROM Tabla;  
SELECT * FROM Tabla;
```

- TOP

Devuelve un cierto número de registros que se encuentran al principio o al final de un rango especificado por una cláusula ORDER BY. Supongamos que queremos recuperar los nombres de los 25 primeros estudiantes del curso:

```
SELECT TOP 25 Nombre, Apellido FROM Estudiantes  
ORDER BY Nota DESC;
```

Si no se incluye la cláusula ORDER BY, la consulta devolverá un conjunto arbitrario de 25 registros de la tabla Estudiantes. El predicado TOP no elige entre valores iguales. En el ejemplo anterior, si la nota media número 25 y la 26 son iguales, la consulta devolverá 26 registros. Se puede utilizar la palabra reservada PERCENT para devolver un cierto porcentaje de registros que se encuentre al principio o al final de un rango especificado por la cláusula ORDER BY. Supongamos que en lugar de los 25 primeros estudiantes deseamos el 10 por ciento del curso:

```
SELECT TOP 10 PERCENT Nombre, Apellido FROM Estudiantes  
ORDER BY Nota DESC;
```

El valor que va a continuación de TOP debe ser de tipo entero sin signo. TOP no afecta a la posible actualización de la consulta.

- DISTINCT

Omite los registros que contienen datos duplicados en los campos seleccionados. Para que los valores de cada campo listado en la instrucción SELECT se incluyan en la consulta deben ser únicos.

Por ejemplo, varios empleados listados en la tabla Empleados pueden tener el mismo apellido. Si dos registros contienen López en el campo Apellido, la siguiente instrucción SQL devuelve un único registro:

```
SELECT DISTINCT Apellido FROM Empleados;
```

Con otras palabras el predicado DISTINCT devuelve aquellos registros cuyos campos indicados en la cláusula SELECT posean un contenido diferente. El resultado de una consulta que utiliza DISTINCT no es actualizable y no refleja los cambios subsiguientes realizados por otros usuarios.

- **DISTINCTROW**

Devuelve los registros diferentes de una tabla; a diferencia del predicado anterior que sólo se fijaba en el contenido de los campos seleccionados, éste lo hace en el contenido del registro completo independientemente de los campos indicados en la cláusula SELECT.

```
SELECT DISTINCTROW Apellido FROM Empleados;
```

Si la tabla empleados contiene dos registros: Antonio López y Marta López el ejemplo del predicado DISTINCT devuelve un único registro con el valor López en el campo Apellido ya que busca no duplicados en dicho campo. Este último ejemplo devuelve dos registros con el valor López en el apellido ya que se buscan no duplicados en el registro completo.

### **Alias**

En determinadas circunstancias es necesario asignar un nombre a alguna columna determinada de un conjunto devuelto, otras veces por simple capricho o por otras circunstancias. Para resolver todas ellas, se cuenta con la palabra reservada AS que se encarga de asignar el nombre que deseamos a la columna deseada.

```
SELECT DISTINCTROW campo AS alias FROM Tabla;
```

### **Recuperar información de una base de datos externa**

En ocasiones, es necesaria la recuperación de información que se encuentra contenida en una tabla que no pertenece a la base de datos que ejecutará la consulta o que en ese momento se encuentra abierta, esta situación se puede salvar con la palabra reservada IN de la siguiente forma:

```
SELECT DISTINCTROW Apellido AS Empleado FROM Empleados IN 'C:\databases\gestion.mdb';
```

En donde C:\databases\gestion.mdb es la base de datos que contiene la tabla Empleados.

### **Criterios de Selección**

Hay que recalcar tres detalles de vital importancia. El primero de ellos es que cada vez que se desee establecer una condición referida a un campo de texto la condición de búsqueda debe ir encerrada entre comillas simples; la segunda es que no es posible establecer condiciones de búsqueda en los campos de tipo memo; y la tercera hace referencia a las fechas. Las fechas se deben escribir

siempre en formato mm-dd-aa en donde mm representa el mes, dd el día y aa el año, hay que prestar atención a los separadores -no sirve la separación habitual de la barra (/), hay que utilizar el guión (-) y además la fecha debe ir encerrada entre almohadillas (#). Por ejemplo si deseamos referirnos al día 3 de Septiembre de 1995 deberemos hacerlo de la siguiente forma; #09-03-95# ó #9-3-95#.

- Operadores Lógicos  
Los operadores lógicos soportados por SQL son: AND, OR, XOR, Eqv, Imp, Is y Not. A excepción de los dos últimos todos poseen la siguiente sintaxis:

<expresión1> operador <expresión2>

En donde expresión1 y expresión2 son las condiciones a evaluar, el resultado de la operación varía en función del operador lógico. La siguiente tabla muestra los diferentes posibles resultados:

<expresión1>	Operador	<expresión2>	Resultado
Verdad	AND	Falso	Falso
Verdad	AND	Verdad	Verdad
Falso	AND	Verdad	Falso
Falso	AND	Falso	Falso
Verdad	OR	Falso	Verdad
Verdad	OR	Verdad	Verdad
Falso	OR	Verdad	Verdad
Falso	OR	Falso	Falso
Verdad	XOR	Verdad	Falso
Verdad	XOR	Falso	Verdad
Falso	XOR	Verdad	Verdad
Falso	XOR	Falso	Falso
Verdad	Eqv	Verdad	Verdad
Verdad	Eqv	Falso	Falso
Falso	Eqv	Verdad	Falso
Falso	Eqv	Falso	Verdad
Verdad	Imp	Verdad	Verdad
Verdad	Imp	Falso	Falso
Verdad	Imp	Null	Null
Falso	Imp	Verdad	Verdad

Falso	Imp	Falso	Verdad
Falso	Imp	Null	Verdad
Null	Imp	Verdad	Verdad
Null	Imp	Falso	Null
Null	Imp	Null	Null

Si a cualquiera de las anteriores condiciones le anteponemos el operador NOT el resultado de la operación será el contrario al devuelto sin el operador NOT.

El último operador denominado Is se emplea para comparar dos variables de tipo objeto <Objeto1> Is <Objeto2>. este operador devuelve verdad si los dos objetos son iguales

- Intervalos de Valores  
Para indicar que deseamos recuperar los registros según el intervalo de valores de un campo, se emplea el operador Between, cuya sintaxis es:

campo [Not] Between valor1 And valor2 (la condición Not es opcional)

En este caso la consulta devolvería los registros que contengan en "campo" un valor incluido en el intervalo valor1, valor2 (ambos inclusive). Si anteponemos la condición Not devolverá aquellos valores no incluidos en el intervalo.

- El Operador Like  
Se utiliza para comparar una expresión de cadena con un modelo en una expresión SQL. Su sintaxis es:

expresión Like modelo

En donde expresión es una cadena modelo o campo contra el que se compara expresión. Se puede utilizar el operador Like para encontrar valores en los campos que coincidan con el modelo especificado. Por modelo se puede especificar un valor completo, o se pueden utilizar caracteres comodín como los reconocidos por el sistema operativo para encontrar un rango de valores (Like An\*).

El operador Like se puede utilizar en una expresión para comparar un valor de un campo con una expresión de cadena. Por ejemplo, si introduce Like C\* en una consulta SQL, la consulta devuelve todos los valores de campo

que comiencen por la letra C. En una consulta con parámetros, puede hacer que el usuario escriba el modelo que se va a utilizar.

En la tabla siguiente se muestra cómo utilizar el operador Like para comprobar expresiones con diferentes modelos.

Tipo de coincidencia	Modelo Planteado	Coincide	No coincide
Varios caracteres	'a*a'	'aa', 'aBa', 'aBBBa'	'aBC'
Carácter especial	'a[*]a'	'a*a'	'aaa'
Varios caracteres	'ab**'	'abcdefg', 'abc'	'cab', 'aab'
Un solo carácter	'a?a'	'aaa', 'a3a', 'aBa'	'aBBBa'
Un solo dígito	'a#a'	'a0a', 'a1a', 'a2a'	'aaa', 'a10a'
Rango de caracteres	'[a-z]'	'f', 'p', 'j'	'2', '&'
Fuera de un rango	'![a-z]'	'9', '&', '%'	'b', 'a'
Distinto de un dígito	'![0-9]'	'A', 'a', '&', '~'	'0', '1', '9'
Combinada	'a[!b-m]#'	'an9', 'az0', 'a99'	'abc', 'aj0'

- El Operador In  
Este operador devuelve aquellos registros cuyo campo indicado coincide con alguno de los en una lista. Su sintaxis es:

expresión [Not] In(valor1, valor2, . . .)

```
SELECT * FROM Pedidos WHERE Provincia In ('Madrid', 'Barcelona', 'Sevilla');
```

- La cláusula WHERE  
La cláusula WHERE puede usarse para determinar cuales registros de las tablas enumeradas en la cláusula FROM aparecerán en los resultados de la instrucción SELECT. WHERE es opcional, pero cuando aparece debe ir a continuación de FROM.

SELECT [lista de campos] FROM Tabla WHERE Condición

## Agrupamiento de Registros

- GROUP BY  
Combina los registros con valores idénticos, en la lista de campos especificados, en un único registro. Para cada registro se crea un valor

sumario si se incluye una función SQL agregada, como por ejemplo Sum o Count, en la instrucción SELECT. Su sintaxis es:

```
SELECT campos FROM tabla WHERE criterio GROUP BY campos_grupo
```

GROUP BY es opcional. Los valores de resumen se omiten si no existe una función SQL agregada en la instrucción SELECT. Los valores Null en los campos GROUP BY se agrupan y no se omiten. No obstante, los valores Null no se evalúan en ninguna de las funciones SQL agregadas. Se utiliza la cláusula WHERE para excluir aquellas filas que no desea agrupar, y la cláusula HAVING para filtrar los registros una vez agrupados. A menos que contenga un dato Memo u Objeto OLE, un campo de la lista de campos GROUP BY puede referirse a cualquier campo de las tablas que aparecen en la cláusula FROM, incluso si el campo no está incluido en la instrucción SELECT, siempre y cuando la instrucción SELECT incluya al menos una función SQL agregada.

Todos los campos de la lista de campos de SELECT deben o bien incluirse en la cláusula GROUP BY o como argumentos de una función SQL agregada.

Una vez que GROUP BY ha combinado los registros, HAVING muestra cualquier registro agrupado por la cláusula GROUP BY que satisfaga las condiciones de la cláusula HAVING.

HAVING es similar a WHERE, determina qué registros se seleccionan. Una vez que los registros se han agrupado utilizando GROUP BY, HAVING determina cuáles de ellos se van a mostrar.

## Funciones agregadas

- **AVG**  
Calcula la media aritmética de un conjunto de valores contenidos en un campo especificado de una consulta. Su sintaxis es la siguiente

Avg(expr)

En donde expr representa el campo que contiene los datos numéricos para los que se desea calcular la media o una expresión que realiza un cálculo utilizando los datos de dicho campo. La media calculada por Avg es la media aritmética (la suma de los valores dividido por el número de valores). La función Avg no incluye ningún campo Null en el cálculo.

- **Count**  
Calcula el número de registros devueltos por una consulta. Su sintaxis es la siguiente

Count(expr)

En donde expr contiene el nombre del campo que desea contar. Los operandos de expr pueden incluir el nombre de un campo de una tabla, una constante o una función (la cual puede ser intrínseca o definida por el usuario pero no otras de las funciones agregadas de SQL). Puede contar cualquier tipo de datos incluso texto.

Aunque expr puede realizar un cálculo sobre un campo, Count simplemente cuenta el número de registros sin tener en cuenta qué valores se almacenan en los registros. La función Count no cuenta los registros que tienen campos null a menos que expr sea el carácter comodín asterisco (\*). Si utiliza un asterisco, Count calcula el número total de registros, incluyendo aquellos que contienen campos null. Count(\*) es considerablemente más rápida que Count(Campo). No se debe poner el asterisco entre dobles comillas ('\*').

Si expr identifica a múltiples campos, la función Count cuenta un registro sólo si al menos uno de los campos no es Null. Si todos los campos especificados son Null, no se cuenta el registro. Hay que separar los nombres de los campos con ampersand (&).

- **Max, Min**  
Devuelven el mínimo o el máximo de un conjunto de valores contenidos en un campo específico de una consulta. Su sintaxis es:

Min(expr)  
Max(expr)

En donde expr es el campo sobre el que se desea realizar el cálculo. Expr puede incluir el nombre de un campo de una tabla, una constante o una función (la cual puede ser intrínseca o definida por el usuario pero no otras de las funciones agregadas de SQL).

- **StDev, StDevP**  
Devuelve estimaciones de la desviación estándar para la población (el total de los registros de la tabla) o una muestra de la población representada (muestra aleatoria). Su sintaxis es:

StDev(expr)  
StDevP(expr)

En donde expr representa el nombre del campo que contiene los datos que desean evaluarse o una expresión que realiza un cálculo utilizando los datos de dichos campos. Los operandos de expr pueden incluir el nombre de un campo de una tabla, una constante o una función (la cual puede ser intrínseca o definida por el usuario pero no otras de las funciones agregadas de SQL).

StDevP evalúa una población, y StDev evalúa una muestra de la población. Si la consulta contiene menos de dos registros (o ningún registro para StDevP), estas funciones devuelven un valor Null (el cual indica que la desviación estándar no puede calcularse).

- Sum  
Devuelve la suma del conjunto de valores contenido en un campo específico de una consulta. Su sintaxis es:

Sum(expr)

En donde expr representa el nombre del campo que contiene los datos que desean sumarse o una expresión que realiza un cálculo utilizando los datos de dichos campos. Los operandos de expr pueden incluir el nombre de un campo de una tabla, una constante o una función (la cual puede ser intrínseca o definida por el usuario pero no otras de las funciones agregadas de SQL).

