

# TRABAJO DE GRADO

## Corrección Automática y Semiautomática de Datos Geográficos



**Integrantes:** Andrés, Martín Diego  
Rovedo, Pablo Andrés  
Spinetti, Mauricio Gustavo

**Director:** Gordillo, Silvia

Setiembre 2000

<p>TES 00/1 DIF-02090 SALA</p>	<p> UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMATICA Biblioteca 50 y 120 La Plata catalogo.info.unlp.edu.ar biblioteca@info.unlp.edu.ar</p> <p> DIF-02090</p>
--	---

DONACION.....  
\$.  
Fecha.....  
Inv. E. Dk-32 Inv. B. 2090.....

TES
00/1 eg. 1

# Índice General

<b>Prólogo</b> .....	v
<b>Agradecimientos</b> .....	vi
<b>Capítulo I: Introducción</b>	
1.1 Qué es GIS ? .....	1
1.2 Cómo trabaja GIS	3
1.2.1 Referencias Geográficas	4
1.2.2 Modelos Raster y Vectorizado	4
1.2.2.1 Raster	5
1.2.2.2 Vectorizado	5
1.2.2.3 Comparación de los modelos Vectorizado y Raster	6
1.3 Componentes de un GIS	7
1.3.1 Hardware	7
1.3.2 Software	7
1.3.3 Datos	8
1.3.4 Personas	8
1.3.5 Métodos	8
1.4 Tareas de GIS	8
1.4.1 Entrada	9
1.4.2 Manejo de Datos (almacenamiento de datos y recuperación)	10
1.4.3 Manipulación y Análisis	10
1.4.3.1 Análisis de Proximidad	11
1.4.3.2 Análisis de Superposición	12
1.4.4 Salida .....	12
1.5 Campos de Aplicación	12
1.6 Tecnologías Relacionadas	14
1.6.1 Sistema de Mapas	15
1.6.2 Medición remota y GPS	15
1.6.3 CAD	15
1.6.4 DBMS	16
<b>Capítulo II: Captura de Datos</b>	
2.1 Introducción a la Captura de Datos	18
2.2 Técnicas de captura de datos espaciales	19
2.3 Captura de datos gráficos con técnica raster	19
2.4 Captura de datos gráficos con técnica vectorial	20

2.4.1 Digitalización con tablero digitalizador .....	20
2.4.2 Digitalización con un sistema posicionador global o local .....	23
2.5 Funciones de captura .....	23
2.5.1 Funciones de digitalización .....	23
2.5.2 Funciones de filtrado de líneas .....	24
2.5.3 Funciones de transformación de coordenadas .....	26
2.6 El modelo de datos vectorial .....	29
2.6.1 Modelo de Datos Spaghetti .....	30
2.6.2 Modelo Topológico .....	31

### Capítulo III: Funciones de Análisis y Localización de Errores

3.1 Funciones de Localización de Errores .....	35
3.1.1 Funciones de Borrado Selectivo .....	36
3.1.1.1 Borrado de nodos que no tienen ningún arco (Delete Dangling Nodes) ....	36
3.1.1.2 Unión de nodos próximos ( <i>Node Clustering</i> ) .....	37
3.1.1.3 Intersección entre todos los arcos de la topología ( <i>Nodify</i> ) .....	37
3.1.1.4 Arco con un extremo sin conectar a otro arco ( <i>Extend Undershoots</i> ) .....	38
3.1.1.5 Eliminación de arcos demasiado pequeños ( <i>Drop Short Links</i> ) .....	38
3.1.1.6 Eliminación de polígonos de área muy pequeña .....	39
3.1.1.7 Unión de polígonos o eliminación de bordes comunes .....	39
3.1.2 Funciones de Creación de Topologías .....	40

### Capítulo IV: Implementación

4.1 Descripción del desarrollo .....	45
4.2 Implementación del Modelo de Datos .....	47
4.3 Descripción Funcional de las operaciones .....	49
4.3.1 Carga del Shapefile en Memoria .....	49
4.3.2 Descripción de las operaciones .....	50
4.3.2.1 Nodificación .....	50
4.3.2.2 Eliminación de arcos demasiado pequeños ( <i>Drop Short Links</i> ) .....	54
4.3.2.3 Borrado de nodos que no tienen ningún arco (Delete Dangling Nodes) ....	55
4.3.2.4 Unión de nodos próximos ( <i>Node Clustering</i> ) .....	57
4.3.2.5 Arco con un extremo sin conectar a otro arco ( <i>Extend Undershoots</i> ) .....	60
4.3.2.6 Eliminación de arcos duplicados ( <i>Delete duplicated links</i> ) .....	62

Conclusión .....	65
------------------	----

**Apéndices**

A.1 Apéndice A: Preliminares Matemáticos	68
A.2 Apéndice B: Descripción del Formato ShapeFile	73
A.3 Apéndice C: Espacio de Coordenadas y Transformaciones	86

<b>Bibliografía</b> .....	93
---------------------------	----

## **Prólogo**

Los datos espaciales se caracterizan por poseer una posición que los ubica en la esfera terrestre utilizando un sistema de referencia absoluto o relativo, y una representación que determina la dimensión del objeto. Los objetos 0-dimensionales se representan por puntos, los 1-dimensionales por líneas y los 2-dimensionales por polígonos.