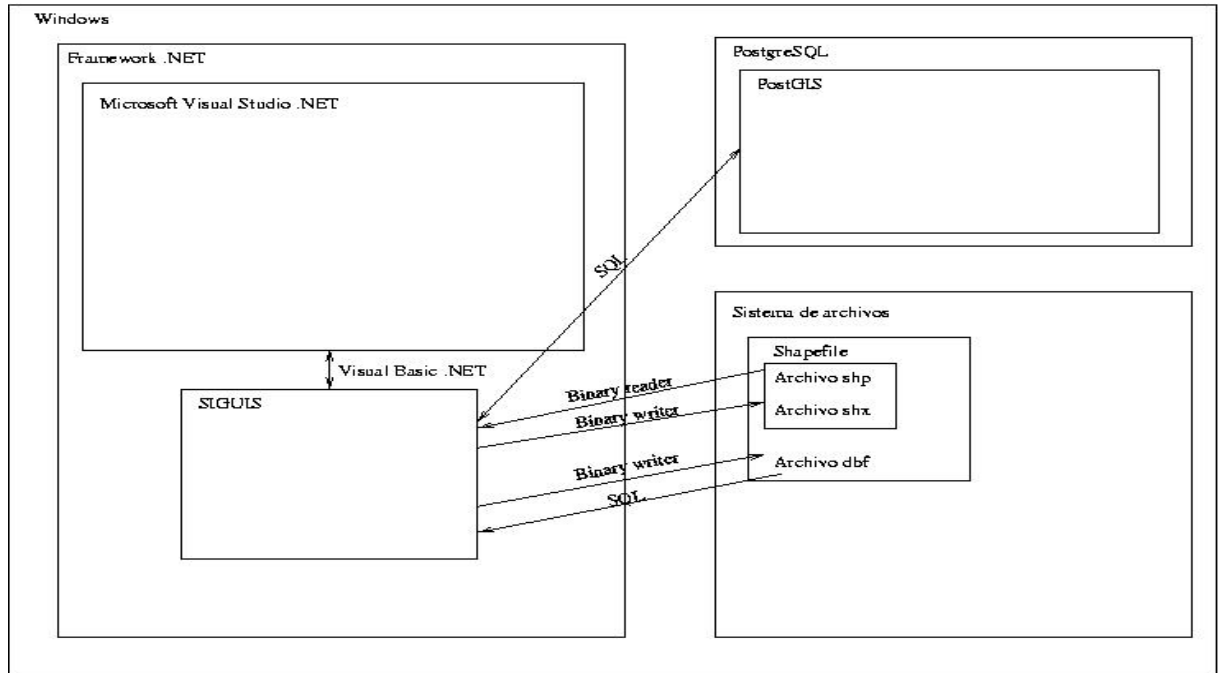
- Var, VarP  
Devuelve una estimación de la varianza de una población (sobre el total de los registros) o una muestra de la población (muestra aleatoria de registros) sobre los valores de un campo. Su sintaxis es:

Var(expr)  
VarP(expr)

VarP evalúa una población, y Var evalúa una muestra de la población. Expr representa el nombre del campo que contiene los datos que desean evaluarse o una expresión que realiza un cálculo utilizando los datos de dichos campos. Los operandos de expr pueden incluir el nombre de un campo de una tabla, una constante o una función (la cual puede ser intrínseca o definida por el usuario pero no otras de las funciones agregadas de SQL).

Si la consulta contiene menos de dos registros, Var y VarP devuelven Null (esto indica que la varianza no puede calcularse). Puede utilizar Var y VarP en una expresión de consulta o en una Instrucción SQL.

## ANEXO D. Estructura del programa



El programa SIGUIS se encuentra sostenido sobre el Framework.Net de Microsoft, del cual utiliza sus funciones, métodos y objetos para un correcto funcionamiento. A su vez, este espacio de trabajo depende del sistema operativo Windows, el cual es requerido para la correcta utilización del programa. El entorno en el que fue desarrollado el programa es el Visual Studio .Net de Microsoft y el lenguaje seleccionado, de los posibles para este entorno, es el Visual Basic .Net, del cual se encuentra una gran cantidad de documentación y ayuda en Internet y en otros medios.

Las bases de datos espaciales (PostGIS) son mantenidas por el sistema gestor de base de datos PostgreSQL. Para el acceso a estas tablas y a sus datos, se utiliza SQL como lenguaje de consulta y de inserción y actualización de datos.

El formato de mapas Shapefile se encuentra directamente sobre el sistema de directorios de Windows. Este formato se compone de archivos binarios, por lo que para acceder a ellos y modificarlos se utilizaron los objetos Lector binario (Binary Reader) y Escritor binario (Binary writer), propios del lenguaje. Los datos alfanuméricos de estos mapas se encuentran en una tabla de dBASE, por lo que su lectura es llevada a cabo mediante SQL.

## **ANEXO E. Manual de instalación de SIGUIS**

La instalación del programa SIGUIS es bastante sencilla con ayuda del asistente de instalación creado para este propósito. Sin embargo, los requisitos previos son importantes para la correcta ejecución del programa, y por tanto el proceso de instalación y configuración de estos programas es explicado a continuación.

### **Requisitos de Hardware**

El programa SIGUIS requiere para su correcto funcionamiento:

- Computador con un procesador mayor de 600 MHz.
- 128 MB de memoria RAM, se recomienda 256 MB o más.
- 10 MB de espacio disponible en el disco duro (después de instalar los requisitos de software).
- Teclado y Mouse compatible con el sistema operativo Microsoft Windows.

### **Requisitos de software**

Para el correcto funcionamiento del programa SIGUIS, se debe tener instalado en el computador en el que se desee utilizar el programa el Framework.Net de Microsoft Corporation. Además, para utilizar la compatibilidad del programa con el formato PostGIS se debe tener instalado el servidor de base de datos PostgreSQL de The PostgreSQL Global Development Group.

El sistema operativo bajo el cual se puede utilizar correctamente SIGUIS es alguno que soporte el Framework.Net y PostgreSQL, los cuales son Microsoft Windows Vista, Microsoft Windows XP, Microsoft Windows XP SP2, Windows 2000 SP4, Microsoft Windows Server 2003 SP1, Windows Server 2003 R2. Se recomienda Windows XP, ya que desde este se realizaron todas las pruebas, sin embargo, la compatibilidad es la misma.

### **Instalación del Framework.NET**

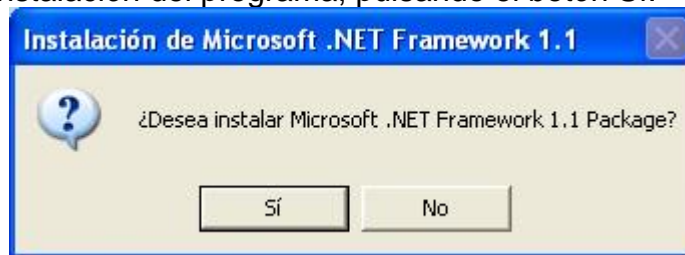
La versión del framework.net requerida para la ejecución del programa SIGUIS es la 1.1.4322.x o superior, pero menor que la 2.x. Este programa puede ser obtenido desde la página en Internet de Microsoft Corporation (<http://www.microsoft.com>) o en el CD de SIGUIS se entrega la versión necesaria de este programa.

El proceso para la instalación de Microsoft Framework.Net es el siguiente:

1. Ejecutar el archivo Microsoft Framework.exe que se encuentra en el CD en la carpeta Requisitos\Framework, o el archivo descargado desde Internet.



2. Aceptar la instalación del programa, pulsando el botón Sí.



3. Esperar a que el programa se instale automáticamente. El tiempo de instalación puede cambiar dependiendo de la configuración del computador.



4. La instalación ha concluido correctamente, en caso contrario, se debe revisar la compatibilidad del sistema operativo y el espacio en disco duro, e intentarlo nuevamente.

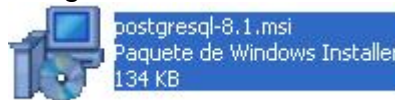


### **Instalación de PostgreSQL y PostGIS**

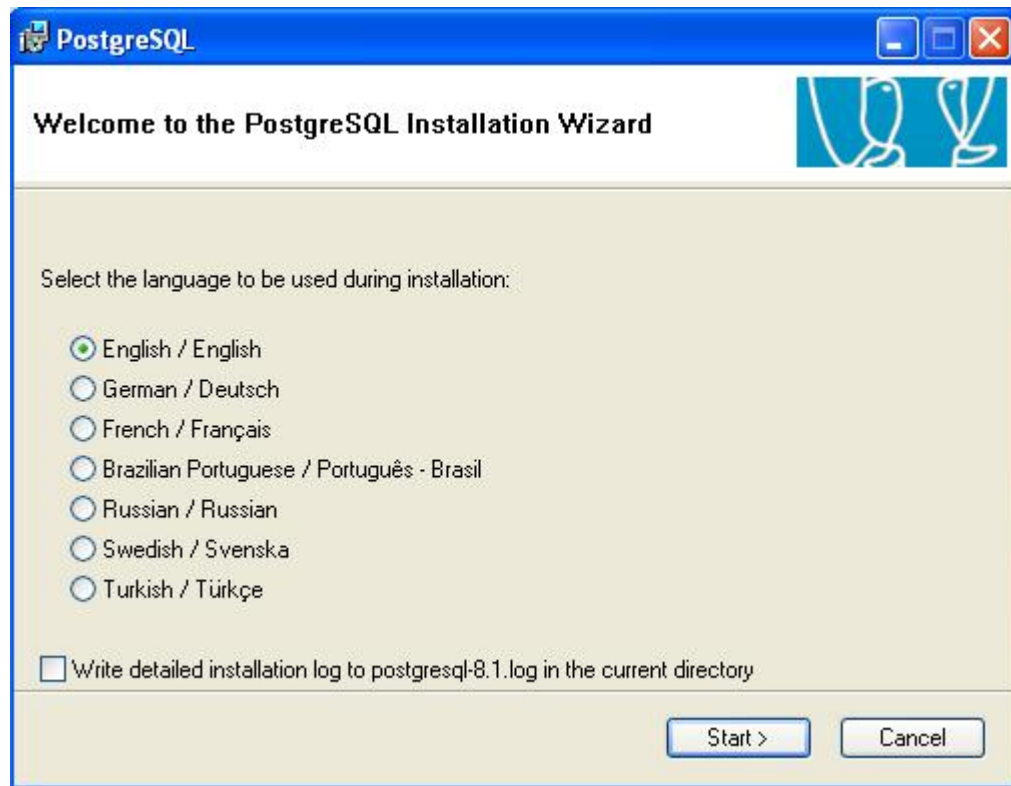
La versión de PostgreSQL requerida para la realización de operaciones con el formato PostGIS es alguna que soporte la extensión PostGIS y funcione bajo el sistema operativo Microsoft Windows, es decir, la versión 8.0 en adelante. Se recomienda la versión 8.1, que posee un entorno más amigable y nuevas funciones para los datos geométricos. Este programa puede ser obtenido desde la página en Internet de PostgreSQL (<http://www.postgresql.org>) o en el CD de SIGUIS se entrega la versión necesaria de este programa.

El proceso para la instalación de PostgreSQL con la extensión PostGIS es el siguiente:

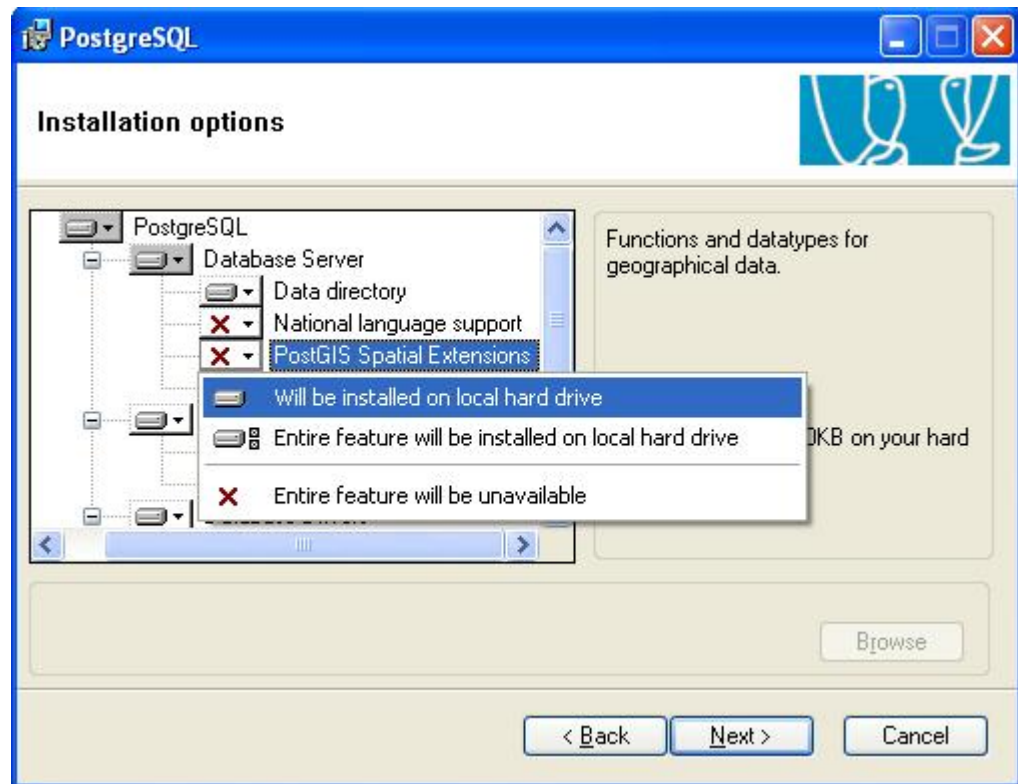
1. Ejecutar el archivo Microsoft Framework.exe que se encuentra en el CD en la carpeta Requisitos\PostgreSQL, o el archivo descargado desde Internet.



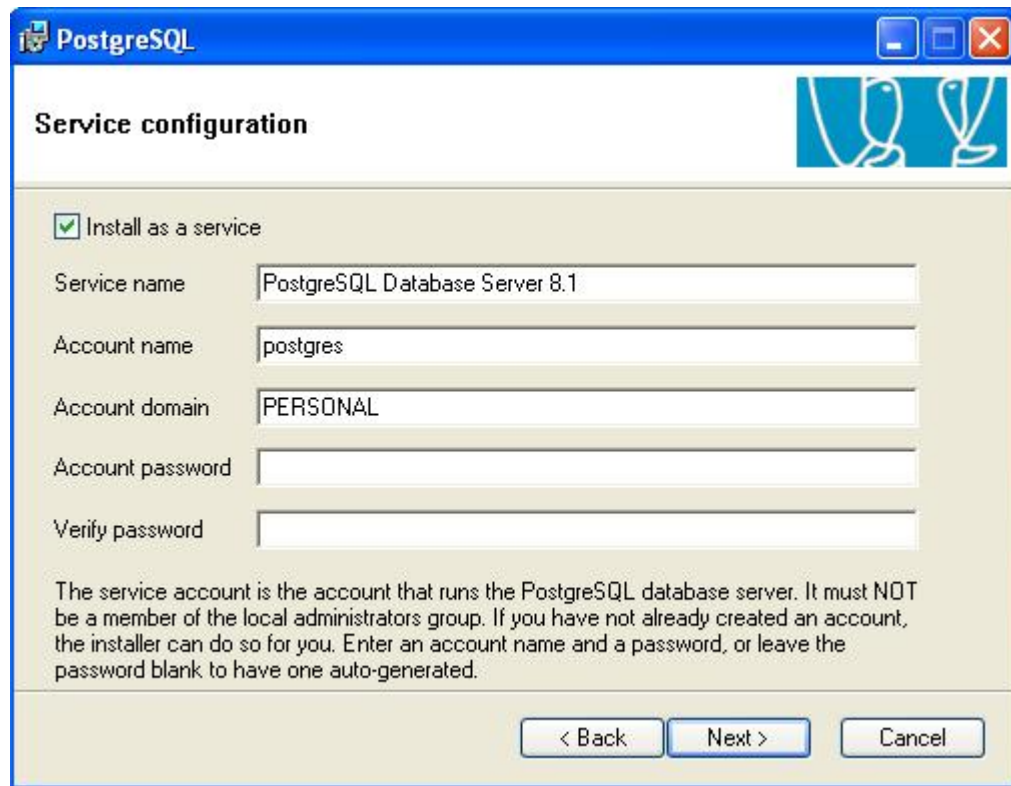
2. Seleccionar el idioma adecuado para la instalación y pulsar el botón Start.



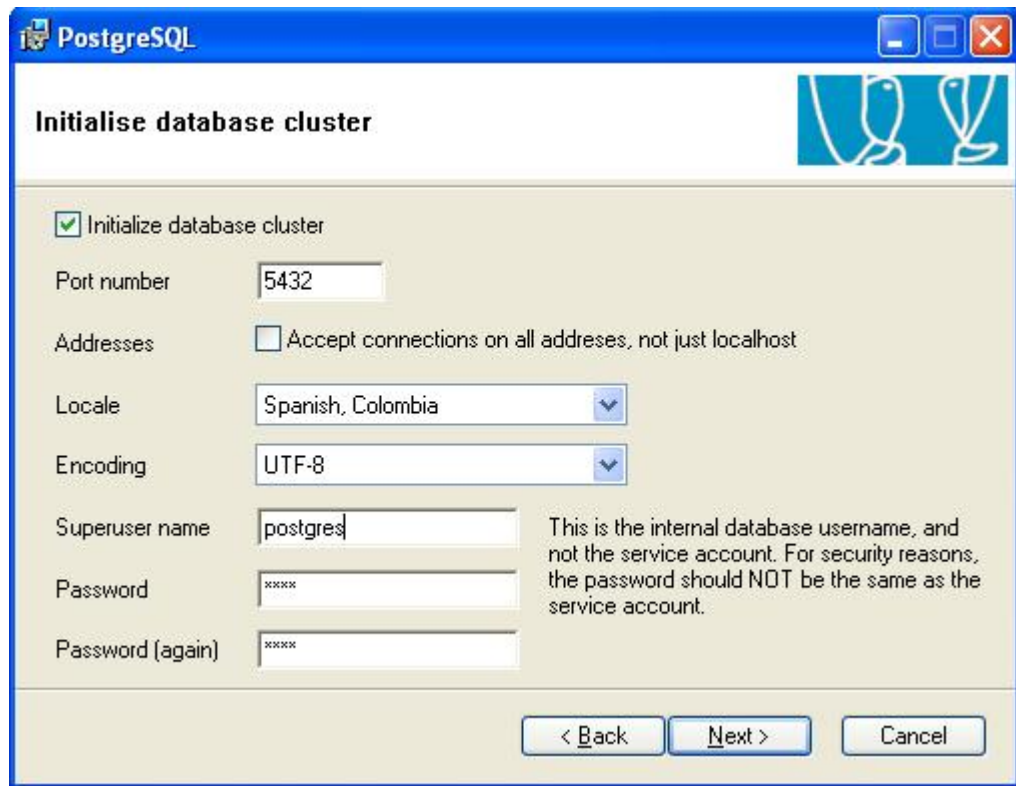
3. Pulsar el botón Next en la pantalla de inicio. y luego leer las notas de instalación, y pulsar el botón Next.
4. Seleccionar los elementos que desea instalar. En este punto es muy importante seleccionar la extensión espacial PostGIS (Pulsando clic sobre la opción Will be installed on local hard drive), ya que ésta es requerida para el funcionamiento de SIGUIS con este formato. Los demás elementos pueden ser dejados como se encuentran por defecto, o escoger una instalación completa para aprovechar el programa PostgreSQL.



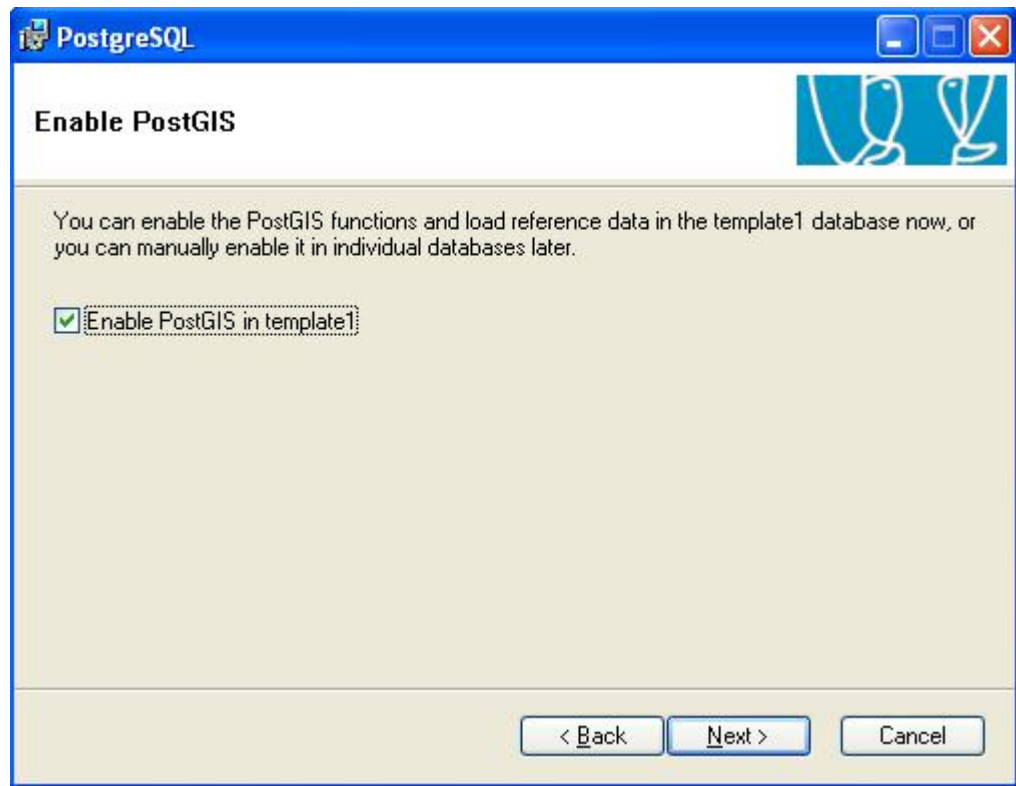
5. Luego de seleccionar PostGIS pulsar el botón Next.
6. En la ventana de configuración del servicio, es importante que el programa sea instalado como un servicio del sistema operativo (como se encuentra por defecto), de esta manera el servicio de PostgreSQL es iniciado al arrancar el sistema los demás datos pueden ser dejados como se encuentran por defecto, ya que no son importantes para el funcionamiento del programa. Al dar clic en Next, el programa informará la creación de un nuevo usuario en el sistema operativo, el cual es el encargado de iniciar este servicio y le asignará una contraseña aleatoria.



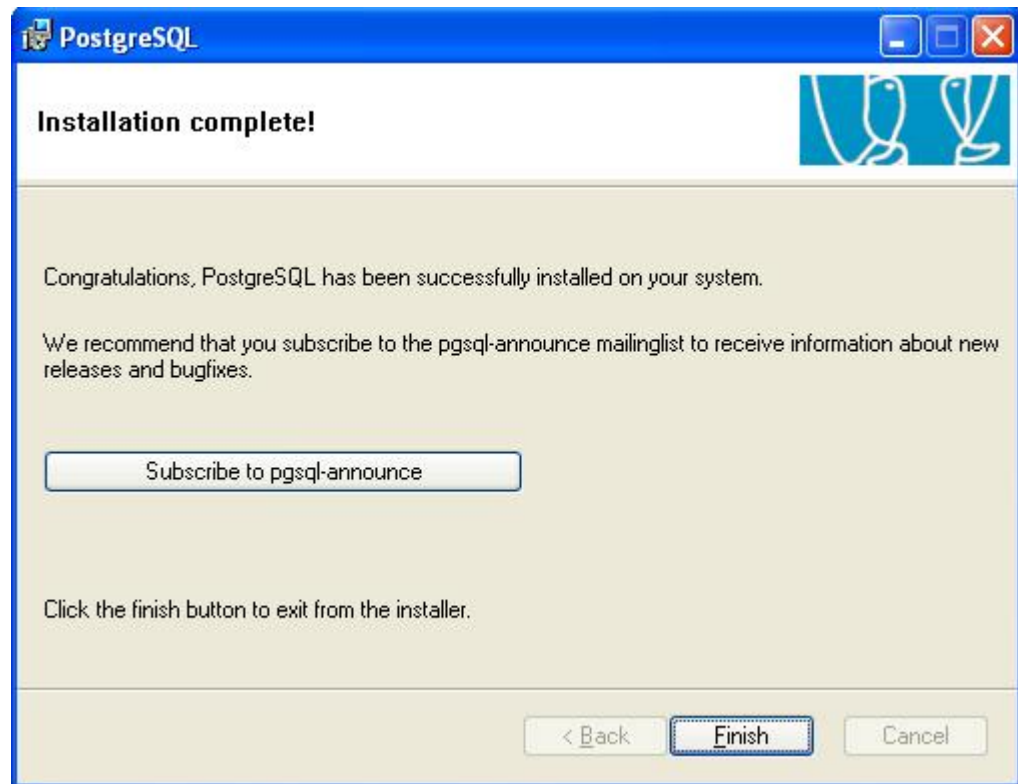
7. En la pantalla de inicialización del cluster de base de datos, es importante la información agregada. El primer campo (Port number) corresponderá al número del puerto al cual se debe conectar el programa SIGUIS, así que este valor es importante conocerlo; se recomienda dejar el valor por defecto 5432 ya que este puerto se encuentra libre de otras comunicaciones. La siguiente casilla de verificación permite al programa conectarse a distintos nombres de servidores, no solamente localhost. La lista de selección siguiente se utiliza para elegir el idioma del entorno del programa, al escoger español los mensajes de información y los menús del programa estarán en este idioma. La siguiente lista se utiliza para la codificación de los mensajes en pantalla; el tipo de codificación UTF-8 permite todas las letras del alfabeto español, como tildes y demás, por lo que mejora la apariencia del programa sobre la SQL\_ASCII que se selecciona por defecto. El campo superuser name se utiliza como el nombre de usuario para conectarse a la base de datos; este campo también es solicitado por el programa SIGUIS y por lo tanto es importante conocer su valor. El campo password es la contraseña con la cual el programa se conecta a las bases de datos, por lo tanto es importante conocerlo. Luego de llenar todos los campos se pulsa el botón Next.