La captura de datos en este tipo de sistemas se puede realizar de diferentes maneras: datos de campo, scanners, satélites, mesa digitalizadora, etc.; y uno de los elementos principales a tener en cuenta es la exactitud y precisión de la información capturada.

Debido justamente a la complejidad, tanto sea de la información como de los mecanismos para obtenerla, se producen errores que luego deben ser corregidos por software. Por ejemplo, es sumamente común encontrar que las líneas que definen un polígono cerrado no se hayan definido con la precisión (o la exactitud) adecuada y, por lo tanto, el polígono no quede cerrado.

En el momento de operar con los datos obtenidos mediante el proceso de captura, también se deben tomar ciertas decisiones. Por ejemplo, cuando dos entidades gráficas están lo suficientemente cerca, es necesario discernir si deben o no unificarse. Estas decisiones deben tomarse en función del contexto en el cual se está trabajando (escala de la información, precisión deseada, tolerancia a errores, etc.).

Asimismo, y a fin de poder operar con la información, es necesario definir algunos elementos geográficos por software, como por ejemplo definir un nodo donde existe la intersección entre dos líneas.

Nuestro trabajo consiste fundamentalmente en analizar los errores más comunes que se producen durante el proceso de captura de datos, y proponer una solución automática o semiautomática para la detección y posterior corrección de los mismos. Hemos dividido nuestro trabajo en cuatro capítulos.

El capítulo 1, consiste en una introducción a los sistemas GIS, conceptos básicos, campos de aplicación y tecnologías relacionadas. Además se describen los

componentes fundamentales y las áreas donde resulta adecuada la utilización de este tipo de sistemas.

En el capítulo 2, se explica el proceso comprendido desde la obtención de datos geográficos hasta su posterior conversión a los diferentes formatos utilizados por los sistemas GIS. El proceso de captura es de suma importancia dentro de la operatoria del GIS, debido a que todas las conclusiones que se obtengan a partir del mismo dependen directamente de la precisión y exactitud de los datos de entrada. Luego se analizan ventajas y desventajas de las dos técnicas de representación digitalizada de los datos; raster y vectorial.

En el capítulo 3, se describen los errores más comunes que se producen durante el proceso de captura de datos, y se clasifican según las características de los mismos.

Se estudia en detalle el espectro de errores y se propone una solución para su corrección. Ciertos tipos de errores pueden solucionarse de manera automática mientras que otros necesitan de la intervención manual de un operador (corrección semiautomática).

En el último capítulo se propone una estructura de datos que permita representar la información ingresada al GIS. También se proponen un conjunto de algoritmos para detectar y corregir los errores descritos en el capítulo anterior basándonos en la estructura de datos elegida para la modelización.

Para finalizar, y con el objeto de profundizar diferentes conceptos que fueron utilizados a lo largo del desarrollo del trabajo, hemos anexado tres apéndices que describen los siguientes puntos: *conceptos básicos de geometría analítica, descripción detallada del formato de representación de datos geográficos utilizado y transformaciones de coordenadas bidimensionales.*

## **Agradecimientos:**

Agradecemos a toda la gente que nos apoyó en este trabajo, a nuestras novias/ esposas y amigos que nos dieron sus opiniones y nos ayudaron con la redacción y con

las correcciones. Queremos agradecer especialmente a Silvia Gordillo y a Cali Meza por toda la experiencia y conocimiento que nos brindaron a lo largo del trabajo, por el tiempo que nos dedicaron, y por estar siempre disponibles y bien predispuestos para nuestras consultas en horarios poco habituales. También agradecemos a Luis Proz que nos dio una mano con la impresión. Sinceramente gracias a todos.



# **Capítulo I**

## **Introducción**

## 1.1 Qué es GIS ?

Un sistema de información geográfica (GIS) es una herramienta de software para mapear y analizar eventos que ocurren en la tierra. La tecnología GIS integra operaciones comunes de base de datos, tales como consultas y análisis estadísticos, con los beneficios de visualización y análisis geográfico que brindan los mapas.

Estas características distinguen a los sistemas GIS de otros sistemas de información y los hacen sumamente útiles para un rango amplio de empresas, públicas y privadas, al momento de explicar eventos, predecir resultados o planear estrategias.

Superpoblación, polución, deforestación y desastres naturales son ejemplos de eventos que tienen una dimensión geográfica crítica. Encontrar el mejor suelo para una plantación de bananas, o determinar el mejor trayecto para un vehículo de emergencia son problemas locales que también tienen una fuerte componente geográfica.

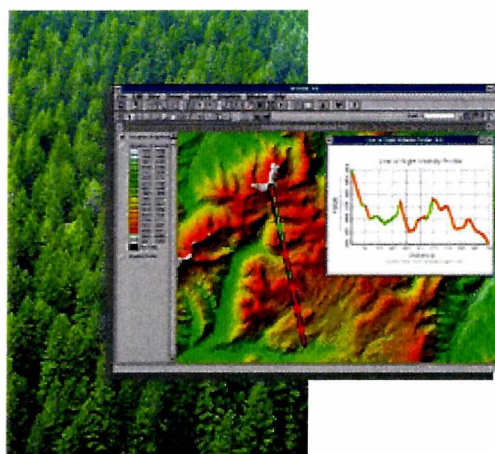


Figura 1.1: Utilización del GIS en forestación

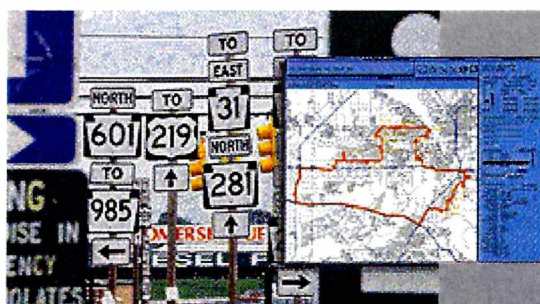


Figura 1.2: Utilización del GIS en control de tránsito

Un GIS permite crear mapas, integrar información, visualizar escenarios, resolver problemas complicados, presentar ideas de utilidad y desarrollar nuevas ideas. GIS es una herramienta usada por organizaciones, escuelas, gobiernos, etc., innovando las formas de resolver problemas.

La confección de mapas y el análisis geográfico no es algo nuevo, pero un GIS proporciona herramientas para desarrollar estas tareas en forma más eficiente que los antiguos métodos manuales. De hecho, antes de la tecnología GIS, solo unos pocos tenían las habilidades adecuadas para manejar la información geográfica necesaria en la toma de decisiones y la resolución de problemas

Si lo tomamos en un sentido más amplio, un sistema de información geográfica es cualquier sistema manual o asistido por computadora que manipula datos referenciados geográficamente. La definición que vamos a usar de aquí en adelante es la dada en [1].

*Un GIS es un sistema basado en computadora que provee los siguientes conjuntos de capacidades para manejar datos referenciados:*

- *Entrada*
- *Manejo de Datos (almacenamiento de datos y recuperación)*
- *Manipulación y Análisis*
- *Salida*

## **1.2 Cómo trabaja un GIS**

Un GIS puede almacenar información sobre el mundo como una colección de capas temáticas que pueden ser vinculadas unas con otras geográficamente. Este simple pero extremadamente poderoso concepto tiene un invaluable resultado para resolver muchos de

los problemas del mundo real, desde controlar trayectos de vehículos de reparto a modelar la circulación de la atmósfera global.



Figura 1.3: Capas temáticas vinculadas geográficamente

### 1.2.1 Referencias Geográficas

La información geográfica contiene o una *referencia geográfica explícita*, tal como latitud y longitud o una *referencia implícita* tal como una dirección, un código postal, un nombre de calle, etc. Un proceso automático llamado georeferenciación (geocoding) se utiliza para crear referencias geográficas explícitas (locaciones múltiples) desde referencias implícitas (descripciones tales como direcciones). Estas referencias geográficas permiten localizar características sobre la superficie de la tierra para su posterior análisis, tales como un negocio o puesto forestal, y diversos eventos como un terremoto o incendio.

### 1.2.2 Modelos Raster y Vectorizado

Los GIS trabajan con dos tipos diferentes de modelos geográficos – el modelo “vectorial” y el modelo “raster”.

### 1.2.2.1 Raster

En el modelo raster los datos se almacenan y presentan con una matriz de puntos. Si se trata de una representación en blanco y negro, a cada uno de los puntos se le asigna un bit de información; por ejemplo, si el punto está en negro, se almacena un uno, y si está en blanco, se almacena un cero. Si la representación debe mantener diferentes tonalidades de gris o colores, se tiene que asignar a cada punto más de un bit. Así, con cuatro bits por punto se pueden representar 16 colores o tonalidades de gris.