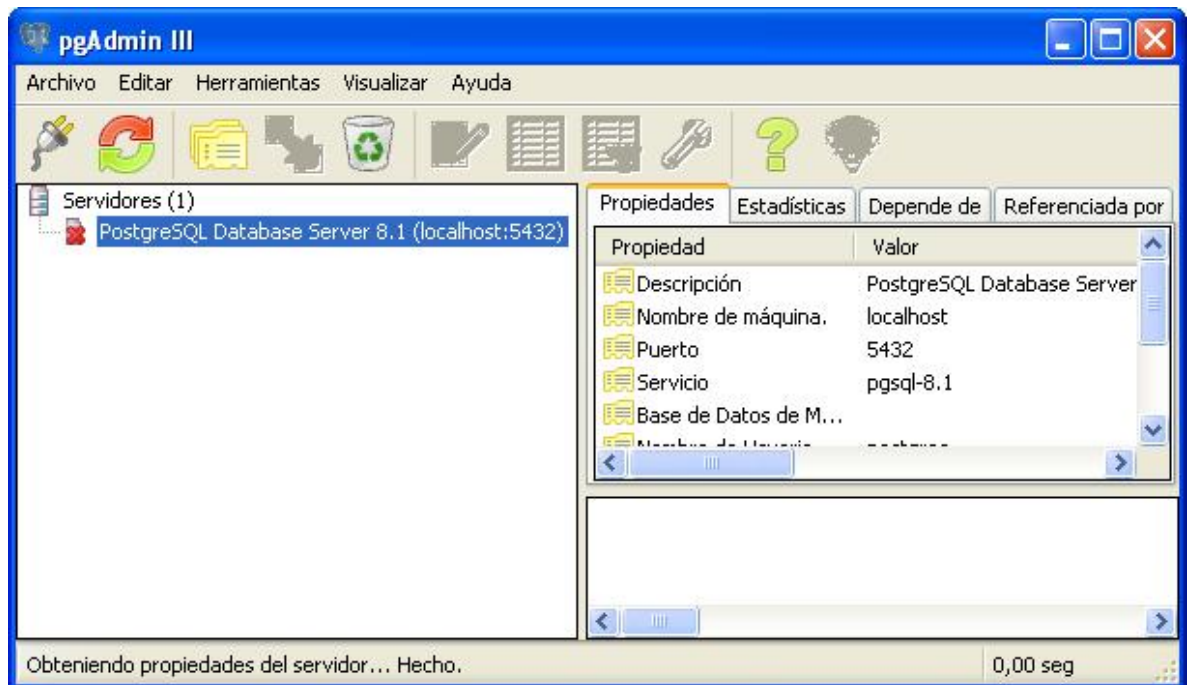
8. En las dos siguientes ventanas se pueden seleccionar lenguajes de procedimientos y algunas funcionalidades especiales para PostgreSQL. Las funciones que se escojan no afectan el funcionamiento de SIGUIS, por lo tanto pueden ser elegidas las que se deseen. Es recomendado dejar seleccionada el modulo Admin81, para que el entorno de configuración visual pgAdmin provea mejoras en la funcionalidad.
9. En la siguiente ventana se recomienda iniciar PostGIS para la base de datos inicial seleccionando la casilla de verificación, ya que esto facilita la creación de una base de datos propia, configurada de inicio para el uso de PostGIS.



10. La instalación esta lista para iniciar, se pulsa el botón Next y se espera un momento mientras el programa queda correctamente instalado. Este tiempo puede variar dependiendo de las capacidades del sistema y de los módulos que se van a instalar. Al finalizar la instalación, se muestra una nueva ventana en la que se informa la correcta instalación de PostgreSQL.



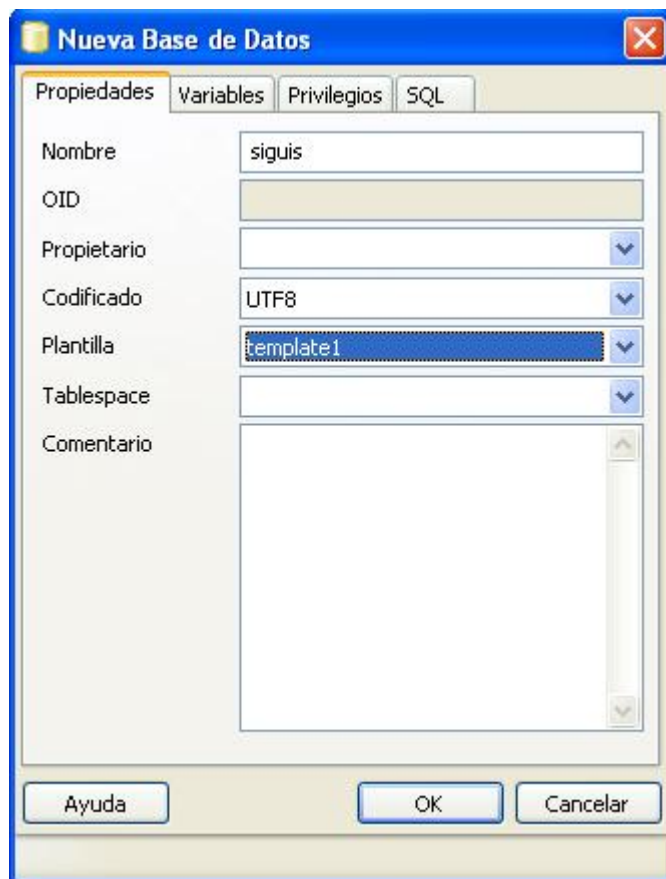
11. Luego de terminada la instalación, se procede a la creación de la base de datos y de las tablas geométricas que se utilizaran con el programa SIGUIS. Para esto, ejecutamos la herramienta visual de configuración pgAdmin III que se encuentra en el menú inicio, programas, PostgreSQL 8.1, pgAdmin III.



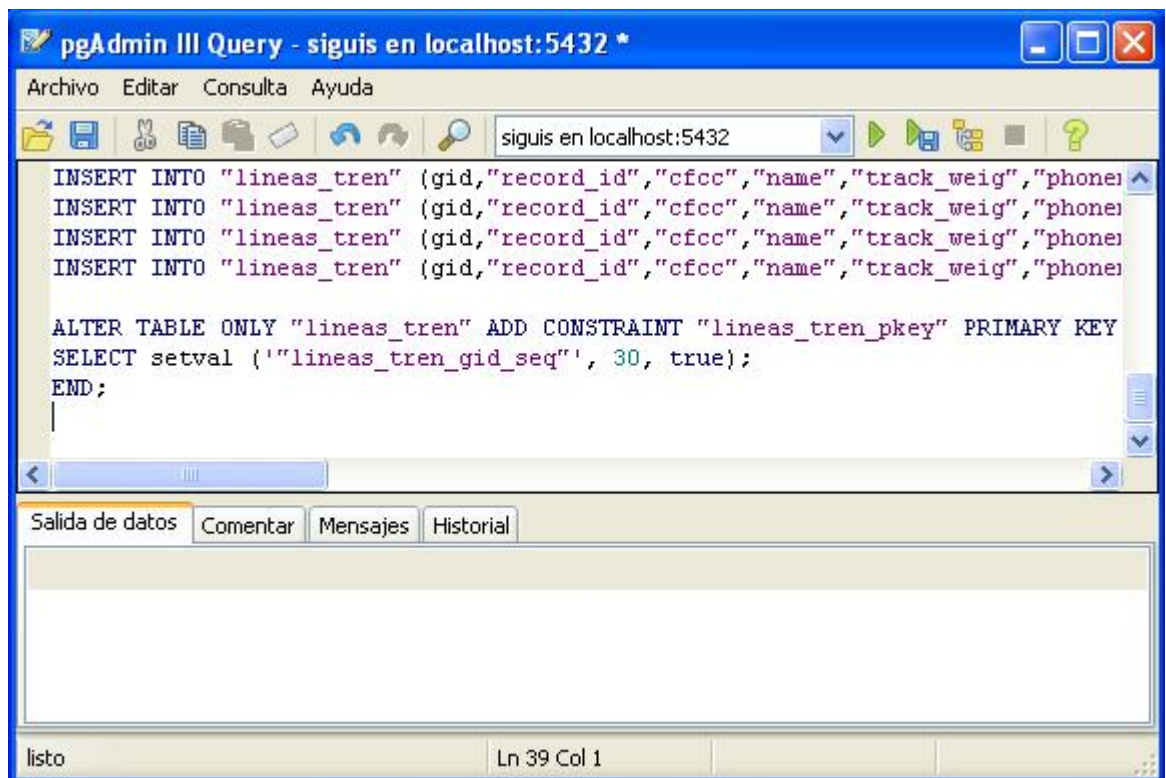
12. Primero se debe conectar al servidor localhost, que es en el que se encuentran las bases de datos. Para esto, seleccionamos el servidor en la lista de servidores, se pulsa clic derecho y se selecciona la función Conectar. En este momento se pide la contraseña de conexión, que es la que se escribió en el paso 7.



13. Cuando se realiza correctamente la conexión al servidor, se despliega el árbol correspondiente al servidor, y luego se selecciona Bases de Datos. Se pulsa clic derecho sobre ésta ubicación, y se selecciona la opción Nueva Base de Datos. Esto produce una nueva ventana en la que se debe crear la base de datos con la cual se va a utilizar el programa SIGUIS. Para esto, se escribe un nombre, se selecciona en la lista de plantillas la opción template1 y se pulsa el botón OK.



14. Luego se despliega el árbol de la nueva base de datos, luego los esquemas, y luego el esquema llamado public. Sobre este esquema se deben crear las tablas con las cuales se desea trabajar. Desde este punto se pueden crear las tablas geográficas con las cuales se desea trabajar. Para esto, luego de la instalación de SIGUIS, y habiendo instalado los archivos de ejemplo, se pulsa el botón SQL en la barra de herramientas, o se accede al menú Herramientas – Herramienta para Consultas. En esta nueva ventana se procede a copiar el contenido de cada uno de los archivos de texto que se encuentran en la carpeta de instalación de SIGUIS, y en la subcarpeta Ejemplos\PostGIS. Luego de copiar uno de los archivos, se pulsa el botón ejecutar consulta, y después de la correcta ejecución de la consulta, se borra el contenido de la ventana y se copia el siguiente archivo, hasta haber copiado todos los archivos. Esto se realiza con el propósito de tener varios mapas para verificar el funcionamiento de SIGUIS.



15. Con esto se encuentra listo para la utilización del programa SIGUIS, se debe tener en cuenta para el funcionamiento, el nombre de la base de datos que se creó, el nombre del servidor al que se encuentra conectado (generalmente es localhost), el puerto al cual se encuentra conectado, el nombre de usuario con el cual realiza la conexión, la clave de usuario, y el nombre de las tablas que se acaban de crear.

Para finalizar la ayuda sobre PostgreSQL, se explica a continuación el proceso que se debe realizar para convertir un archivo en formato shapefile al formato PostGIS.

1. Como requisito, se debe tener instalado el programa PostgreSQL, o poseer el archivo shp2pgsql.exe que se almacena en la carpeta de instalación.
2. Se debe abrir una ventana de comandos en modo DOS, para esto, se pulsa el botón ejecutar en el menú inicio, se escribe cmd y se pulsa la tecla Enter.
3. Luego se debe acceder al directorio que posee los archivos de instalación de PostgreSQL. Para realizar esta operación, se escribe el comando cd "C:\Archivos de programa\PostgreSQL\8.1\bin" y se pulsa la tecla Enter. La dirección del programa puede cambiar de acuerdo a la configuración del sistema.

4. Desde el explorador de Windows, se copian los tres archivos de cada capa en formato shapefile que se desea convertir a la carpeta a la cual se acaba de acceder.
5. Se ejecuta la siguiente instrucción: `shp2pgsql.exe nombre_shape.shp nombre_tabla > salida.txt`  
En donde `nombre_shape` corresponde al nombre del archivo que se encuentra en formato shapefile y que se desea convertir; `nombre_tabla` corresponde al nombre de la tabla de PostgreSQL que se va a crear; y `salida.txt` es el nombre del archivo en el cual se van a copiar las instrucciones SQL necesarias para la creación de la tabla.
6. Se procede a copiar los archivos de salida creados en el ejecutor de consultas SQL del programa PostgreSQL (véase el paso 14 del proceso de instalación de PostgreSQL).

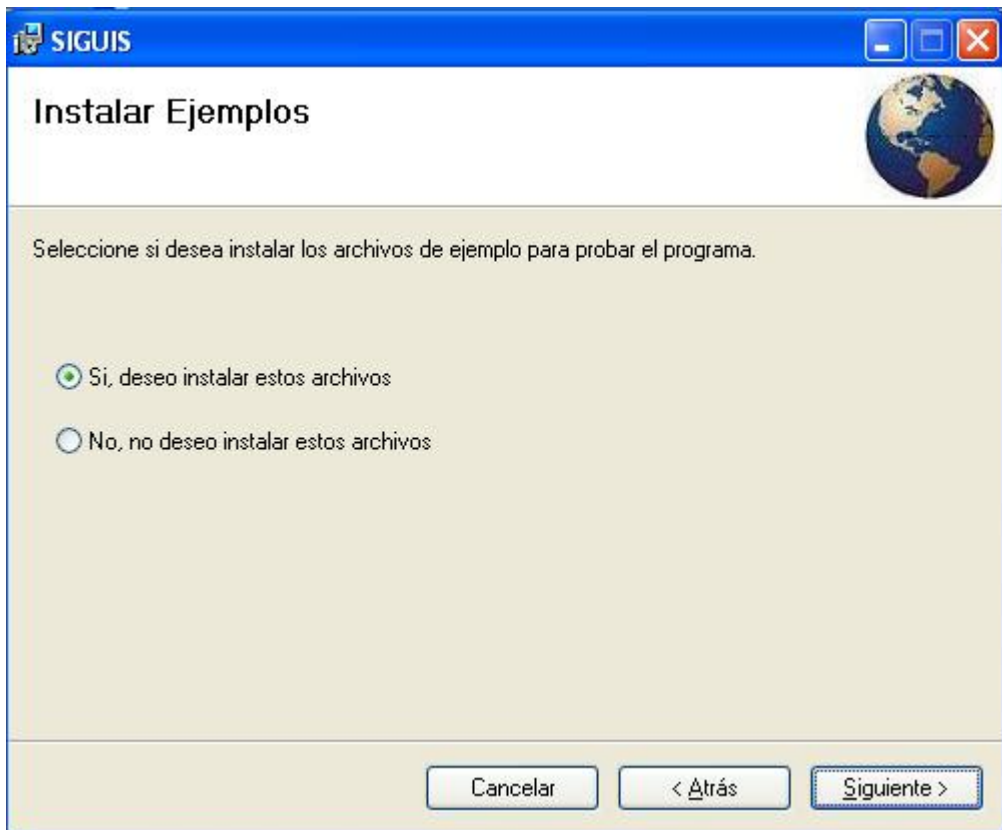
### Instalación de SIGUIS

El proceso para la instalación de SIGUIS es el siguiente:

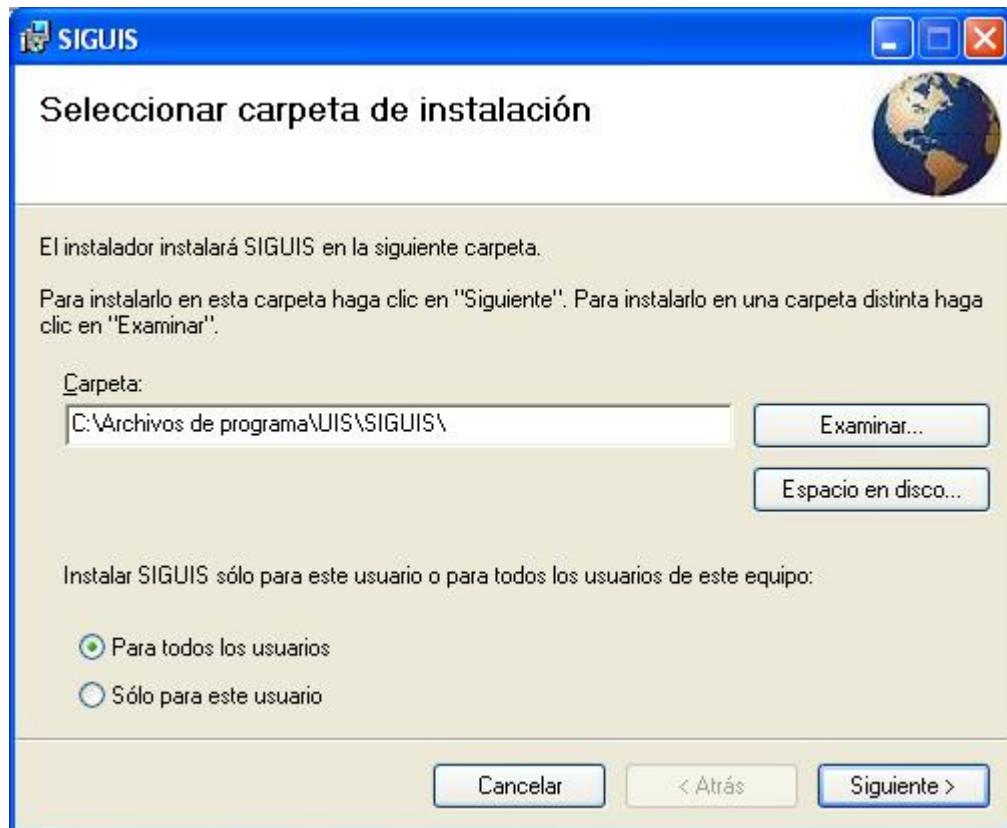
1. Ejecutar el archivo Setup.Exe que se encuentra en el CD del programa.



2. Pulsar Siguiente en la pantalla de presentación del programa y en la página de inicio del asistente de instalación. Luego leer la información sobre el programa SIGUIS y pulsar el botón Siguiente.
3. Seleccionar si desea o no instalar los mapas de ejemplo de SIGUIS. Se recomienda seleccionar la instalación de éstos, pues incluyen mapas tanto en formato shapefile como los archivos para generar las tablas en PostGIS que contienen mapas en este formato. Luego de decidir esta opción, se pulsa el botón Siguiente.



4. Se procede a seleccionar la carpeta en la cual desea instalar el programa SIGUIS, para cambiar la carpeta que se escribe por defecto, se debe pulsar el botón Examinar y seleccionar una diferente. El botón Espacio en disco permite verificar si se posee el espacio suficiente para completar la instalación en la unidad seleccionada. La opción de los usuarios que podrán utilizar el programa SIGUIS se utiliza para realizar instalaciones administrativas; se recomienda permitir que todos los usuarios del sistema puedan utilizar este programa. Luego de completar la información requerida en esta página, se pulsa el botón Siguiete.



5. El programa está listo para su instalación, y luego de pulsar el botón Siguiete, se inicia la instalación del programa y después de un tiempo en el que se copian los archivos nuevos y se crean las entradas en el registro, se muestra la información de finalización del proceso de instalación, y se encuentra listo para la utilización del programa SIGUIS.
6. Para abrir el programa SIGUIS, se puede acceder al icono de acceso directo creado en el escritorio, o al acceso directo creado en el menú inicio.

## ANEXO F. Manual de usuario de SIGUIS

### **Visualización:**

#### **Abrir capas**

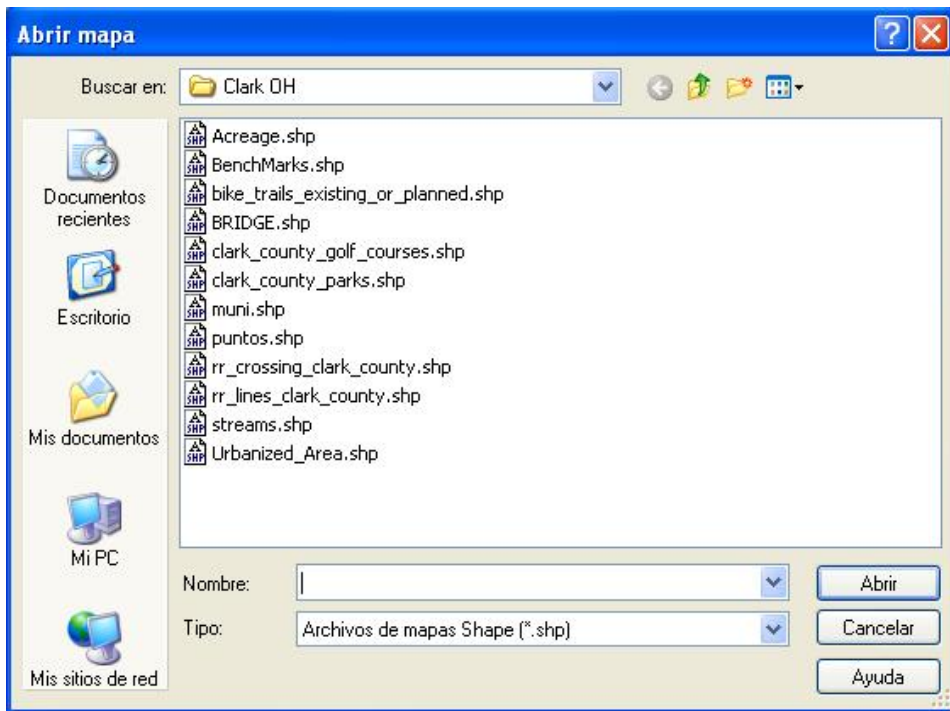
La primera función del visualizador, es la de agregar capas al formulario, para esto se puede pulsar el icono de agregar capa, o acceder a esta función mediante el menú Capas - Agregar, o mediante la combinación de teclas ctrl+O.



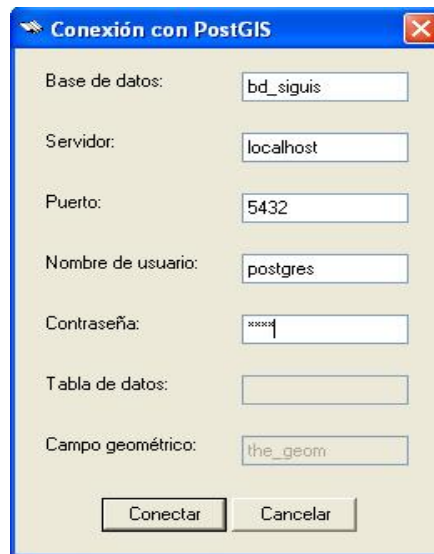
Luego se debe definir que tipo de capa se desea abrir; el programa SIGUIS soporta capas tipo shape y capas de PostGIS. Para abrir una capa de PostGIS se debe acceder al servidor que las posee, por lo tanto es necesario que el programa PostgreSQL con su extensión PostGIS este instalado en el equipo cliente.



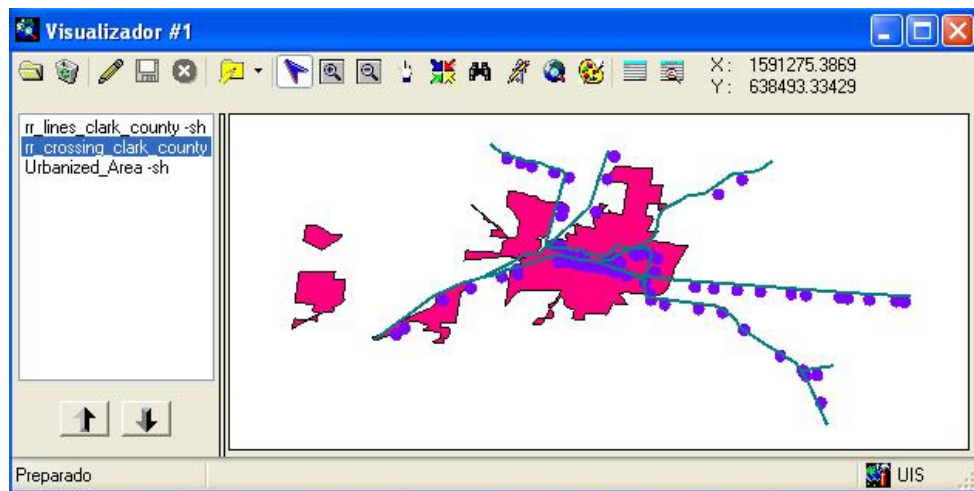
Si se elige el tipo de capa Shape, luego se debe seleccionar el archivo .shp correspondiente a la capa que se desea abrir, y posteriormente escoger el color que se desea aplicar a la capa.



Si se selecciona un tipo de capa PostGIS, se debe llenar los datos solicitados para poder acceder al servidor, y luego escoger el color que se desea aplicar a la capa.



Al finalizar todo este proceso, se obtendrá un formulario similar al siguiente, en donde se puede apreciar las capas agregadas, en el visor de capas en orden descendente es decir la primera capa en la lista es la capa superior en el visualizador.



## Cerrar capas

Se pueden cerrar las capas abiertas, para permitir una mejor visualización de otras capas, para acelerar el proceso de dibujo de las capas que interesan, o porque ya no se va a necesitar. Para esto se puede pulsar el icono de cerrar capa, o acceder a esta función mediante el menú Capas - Cerrar, o mediante la combinación de teclas ctrl+Q.



## Cambiar propiedades a la capa

Se pueden cambiar las propiedades de visualización de las capas, tales como el color, el grosor de la línea y la visibilidad de la capa, para esto se debe hacer doble click en la lista de capas, sobre la capa que se desea modificar. Con esto, se mostrará el formulario de las propiedades de la capa, en el cual se pueden apreciar y modificar las principales características de visualización de la capa.



La forma de realizar los cambios es la siguiente:

1. Para cambiar el color de una capa se pulsa el cuadro relleno con el color actual. Con esta operación, aparece un formulario en el que se selecciona el nuevo color, o se elige un color personalizado. En caso de que la casilla de Ningún color este marcada, primero se debe desmarcar antes de poner seleccionar un nuevo color.
2. Si no se desea que la capa tenga color, se selecciona la casilla que dice Ninguno. Esto no hace que la capa sea invisible, esto hace que el relleno no se vea; para las capas de puntos, líneas y multipuntos seleccionar esta opción hace que la capa no se vea; para las capas de polígonos hace que solo se vea el contorno.

3. Para cambiar el grosor de la línea de dibujo se selecciona el valor deseado en la lista denominada Grosor. A las capas de punto y multipunto no se les puede realizar este cambio, ya que no poseen líneas.
4. Si se desea que la capa sea invisible, se pulsa el botón de radio denominado No, al frente del marcador visible.
5. Si se desea que la capa sea visible, se pulsa el botón de radio denominado Si, al frente del marcador visible.
6. Para aplicar los cambios se pulsa el botón Aceptar, o para omitirlos se pulsa el botón Cancelar o se cierra el formulario.

### Operaciones de vista

Se pueden modificar algunas propiedades del visualizador como el zoom, centrar en un punto, ver todas las capas, ajustar la vista a la capa actual, seleccionar una única zona del mapa para hacer visible, medir distancias, agregar etiquetas al mapa, y realizar consultas SQL sobre una capa abierta para la creación de una nueva capa. Para realizar cualquiera de estas operaciones, se debe elegir la opción correspondiente en el menú del programa o en la barra de herramientas.



**Zoom in, Zoom out:** Para realizar zoom in (acercar) o zoom out (alejar) se debe elegir esta opción en el menú Vista - Zoom in (out) o presionar el botón correspondiente en la barra de herramientas. Para verificar que la opción fue seleccionada correctamente, el botón de la barra de herramientas debe aparecer como si estuviera iluminado, el mouse debe cambiar de forma a una lupa con un signo de + o - según sea el caso, y la barra de estado del programa debe informar

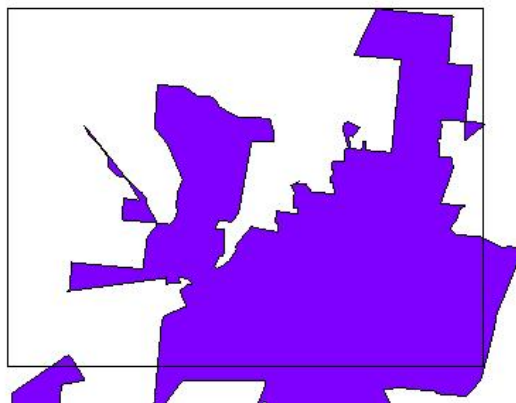
la operación que se está llevando a cabo. Para realizar el acercamiento o alejamiento, después de haber seleccionado la opción, se debe pulsar click en el formulario, en el lugar que se desea que sea el centro de la nueva imagen. Esta operación también puede ser llevada a cabo, en caso de que el mouse del computador utilizado posea una rueda central, rotando esta rueda hacia arriba (acercar) o abajo (alejar), teniendo el cursor ubicado en el área de dibujo.

**Centrar:** La operación de centrar ubica el punto seleccionado en el centro del área de dibujo. Para realizar esta operación, se debe pulsar el botón correspondiente en la barra de herramientas o en el menú Vista - Centrar; luego se debe dar click en el área de dibujo, en el lugar que se desea centrar la imagen.