Las imágenes o dibujos raster tienen el inconveniente de requerir mucha memoria, ya que hay que almacenar todos los puntos que hay en la matriz, tengan o no gráficos.

Existen técnicas de compresión de imágenes que permiten reducir el espacio ocupado por una imagen, sin embargo esta compresión sólo sirve para el almacenamiento, y no para su presentación y posterior tratamiento. El modelo raster tiene la ventaja de que permite la representación de formas gráficas complejas, tal como una fotografía.

### 1.2.2.2 Vectorizado

En el modelo vectorial, la información de puntos, líneas y polígonos se codifica y almacena como una colección de coordenadas  $x, y$ . La ubicación de un punto, tal como un pozo de petróleo, puede describirse como una simple coordenada  $x, y$ . Una línea, que represente una ruta o un río, puede almacenarse como una colección de puntos. Un polígono, representando zonas de ventas o divisiones de estados o provincias, puede almacenarse como una lista de coordenadas, cuya coordenada inicial sea igual a la final.

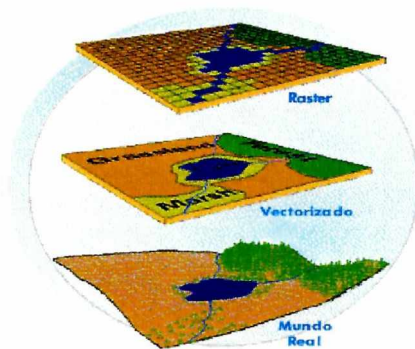


Figura 1.4: Representación Raster y Vectorizada

### 1.2.2.3 Comparación de los modelos Vectorizado y Raster

A continuación se listan las ventajas y desventajas de cada uno de los modelos. Por supuesto, la elección de cada modelo depende de que tipo de análisis se quiera hacer con la información

Modelo Raster	Modelo Vectorizado
<p><b>Ventajas</b></p> <ol style="list-style-type: none"> <li>1. Es una estructura de datos simple</li> <li>2. Sencillez en los procesos de comparación entre mapas temáticos basados en la misma retícula.</li> <li>3. La alta variabilidad espacial es representada eficientemente.</li> <li>4. Facilidad de captura de datos con un scanner.</li> <li>5. Las imágenes digitales son manipuladas con mayor eficiencia</li> </ol> <p><b>Desventajas</b></p> <ol style="list-style-type: none"> <li>1. Los gráficos almacenados con técnica raster ocupan mucho espacio en memoria.</li> <li>2. La representación en cuadrículas es poco adecuada para representar entidades lineales, tales como carreteras, ríos, etc.</li> <li>3. Tiene en general poca precisión en los cálculos de superficie, distancias, etc, ya que por motivos de espacio suele ser imposible utilizar celdas muy pequeñas, por lo que este modelo es valido para análisis globales a grandes escalas.</li> </ol>	<p><b>Ventajas</b></p> <ol style="list-style-type: none"> <li>1. Provee una estructura de datos más compacta que el modelo raster.</li> <li>2. La representación es más precisa que en el modelo raster, sobre todo las entidades lineales muy pequeñas o entidades puntuales.</li> <li>3. Permite medir distancias, superficies y volúmenes de forma más precisa que con los sistemas raster.</li> <li>4. Permite modificar fácilmente la escala y grado de detalle de un mapa o gráfico.</li> </ol> <p><b>Desventajas</b></p> <ol style="list-style-type: none"> <li>1. Es una estructura de datos más compleja.</li> <li>2. No permite representar de forma satisfactoria entidades complejas, tales como fotografías, paisajes, etc.</li> <li>3. La comparación entre diferentes gráficas requiere mucho tiempo de proceso y es poco flexible para la realización de análisis y simulaciones en tiempo real.</li> </ol>

## 1.3 Componentes de un GIS



Figura 1.5: Componentes fundamentales de un GIS

Un GIS integra cinco componentes fundamentales: hardware, software, datos, gente, y métodos

### 1.3.1 Hardware

El hardware es la computadora en la cual opera el GIS. Actualmente un GIS corre en un amplio rango de tipos de hardware, desde supercomputadoras a computadoras hogareñas con conexión o sin conexión a una red.

### 1.3.2 Software

El software GIS provee funciones y herramientas necesarias para almacenar, analizar y mostrar información geográfica. Los componentes fundamentales son:

- *Herramientas para la entrada y manipulación de información geográfica*
- *Un sistema manejador de base de datos (DBMS)*
- *Herramientas que soportan consultas geográficas, análisis, y visualización*
- *Una herramienta gráfica de usuario (GUI) para acceder fácilmente a las herramientas*

### 1.3.3 Datos

El componente posiblemente más importante de un GIS son los datos. Los datos geográficos y los datos descriptivos pueden recolectarse en casa o comprarse a un proveedor comercial de datos. Un GIS integrará datos espaciales con otras fuentes de datos, y puede incluso, usar manejadores de bases de datos (DBMS), para manipular los datos espaciales. Los DBMS se utilizan por la mayoría de las organizaciones para clasificar y manejar sus datos.

### 1.3.4 Personas

Las tecnologías GIS son de un valor limitado sin gente que manipule estos sistemas y desarrolle planes para aplicarlos en los problemas del mundo real. Los usuarios GIS pueden abarcar un amplio rango; desde los especialistas técnicos que diseñan y mantienen los sistemas a aquellos que los usan para desarrollar su trabajo diario.

### 1.3.5 Métodos

Un GIS exitoso opera acorde a un buen plan de diseño y reglas de negocios, las cuales son modeladas y puestas en funcionamiento de acuerdo a los requerimientos de cada empresa en particular.

## 1.4 Tareas de GIS

Los GIS de propósito general desarrollan esencialmente cuatro procesos o tareas.

- *Entrada*
- *Manejo de Datos (almacenamiento de datos y recuperación)*



- *Manipulación y Análisis*
- *Salida*

### **1.4.1 Entrada**

La entrada de datos es, en general, el cuello de botella en la implementación de un GIS. La toma de datos y la construcción de grandes bases de datos puede costar cinco o diez veces más que el software y el hardware del GIS. La entrada de datos puede tomar meses a años para completarse. Este gasto de tiempo inicial necesario para llevar a un GIS a su operación completa debe ser previsto y presupuestado desde el comienzo. Por otro lado, la presión para mostrar resultados lo antes posible puede comprometer el procedimiento de entrada de datos. Cuando los tiempos en el procedimiento inicial se acortan por falta de presupuesto o necesidad de resultados a corto o mediano plazo, se pueden cometer errores en otras etapas que son muy difíciles de resolver.

Los métodos de entrada de datos y los estándares de calidad de los datos, deberían ser cuidadosamente considerados en una etapa anterior al comienzo de entrada de información.

Antes de que los datos geográficos puedan ser usados en un GIS, estos deben ser convertidos a un formato digital adecuado. Los datos geográficos se proveen comúnmente como mapas en papel, tablas de atributos, archivos electrónicos de mapas y atributos asociados, fotos aéreas, imágenes satelitales y datos de campo. El proceso de conversión de mapas en papel a archivos electrónicos se llama digitalización. Las tecnologías modernas de GIS pueden automatizar completamente este proceso para grandes proyectos usando tecnología con scanners, los trabajos más pequeños pueden requerir alguna digitalización manual (usando una tabla digitalizadora). Actualmente muchos tipos de datos geográficos ya existen en formatos aceptados por los GIS, es decir son GIS-compatibles. Estos datos pueden ser obtenidos por terceros que se encargan de la recolección de los datos requeridos, que luego son almacenados directamente en el GIS.

### **1.4.2 Manejo de Datos (almacenamiento de datos y recuperación)**

Para proyectos GIS pequeños puede resultar suficiente almacenar información geográfica como simples archivos. Sin embargo, cuando el volumen de los datos se vuelve más grande y el número de usuarios que los utiliza aumenta, es un buen punto para tomar en consideración el uso de sistemas de manejo de base de datos para ayudar a almacenar, organizar y manejar datos. Un DBMS no es más que un software de computadora para manejar bases de datos. Existen diferentes diseños de DBMS, pero en el caso de GIS el diseño relacional ha sido el más utilizado. En el diseño relacional, los datos se almacenan conceptualmente como un conjunto de tablas. Este simple diseño es el más común debido a su flexibilidad y amplia difusión en aplicaciones que usan o no tecnología GIS.

La forma en que se estructuran los datos (estructura de datos) y la forma en que se relacionan unos con otros los archivos (la organización de la base de datos) establece restricciones en los métodos de recuperación de datos y la velocidad en que los mismos pueden obtenerse. Por estos motivos es necesario contar en esta etapa con una persona especializada en diseño y análisis de bases de datos.

### **1.4.3 Manipulación y Análisis**

Es común que los tipos de datos que se requieren para un proyecto GIS en particular necesiten ser transformados o manipulados de alguna forma para hacerlos compatibles con su sistema. Por ejemplo, la información geográfica puede estar disponible en diferentes escalas. Antes de que esta información pueda ser integrada, debe ser previamente transformada a la misma escala (grado de detalle o precisión). Esta podría ser una transformación temporaria para mostrar datos requeridos para el análisis. La tecnología GIS ofrece muchas herramientas para manipular datos espaciales y para descartar datos innecesarios.