**Ver todas las capas:** Esta operación se utiliza para ampliar el rango de la imagen (los valores de X mínimo y máximo, y Y mínimo y máximo) de tal forma que todas las capas abiertas puedan ser vistas en su totalidad en el área de dibujo.

**Ver capa actual:** Esta operación se utiliza para ajustar el rango de la imagen (los valores de X mínimo y máximo, y Y mínimo y máximo) de tal forma que la capa seleccionada en la lista de capas pueda ser vista en su totalidad en el área de dibujo.

**Ver zona:** Esta operación se utiliza para ajustar el rango de la imagen (los valores de X mínimo y máximo, y Y mínimo y máximo) de tal forma que lo único que se pueda ver en el área de dibujo sean los elementos contenidos en el rectángulo dibujado luego de seleccionar esta opción. Para llevar a cabo esta operación, se debe pulsar el botón correspondiente, o mediante el menú Vista - Ver zona, y luego dibujar un rectángulo en el área de dibujo, pulsando el botón derecho del mouse y sin soltarlo moverlo de tal forma que el área deseada esté contenida en el rectángulo; para terminar, se suelta el botón derecho del mouse, y el programa se encarga de dibujar solo la zona seleccionada.



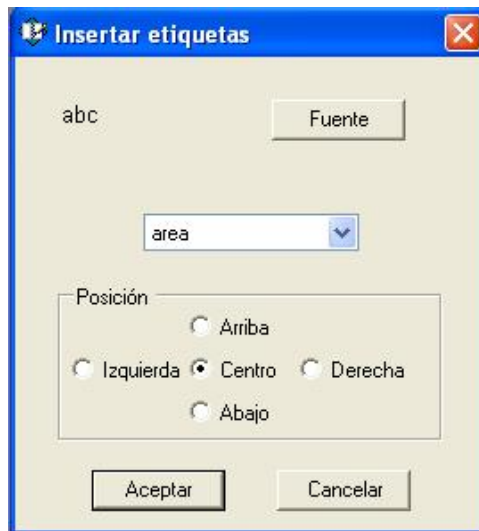
**Medir distancia:** Esta operación mide la distancia total entre varios puntos seleccionados. Para realizar esta operación, se debe pulsar el botón correspondiente en la barra de herramientas o en el menú Vista - Medir distancia, y luego ubicar los puntos por los que se desea pasar dando click en el área de dibujo. Para mostrar la distancia total se pulsa doble click en el área de dibujo en el momento en el que se ubique el último punto, y se muestra un cuadro de diálogo con el valor calculado.



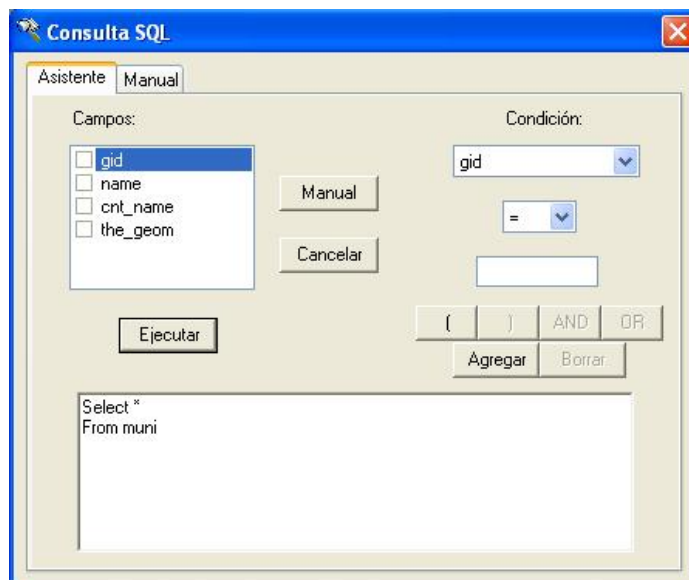
**Etiquetas:** Esta operación agrega o quita texto del mapa, que da una explicación sobre cada objeto geográfico (Ej.: Nombre, descripción, etc.). Para agregar las etiquetas se debe acceder al menú Vista - Etiquetas - Agregar etiquetas, y con esto se invoca un formulario en el que se escoge el formato del texto, el campo de la base de datos que será escrito en cada elemento, y la posición de la etiqueta con respecto al objeto correspondiente. La posición puede ser:

- Arriba: La etiqueta se ubica encima de cada elemento, centrada en cuanto a su espacio horizontal.
- Derecha: La etiqueta se ubica a la derecha de cada elemento, centrada en cuanto a su espacio vertical.
- Abajo: La etiqueta se ubica debajo de cada elemento, centrada en cuanto a su espacio horizontal.
- Izquierda: La etiqueta se ubica a la izquierda de cada elemento, centrada en cuanto a su espacio vertical.
- Centro: La etiqueta se ubica en el centro del rectángulo que encierra cada elemento.

Para quitar las etiquetas a una capa que las tenga, se debe acceder al menú Vista - Etiquetas - Quitar etiquetas.



**Consulta SQL:** Esta opción permite realizar una consulta sobre los datos alfanuméricos de una capa de tipo PostGIS, para generar una nueva capa cuyos elementos sean solo los registros que cumplan con las condiciones impuestas en la consulta. La consulta SQL se puede realizar utilizando el asistente que aparece por defecto, en el cual se eligen los campos que harán parte de la nueva capa, y se agregan condiciones a la consulta; o mediante la escritura manual del código SQL. La opción manual solo debe ser elegida por personas con conocimiento de SQL y de la estructura de la capa seleccionada; con esta opción se pueden incluir funciones propias de PostgreSQL y no solo del lenguaje de consulta SQL.



- *Utilización del asistente de consulta SQL:* Los campos que se desea que aparezcan en la nueva capa, deben ser seleccionados en la lista de campo, de tal forma que el cuadro a la izquierda de cada elemento se encuentre relleno. En caso de no elegir ningún campo en particular, a la nueva capa se le agregarán todos los campos de la capa seleccionada. Las condiciones se deben agregar una por una, primero seleccionando el campo al cual se le desea aplicar una restricción, luego la operación booleana que debe cumplir (igual que, mayor que, menor que, etc.) y el valor que lo restringe, seguido por la pulsación del botón Agregar. En caso de que el campo al cual se le desea agregar la restricción sea de tipo texto se debe escribir el valor entre comillas sencillas, así: 'valor'.

Las consultas pueden poseer múltiples condiciones, y para esto se deben utilizar los botones de AND, OR y los paréntesis para agrupar consultas. Al finalizar la consulta, se debe pulsar el botón Ejecutar y, en caso de que algún elemento cumpla con las condiciones planteadas, se pedirá el nombre de la nueva tabla de PostGIS que se creará, y en caso contrario se mostrará un mensaje de advertencia.

## Mostrar tabla

Se pueden obtener los datos asociados a la capa, seleccionando mostrar tabla, ya sea pulsando el botón adecuado en la barra de herramientas, o mediante el menú Datos - Ver tabla, o pulsando la tecla F11.



Con esto, se hará el llamado al formulario que contiene los datos asociados a la capa correspondiente.

SIGUIS - [Datos asociados a la capa Urbanized\_Area en Visualizador #1]

Proyecto Tabla Ventana Ayuda

Urbanized_Area								
	area	perimeter	ua99_d00_	ua99_d00_i	ua	name	lsad	lsad_trans
▶	0.01172	0.9971	4897	4896	83980	Springfield, O	75	Urbanized Ar
	0.00003	0.0234	4940	4939	83980	Springfield, O	75	Urbanized Ar
	0.00002	0.04887	4943	4942	83980	Springfield, O	75	Urbanized Ar
	0.00009	0.05619	4956	4955	83980	Springfield, O	75	Urbanized Ar
	0.00041	0.08709	4958	4957	62083	New Carlisle,	76	Urban Cluster
	0.00123	0.15517	4995	4994	22528	Dayton, OH	75	Urbanized Ar
	0.00122	0.2888	5007	5006	22528	Dayton, OH	75	Urbanized Ar
	0.00004	0.02625	5024	5023	83980	Springfield, O	75	Urbanized Ar
	0.00008	0.05941	5035	5034	22528	Dayton, OH	75	Urbanized Ar
	0.00004	0.0285	5036	5035	83980	Springfield, O	75	Urbanized Ar

En este formulario se pueden efectuar las operaciones básicas de una base de datos tales como agregar campo, eliminar campo y modificar los datos.

Si se desea agregar un campo, se accede al menú Tabla - Agregar campo, o mediante la combinación de teclas ctrl+A. Con esto, se invoca un formulario en el cuál se pide al usuario el nombre del campo y el tipo de datos que poseerá.

Agregar Campo

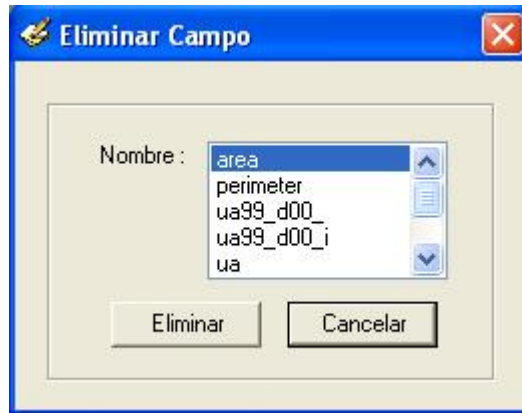
Nombre :

Tipo:

Aceptar Cancelar

Si lo que desea es eliminar un campo, se accede al menú Tabla - Eliminar campo, o mediante la combinación de teclas ctrl+E. Con esto, se hace el llamado a un formulario en el cual se muestra una lista con los campos existentes en el momento; el usuario debe seleccionar el campo que desea eliminar y pulsar el botón aceptar, cabe anotar que debe haber por lo menos un campo, o de lo

contrario NO se mostrará el formulario. Una vez realizada esta operación, no se pueden volver a obtener los datos perdidos, por lo tanto solo debe ser realizada si se esta seguro de lo que se desea hacer.



Para editar los datos de la tabla se accede al menú Tabla - Modificar, o mediante la combinación de teclas ctrl+M. Una vez realizado esto, se puede modificar toda la información que se desee, teniendo en cuenta los tipos de datos, es decir, en un campo numérico no esta permitido escribir texto o en un campo de tipo fecha se debe escribir una fecha válida. Cuando se realizan los cambios deseados, se puede guardar la nueva tabla pulsando el botón Guardar, o cancelar los cambios realizados pulsando el botón Cancelar, o accediendo al menú Tabla - Cancelar, o mediante la combinación de teclas ctrl+N.

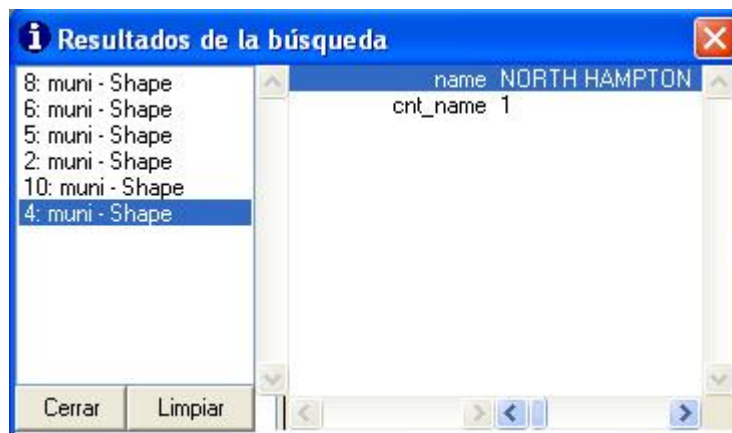
	area	perimeter	ua99_d00_	ua99_d00_i	ua	name	lsad	lsad_trans
▶	0.01172	0.9971	4897	4896	2002	Springfield, O	75	Urbanized Ar
	0.00003	0.2528	4940	4939	83980	Springfield, O	75	Urbanized Ar
	0.00002	0.04887	4943	4942	83980	Shelbyville, O	80	Urbanized Ar
	0.00009	0.05619	4956	4955	83980	Springfield, O	75	Urbanized Ar
	0.00041	0.08709	4958	4957	62083	New Carlisle,	76	Urban Cluster
	0.00123	0.15517	4995	4994	22528	Dayton, OH	75	Urbanized Ar
	0.00122	0.2888	5007	5006	22528	Dayton, OH	75	Urbanized Ar
	0.00004	0.02625	5024	5023	83980	Springfield, O	75	Urbanized Ar
	0.00008	0.05941	5035	5034	22528	Dayton, OH	75	Urbanized Ar
	0.00004	0.0285	5036	5035	83980	Springfield, O	75	Urbanized Ar
*								

## Buscar información

Se pueden obtener los datos alfanuméricos asociados a los elementos geográficos, seleccionando Buscar información en la barra de herramientas, o mediante el menú Datos - Buscar Información, o pulsando la tecla F12.



Luego se deben seleccionar los objetos geográficos que se desean consultar, y aparecerá un formulario como el siguiente, en el cual se muestran los datos asociados a los elementos seleccionados.



## Operaciones de ventana

Se pueden tener varios visualizadores abiertos a la vez, por esto, se puede personalizar las características de visualización tales como ventanas en cascada, mosaico horizontal, mosaico vertical o cerrar todas las ventanas. Esto con el propósito de organizar todas las ventanas abiertas, para una mejor apreciación de los mapas.