Una vez que el GIS está funcionando, se puede comenzar la etapa de análisis de los datos, pudiéndose realizar consultas simples tales como:

- *¿Quién es el propietario de la parcela de la esquina?*
- *¿Dónde esta la zona de tierras para uso industrial?*

y preguntas de carácter analítico tales como:

- *¿Cuáles son los sitios más adecuados para construir casas?*
- *¿Si construyo una nueva autopista aquí, como se verá afectado el tránsito?*

Un GIS provee capacidades de consulta y herramientas de análisis sofisticadas para proveer información tanto a directores como a analistas al mismo tiempo. La tecnología GIS se vuelve realmente interesante cuando se usa para análisis de datos geográficos y búsqueda de patrones y tendencias. Los GIS modernos tienen herramientas de análisis muy poderosas, pero hay dos que son especialmente importantes:

#### 1.4.3.1 Análisis de Proximidad

A continuación se muestran dos tipos de consultas típicas sobre un GIS

- *¿Cuántas casas se encuentran a menos de 100m del conducto de agua principal?*
- *¿Cuál es el número total de clientes que se encuentra en un radio de 10km de un mercado?*

Para responder a tales preguntas, la tecnología GIS usa un proceso llamado “buffering” que determina la relación de proximidad entre las diferentes características.



**Figura 1.6:** Análisis de proximidad catastral

### 1.4.3.2 Análisis de Superposición

La integración de diferentes capas de datos es llamada superposición. En su forma más simple esta podría ser una operación visual, pero para las operaciones analíticas se requiere de una o más capas de datos unidas físicamente. Estas superposiciones, o uniones espaciales, pueden integrar datos sobre terrenos, elevaciones y vegetación, o el propietario de una tierra con su tasación fiscal.

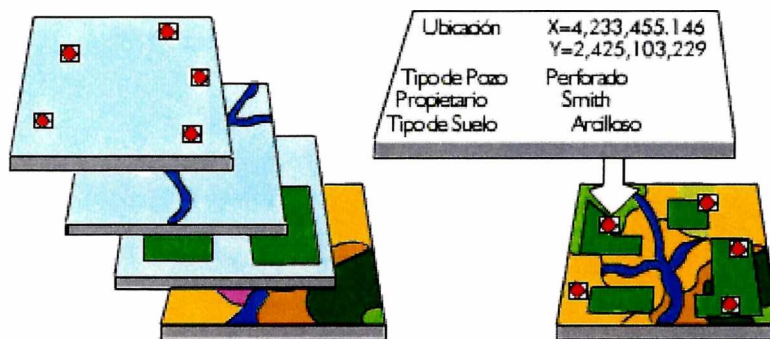


Figura 1.7: Superposición en capas

### 1.4.4 Salida

Para un amplio rango de operaciones geográficas, el resultado final se visualiza mejor en un mapa o un gráfico. Los mapas son muy eficientes para almacenar y comunicar información geográfica. Un GIS provee nuevas herramientas que extienden el arte y ciencia de la cartografía. A los mapas se les pueden adjuntar reportes, vistas tridimensionales, y otras salidas tales como multimedia (audio, video, etc).

## 1.5 Campos de Aplicación

El campo de aplicación de los sistemas GIS es amplio y puede abarcar las áreas más diversas, tales como la agricultura, medio ambiente, petróleo y minería, transporte, etc. Ejemplos de su utilización pueden ser los siguientes:

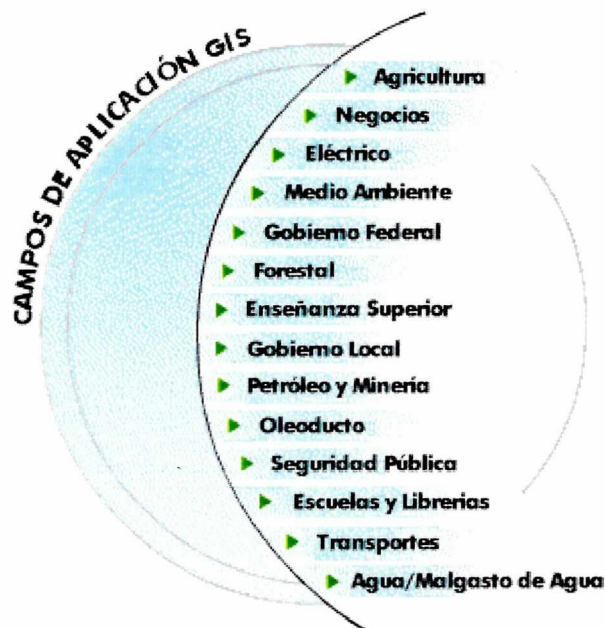


Figura 1.8: Campos de Aplicación de un GIS

En el transporte:

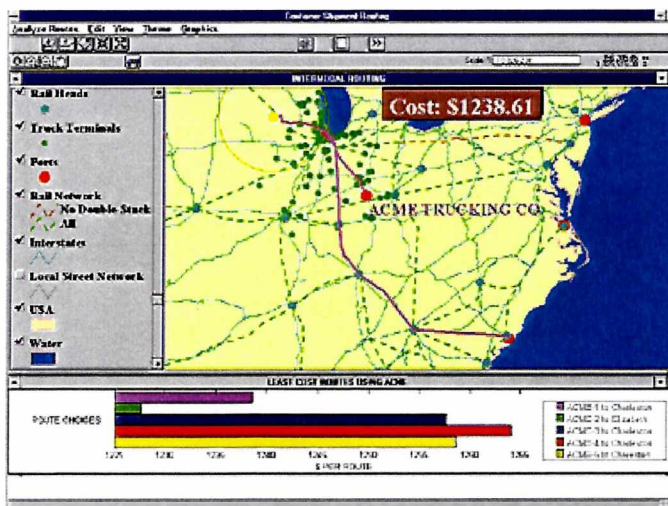


Figura 1.9: GIS aplicado al transporte

Una empresa de reparto puede usar GIS para comparar distintas alternativas de transporte tales como vía terrestre, vía aérea, por agua, etc. El costo para cada ruta

puede mostrarse en un reporte o en un gráfico para una comparación más fácil. Además puede incluirse en el análisis, las condiciones del tráfico, las ordenes de último minuto, los pedidos programados de los clientes, etc.

Seguridad Pública:

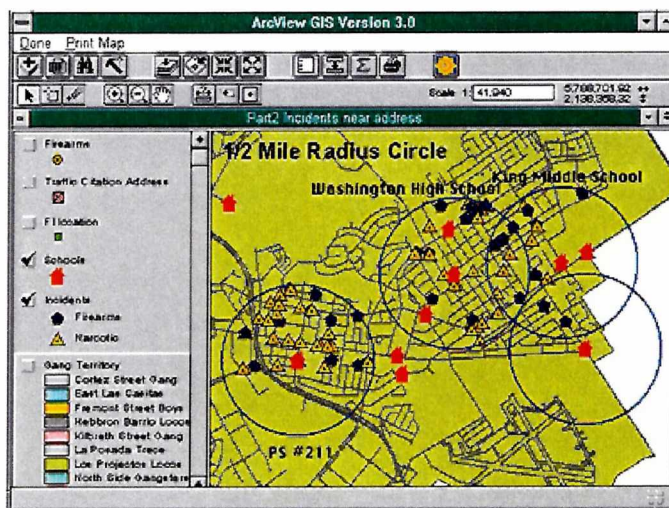


Figura 1.10: GIS aplicado a la seguridad

Utilizando GIS para ver los incidentes policiales se pueden revelar patrones ocultos. En la imagen se puede observar los incidentes de droga y armas de fuego próximos a escuelas que han sido previamente seleccionadas.

## 1.6 Tecnologías Relacionadas

Los GIS están fuertemente ligados a otros tipos de sistemas de información, pero la habilidad para manipular y analizar datos geográficos es lo que lo diferencia de otros tipos de sistemas de información.

Aunque no existen reglas estrictas para clasificar los sistemas de información podemos diferenciarlos entre GIS y sistemas de mapas, diseño asistido por computadora (CAD), medición remota, DBMS, y sistemas de posicionamiento global (GPS).

### **1.6.1 Sistema de Mapas**

Utilizan el concepto de mapa para organizar los datos y la interacción con el usuario.

El enfoque de tales aplicaciones es la creación de mapas: el mapa es la base de datos. La mayoría de los sistemas basados en mapas tienen un manejo limitado de datos, análisis espacial y capacidades de personalización. Estos sistemas operan principalmente en computadoras personales tales como PCs o pequeñas estaciones UNIX.

### **1.6.2 Medición remota y GPS**

La medición remota es la ciencia de medir la Tierra usando sensores, tales como cámaras en aviones, receptores GPS, u otros dispositivos. Estos sensores recogen los datos en forma de imágenes y proveen capacidades especializadas para manipular, analizar y visualizar esas imágenes. No pueden ser denominados GIS ya que carecen de un juego completo de operaciones para analizar y manejar los datos geográficos.