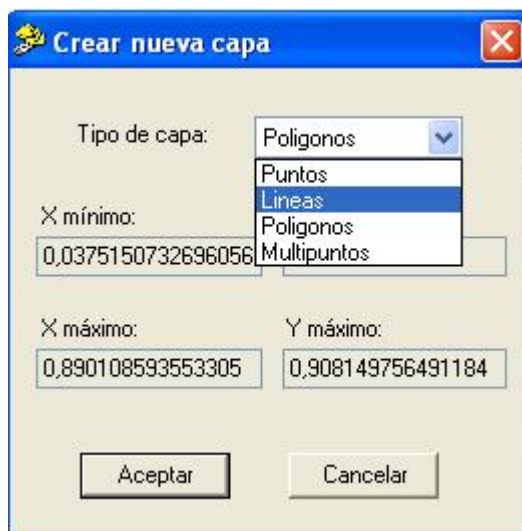


## Crear capas

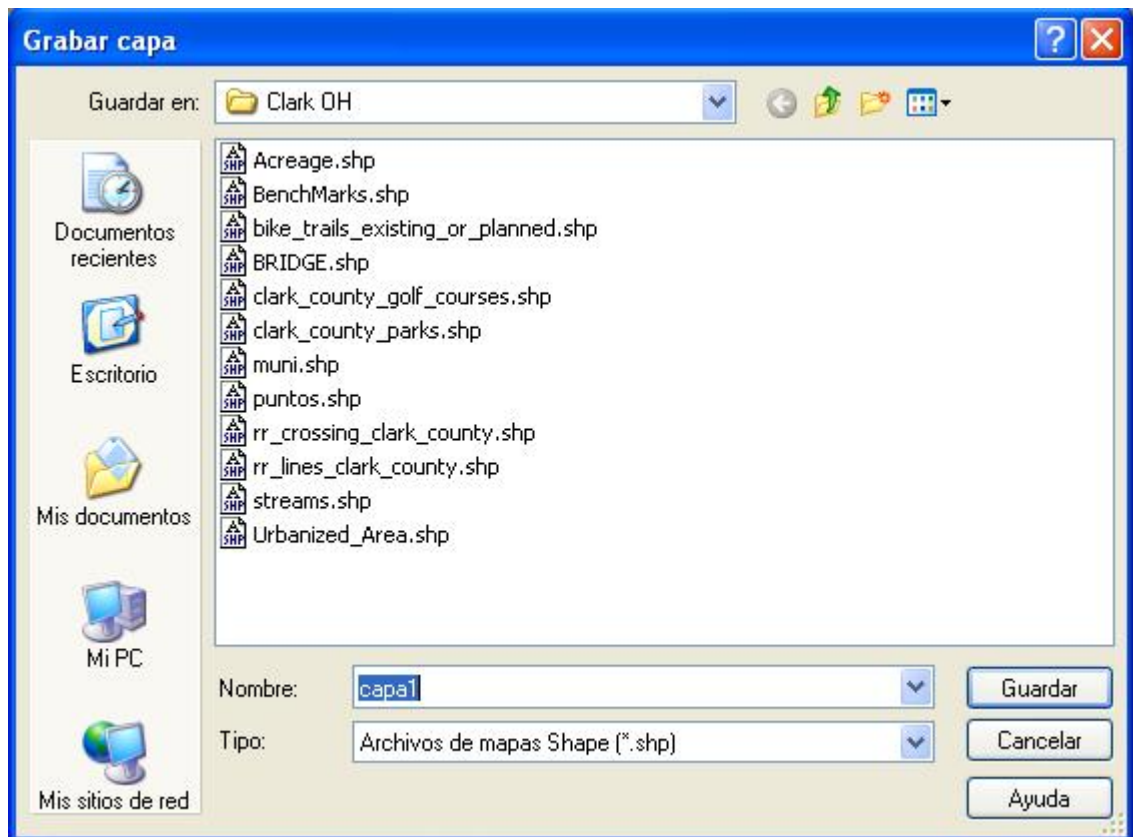
Se pueden crear nuevas capas escogiendo esta opción en el menú del programa (Capas - Crear), o a través de la barra de herramientas pulsando el botón Crear nueva capa, o mediante la combinación de teclas ctrl+N.



Con esto, aparecerá un nuevo formulario, en el que se pedirá el tipo de la capa que se desea crear y el eje de coordenadas, los cuales pueden ser modificados si no hay capas abiertas previamente.



Las capas creadas pueden ser guardadas en formato Shapefile o en PostGIS. Para el formato Shapefile se pedirá el nombre del archivo con el cual desea guardar la nueva capa, y para PostGIS se pedirán los datos para conectarse al servidor y el nombre de la nueva tabla que se va a crear; en caso de que la tabla ya exista, se pedirá confirmación, para proceder a eliminar la tabla y reemplazarla con la nueva.



Guardar en formato Shapefile

Conexión con PostGIS

Base de datos: siguis

Servidor: localhost

Puerto: 5432

Nombre de usuario: postgres

Contraseña: [masked]

Tabla de datos: mi\_tabla

Campo geométrico: the\_geom

Aceptar Cancelar

Guardar en PostGIS

Para guardar una capa que se encuentra en creación, se accede al menú Capas - Guardar, o pulsando el botón Guardar la capa creada en la barra de herramientas, o mediante la combinación de teclas ctrl+G. Para cancelar la creación de una capa, se accede al menú Capas - Cancelar, o pulsando el botón Cancelar la creación de la capa en la barra de herramientas, o mediante la combinación de teclas ctrl+M.

### Creación de puntos

Se pueden crear capas de puntos, escogiendo Puntos en la lista de tipo de capa del formulario para crear una capa nueva:

Crear nueva capa

Tipo de capa: Poligonos

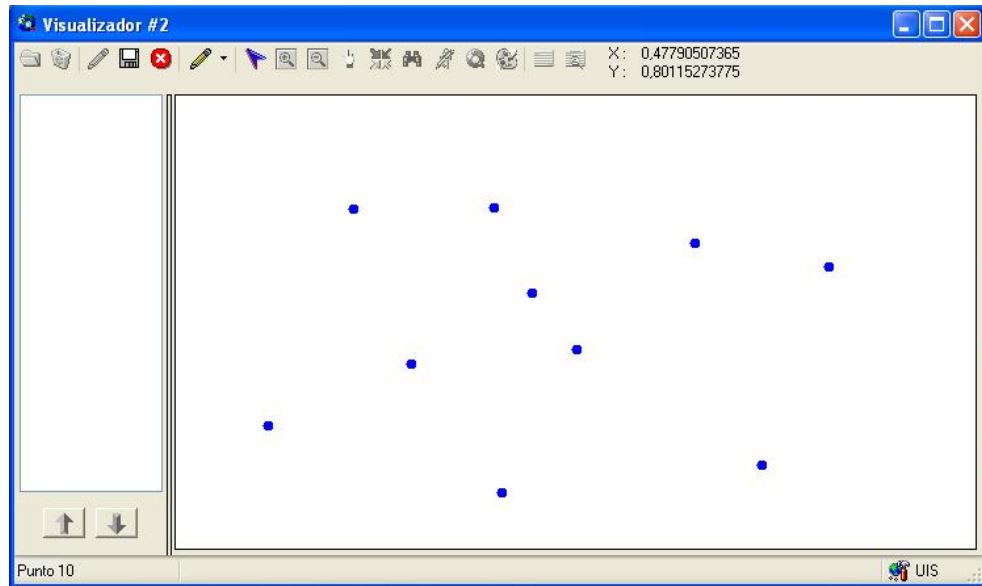
X mínimo: 0,0375150732696056

X máximo: 0,890108593553305

Y máximo: 0,908149756491184

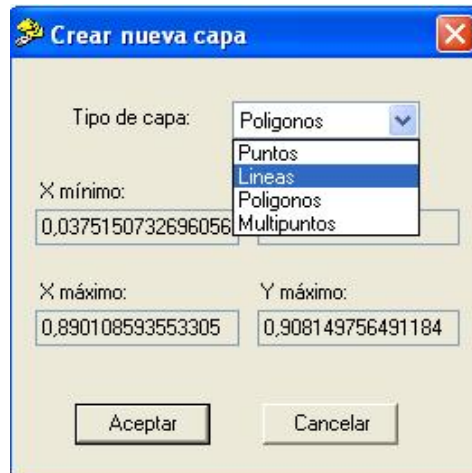
Aceptar Cancelar

Para ubicar cada punto de la capa se pulsa click sobre el formulario. Cuando se hayan terminado de crear los puntos, se procede a guardar la capa o a cancelar la creación, en caso de no estar conforme con el resultado.

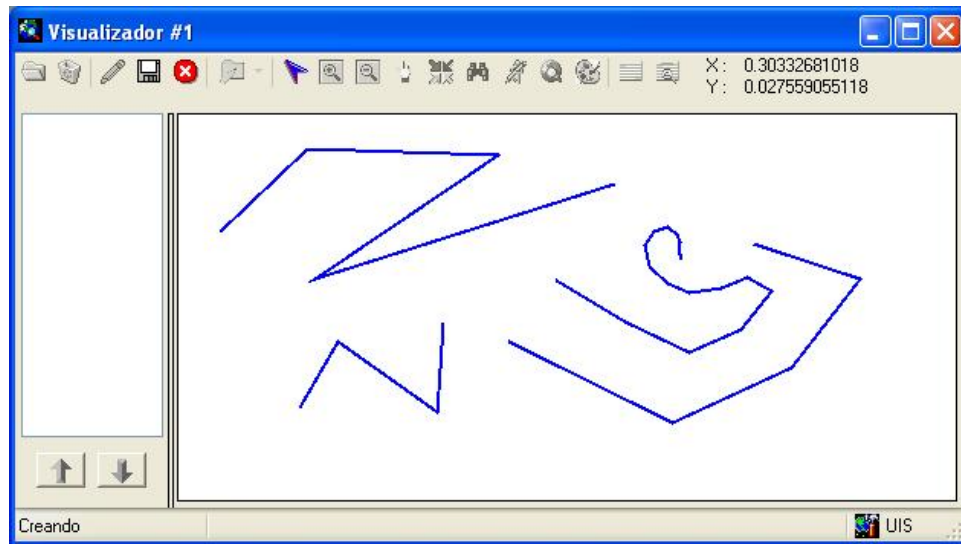


## Creación de líneas

Se pueden crear capas de líneas, escogiendo Líneas en la lista de tipo de capa del formulario para crear una capa nueva:



Para dibujar las líneas, se presiona click sobre el formulario en el punto inicial; los vértices de esta línea se ubican presionando nuevamente click, y para terminar la creación de una línea se debe pulsar doble click. Este proceso se repite con todas las líneas que se deseen crear y que pertenezcan a una misma capa. Cuando se hayan terminado de crear las líneas, se procede a guardar la capa o a cancelar la creación, en caso de no estar conforme con el resultado.

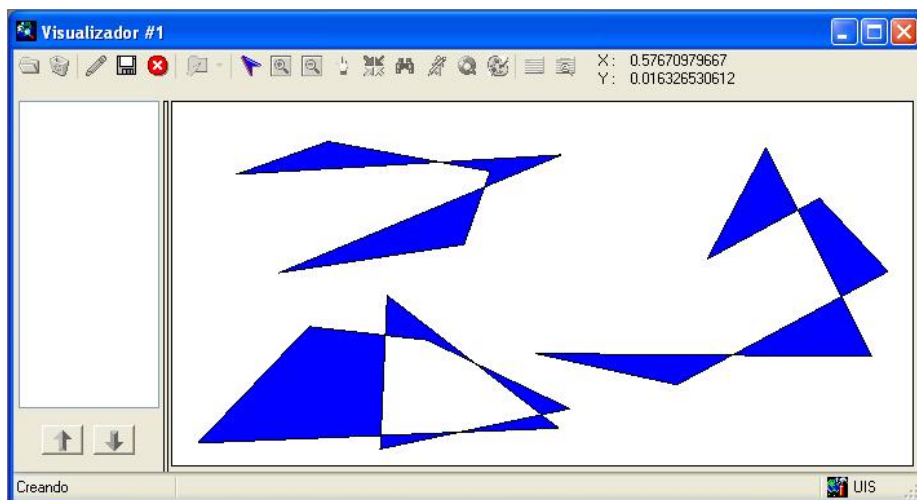


## Creación de polígonos

Se pueden crear capas de polígonos, escogiendo Polígonos en la lista de tipo de capa del formulario para crear una capa nueva:

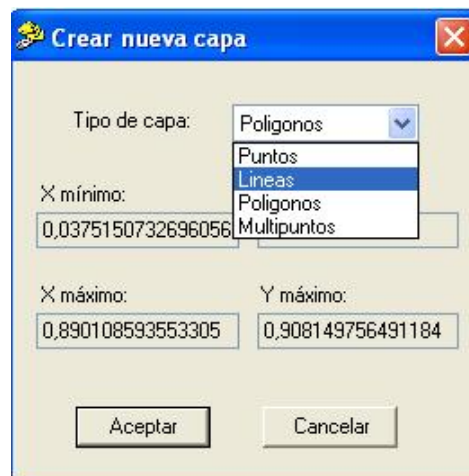


Para dibujar los polígonos, se presiona click sobre el formulario en el punto inicial; los vértices del polígono se ubican presionando nuevamente click, y para terminar la creación de un polígono se pulsa doble click. Este proceso se repite con todos los polígonos que se deseen crear y que pertenezcan a una misma capa. Un mismo polígono no puede intersectarse consigo mismo, por lo tanto cuando esto suceda, la parte que se intersecte quedará en blanco y producirá un hueco en el elemento. Cuando se hayan terminado de crear los polígonos, se procede a guardar la capa o a cancelar la creación, en caso de no estar conforme con el resultado.

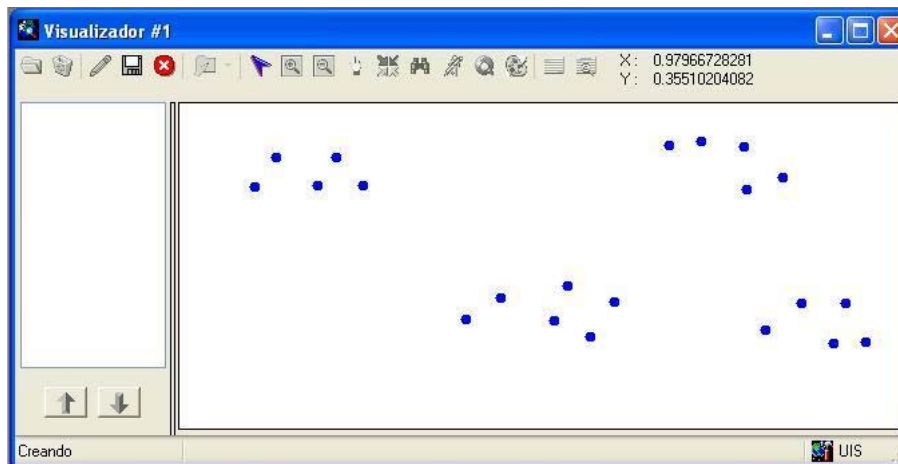


## Creación de multipuntos

Se pueden crear capas de multipuntos, escogiendo Multipuntos en la lista de tipo de capa del formulario para crear una capa nueva:



Para ubicar cada punto perteneciente a un multipunto se pulsa click sobre el formulario; para terminar la creación del multipunto se pulsa doble click. Cuando se hayan terminado de crear los multipuntos, se procede a guardar la capa o a cancelar la creación, en caso de no estar conforme con el resultado.



## Editar capas

Se puede editar la capa seleccionada en la lista de capas abiertas, escogiendo la opción de editar en el menú del programa (Capas - Editar - Editar), o pulsando el botón Empezar a editar de la barra de herramientas, o mediante la combinación de teclas ctrl+T.



Luego se puede elegir que tipo de cambio que se desea realizar en la capa. Estos cambios son Desplazar elementos, eliminar elementos y agregar nuevos elementos.

**Desplazar elementos:** Esta opción permite desplazar objetos geográficos ya existentes, de forma que la ubicación cambia, pero los datos asociados a ella no. Para acceder a esta operación, se debe escoger en el menú Capas - Editar - Desplazar, o seleccionando la opción Desplazar en el submenú del botón

Empezar a editar, o mediante la combinación de teclas ctrl+D. Para realizar esta operación, se pulsa el botón izquierdo del mouse sobre el elemento que se desea desplazar y sin soltar este botón se arrastra hasta su nueva ubicación; una vez allí se suelta el botón.

**Eliminar elementos:** Esta opción permite eliminar objetos geográficos ya existentes, de tal forma que no solo se eliminar el objeto del mapa, sino también los datos asociados a este. Para acceder a esta operación, primero se debe seleccionar el elemento o los elementos a eliminar; para esto, se debe escoger en el menú Capas - Editar - Seleccionar, o seleccionando la opción Eliminar - Seleccionar en el submenú del botón Empezar a editar, o mediante la combinación de teclas ctrl+X, y luego dar clic sobre el elemento que se va a eliminar. Una vez que se tiene seleccionado el objeto a eliminar, se accede al menú Capas - Editar - Eliminar, o se selecciona la opción Eliminar - Eliminar en el submenú del botón Empezar a editar, o pulsando la tecla Supr.

**Agregar elementos:** Esta opción permite agregar nuevos objetos geográficos a la capa seleccionada. Para acceder a esta operación, se debe escoger en el menú Capas - Editar - Agregar, o seleccionando la opción Agregar en el submenú del botón Empezar a editar, o mediante la combinación de teclas ctrl+A. Una vez seleccionada esta operación, se procede a dibujar los elementos, del mismo modo que en la creación de una capa (Ver).

**Deshacer:** Esta operación permite deshacer el último cambio realizado a la capa que se encuentra en edición, es decir, volver al estado anterior de la capa. Solo se puede realizar cinco (5) veces consecutivas esta operación. Para realizar esta operación, se debe escoger en el menú Capas - Editar - Deshacer, o seleccionando la opción Deshacer en el submenú del botón Empezar a editar, o mediante la combinación de teclas ctrl+Z.

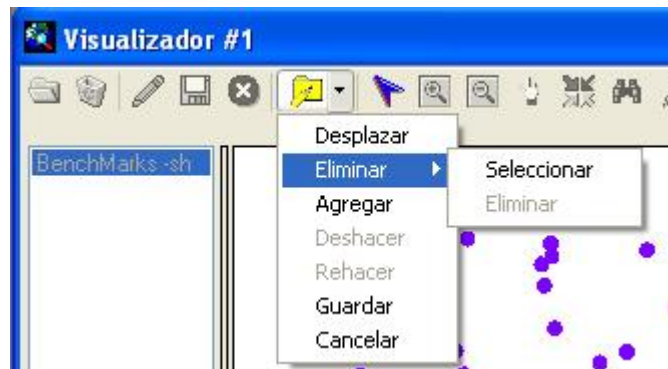
**Rehacer:** Esta operación permite rehacer el último cambio realizado a la capa que se encuentra en edición y que fue deshecho mediante la operación de deshacer, es decir, avanzar al siguiente estado de la capa, en caso de existir alguno. Esta operación solo puede ser realizada después de una o varias operaciones de deshacer. Para realizar esta operación, se debe escoger en el menú Capas - Editar - Rehacer, o seleccionando la opción Rehacer en el submenú del botón Empezar a editar, o mediante la combinación de teclas ctrl+Y.

**Guardar cambios:** Con esto se puede guardar la capa editada, como un archivo diferente, o reemplazando el archivo existente. Se pueden editar las capas en cualquiera de los dos formatos aceptados, pero solo se pueden guardar en formato Shape. Para realizar esta operación, se debe escoger en el menú Capas - Editar - Guardar, o seleccionando la opción Guardar en el submenú del botón Empezar a editar, o mediante la combinación de teclas ctrl+S. Una vez guardada

la capa, el nombre de la capa y el tipo de capa que se encuentra en la lista de capas, son actualizados. No se puede guardar una capa sin elementos.

**Cancelar edición:** Con esto se cancela la edición de la capa, y se pierden todos los cambios realizados. Para realizar esta operación, se debe escoger en el menú Capas - Editar - Cancelar, o seleccionando la opción Cancelar en el submenú del botón Empezar a editar, o mediante la combinación de teclas ctrl+E.

Para acceder al submenú del botón Empezar a editar, se debe pulsar la flecha que se encuentra al lado de este botón. Con cada operación seleccionada, la imagen de este botón y el texto en la barra de estado cambian de forma que se sepa que operación se esta realizando.



## GLOSARIO

*ALFANUMÉRICO*: conjunto de letras, números y otros símbolos, como signos de puntuación o símbolos matemáticos. Hace referencia a los caracteres del teclado y al conjunto de caracteres disponibles para las diferentes operaciones de transferencia de datos del ordenador.

*ANILLO*: un conjunto ordenado de vértices X, Y en los cuales el primer vértice tiene la misma ubicación que el último; una polilínea cerrada o un polígono.

*ARC MACRO LANGUAGE (AML)*: ARC Macro Language es un lenguaje algorítmico de alto nivel que provee capacidades completas de programación y un conjunto de herramientas para hacer a la medida la interfaz de usuario de su aplicación.

*ARCHIVO DE ÍNDICE*: en ArcView GIS el archivo de índice de un shapefile es un archivo que permite el acceso directo a los registros en el correspondiente archivo principal.

*BINARIO*: sistema de numeración en base 2, de modo que sólo hay dos dígitos posibles: el 0 y el 1. Para formar números "grandes", se usan varios dígitos binarios, que representan cada una de las potencias de 2.

*BYTE*: es la unidad básica de información. En la práctica, se puede considerar que un byte es la cantidad de espacio necesaria para almacenar una letra. Tiene múltiplos como el Kilobyte, Megabyte, Gigabyte y Terabyte. Internamente, corresponde a 8 bits.

*CAJA LÍMITE*: una caja límite es un rectángulo que rodea cada figura (Ej. Polilínea) que es apenas lo suficientemente grande para contener la figura entera. Está definida como Xmin, Ymin, Xmax, Ymax.

*COVERAGE*: 1. versión digital de un mapa que forma la unidad básica de almacenamiento de datos vectoriales en el software ARC/INFO. Un coverage almacena características geográficas como características primarias (tales como arcos, nodos, polígonos, y puntos de etiqueta) y características secundarias (tales como extensión del mapa, vínculos, y anotaciones). Tablas con atributos asociados a las figuras describen y almacenan atributos de los objetos geográficos.

2. un conjunto de datos temáticos asociados considerados como una unidad. Un coverage normalmente representa un solo tema, como ríos, caminos, o uso de tierra.

*DBASE*: gestor de bases de datos, realizado por Ashton Tate y posteriormente adquirido y mejorado por Borland.

*DBMS*: sigla en inglés de Database Management System (Sistema de administración de base de datos). Un DBMS es sustancialmente un software que se coloca entre el usuario y los datos como tales. Gracias a este estrato intermedio el usuario y las aplicaciones no acceden a los datos tal y como se memorizan efectivamente, es decir a su representación física, sino que se ve sólo una representación lógica. Esto permite un grado elevado de independencia entre las aplicaciones y la memorización física de los datos.

*DCL*: parte del SQL que permite administrar los permisos y los usuarios de las bases de datos.

*DLL*: parte del SQL que permite crear y definir nuevas bases de datos, campos e índices.

*DML*: parte del SQL que permite generar consultas ordenar, filtrar, extraer, insertar o eliminar datos de la base de datos.

*DOS*: sistema operativo de disco (Disk Operating System). Se trata de un sistema operativo monousuario y monotarea. Hay diversas versiones, con distintos nombres según la casa que lo desarrolle: MsDos (Microsoft), DrDos (Digital Research), PcDos (IBM), Novell Dos (Novell), etc.

*FOTOGRAFETRÍA*: ciencia desarrollada para obtener medidas reales a partir de fotografías, tanto terrestres como aéreas, para realizar mapas topográficos, mediciones y otras aplicaciones geográficas. Normalmente se utilizan fotografías tomadas por una cámara especial situada en un avión o en un satélite. Las distorsiones de las fotografías se corrigen utilizando un aparato denominado restituidor fotogramétrico. Este proyector crea una imagen tridimensional al combinar fotografías superpuestas del mismo terreno tomadas desde ángulos diferentes. Los límites, las carreteras y otros elementos se trazan a partir de esta imagen para obtener una base sobre la cual se realizará el mapa.

*INFORMACIÓN GEOESPACIAL*: es aquella información de entes o eventos en la que está incluida la referencia a su localización en la superficie de la Tierra o las proximidades de esta.

*INTERFAZ*: conexión de un ordenador con el exterior, o entre dos dispositivos. Generalmente se refiere al entorno de interacción de los usuarios con un programa software.

*METADATOS*: literalmente "sobre datos", son datos que describen otros datos. En general, un grupo de metadatos se refiere a un grupo de datos, llamado recurso. El concepto de metadatos es análogo al uso de índices para localizar objetos en vez de datos. Por ejemplo, en una biblioteca se usan fichas que especifican autores, títulos, casas editoriales y lugares para buscar libros. Así, los metadatos ayudan a ubicar datos.

*MULTIPUNTO*: un objeto compuesto de un grupo de puntos y un solo registro de atributos. El grupo de puntos representa el objeto geográfico.

*ORDEN DE BYTES BIG ENDIAN*: consiste en el ordenamiento de bytes para una palabra entera de izquierda a derecha. Este método de ordenamiento es usado en muchos sistemas UNIX incluyendo Sun, Hewlett-Packard®, IBM® y Data General AviiON®.

*ORDEN DE BYTES LITTLE ENDIAN*: consiste en el ordenamiento de bytes para una palabra entera de derecha a izquierda. Este método de ordenamiento es usado en muchos sistemas operativos que incluyen DEC OSF/1™, DEC OpenVMS™, MS-DOS®, y Windows NT™.

*POLILÍNEA*: un conjunto ordenado de vértices X, Y que representan una línea o frontera.

*RASTER*: un tipo de imágenes para ordenador, en las que se almacena información sobre los puntos que las componen y el color de cada punto (al contrario que en las imágenes vectoriales). Esto supone que al ampliar la imagen se pierde definición, se ven "puntos gordos".

*RETROALIMENTACIÓN*: la retroalimentación se produce cuando las salidas del sistema o la influencia de las salidas del sistema en el contexto, vuelven a ingresar al sistema como recursos o información. La retroalimentación permite el control de un sistema y que el mismo tome medidas de corrección en base a la información retroalimentada. En el desarrollo de software se refiere a modificar el diseño del programa de acuerdo a conceptos de los usuarios.

*SEGMENTOS COLINEALES*: se dice de dos segmentos de recta consecutivos que tienen en común solamente un extremo y que no pertenecen a la misma recta.

*SISTEMA GPS*: el Global Positioning System (GPS) o Sistema de Posicionamiento Mundial (aunque se le suele conocer más con las siglas GPS su nombre más correcto es NAVSTAR GPS) es un Sistema Global de Navegación

por Satélite (GNSS) el cual permite determinar en todo el mundo la posición de una persona, un vehículo o una nave, con una precisión hasta de centímetros usando GPS diferencial, aunque lo habitual son unos pocos metros. El sistema fue desarrollado e instalado, y actualmente es operado, por el Departamento de Defensa de los Estados Unidos.

*TELEDETECCIÓN*: la teledetección o lo que es lo mismo la percepción remota es la ciencia de adquirir y procesar información de la superficie terrestre desde sensores instalados en plataformas espaciales, gracias a la interacción de la energía electromagnética que existe entre el sensor y la tierra

*VECTORIAL*: un tipo de imágenes para ordenador, en las que se almacena información sobre las líneas y figuras geométricas que las componen. Esto permite que no pierdan definición si se amplían, al contrario de lo que ocurre con las imágenes Raster.

*XBASE*: nombre genérico que se suele dar al lenguaje de programación para bases de datos que usaba dBase y que aplicaron (y ampliaron) otras herramientas como Clipper o FoxPro.