### **1.6.3 CAD**

Los sistemas CAD son adecuados para crear diseños y planos de edificios y estructuras. Esta actividad requiere que los componentes sean ensamblados para crear toda la estructura. Estos sistemas requieren pocas reglas para especificar cómo pueden ser ensamblados los componentes, y tienen una capacidad analítica limitada. Los sistemas CAD han sido extendidos para soportar mapas pero generalmente tienen una utilidad limitada para manejar y analizar grandes bases de datos geográficas.

#### **1.6.4 DBMS**

Los Sistemas de Manejo de Base de Datos se especializan en el almacenamiento y manejo de todos los tipos de datos, inclusive los geográficos. Los DBMS están optimizados para guardar y recuperar datos y muchos GIS se basan en éstos para ese propósito. Los DBMS no poseen las herramientas de visualización y análisis que son comunes en los GIS.



# **Capítulo II**

## **Captura de Datos**

## **2.1 Introducción a la Captura de Datos**

La entrada de datos es el procedimiento para ingresar la información a la computadora y alimentar el sistema GIS. La entrada es el principal cuello de botella a la hora de implementar el GIS, el costo inicial de construir la base de datos es generalmente de 5 a 10 veces mayor que el costo de hardware y del software. La creación de una base de datos precisa y bien documentada es crítica para la operación del GIS.

Los datos a ser ingresados a un GIS son de dos tipos: datos espaciales y atributos no espaciales asociados (datos alfanuméricos). Los datos espaciales representan la ubicación geográfica de las entidades gráficas típicas de la cartografía, tales como líneas, puntos y áreas, las cuales son usadas para representar calles, lagos, bosques, estados, etc. Los atributos no espaciales proveen información descriptiva como el nombre de una calle, cantidad de habitantes en una región, composición química de un suelo, etc. Los datos espaciales y los atributos deben ser ingresados y vinculados correctamente (el nombre de una calle debe estar lógicamente asociado a su representación gráfica).

En cuanto a la dificultad de captura hay mucha diferencia entre el ingreso de datos alfanuméricos, técnica conocida como grabación, y la captura de datos espaciales o digitalización. En la digitalización, la problemática es muy distinta si se trata de digitalización automática con scanner, con cámara de video o la digitalización manual con tablero digitalizador.

Nuestro trabajo esta enfocado principalmente en el ingreso de los datos espaciales y en su posterior análisis, detección y corrección de errores por lo tanto no nos extenderemos en la descripción del proceso de entrada de atributos (base de datos asociada) el cual es un proceso posterior.

## 2.2 Técnicas de captura de datos espaciales

A continuación se resume las diferentes técnicas de captura de datos espaciales

### *Captura de datos gráficos con técnica raster*

- *Digitalización con scanner.*
- *Digitalización con cámara de video o fotografía digital.*
- *Digitalización con sensores remotos.*

### *Captura de datos geográficos con técnica vectorial*

- *Digitalización con tablero.*
- *Digitalización con mouse.*
- *Vectorización de digitalización raster.*
- *Digitalización sobre terreno con aparatos de posicionamiento local o global (GPS).*

## 2.3 Captura de datos gráficos con técnica raster

La técnica raster de representación de entidades gráficas permite una fácil captura de los datos mediante técnica de barrido de imágenes; así, con una cámara de video que esté conectada a una computadora mediante una placa digitalizadora, es posible en pocos segundos pasar a un medio electrónico una foto o un mapa. Esta facilidad contrasta con la dificultad de capturar las entidades en técnicas vectoriales, que permite una fácil actualización de las mismas.

El scanner es el sistema más utilizado, que mediante células fotosensibles permite descomponer una imagen en multitud de impulsos eléctricos los cuales se almacenarán en la memoria de la computadora como bits de información. Una de las principales características de los scanners es su resolución, que mide el número de puntos por pulgada que es capaz de capturar, de forma que a mayor resolución, la imagen capturada tendrá más nitidez. Otra característica importante es el número de colores que es capaz de obtener.

Es evidente que a mayor cantidad de colores y mayor resolución, la imagen representada tendrá más calidad y nitidez si bien requerirá mucho espacio de memoria. El tamaño de la imagen a capturar es otra característica importante; así, hay scanners desde menos de una pulgada a tamaños DIN A0. En general los más pequeños, DIN A4 y DIN A3, permiten capturar una amplia gama de colores y se utiliza para capturar fotografías y mapas temáticos en los que el colorido tenga especial interés; por el contrario, los scanners de gran tamaño suelen capturar muy pocos colores y suelen estar destinados a capturar planos y mapas en blanco y negro, ya sea para almacenar en técnica raster o para convertir a técnica vectorial.

## **2.4 Captura de datos gráficos con técnica vectorial.**

### **2.4.1 Digitalización con tablero digitalizador**

La digitalización con tablero es sin duda la más utilizada y la que mejor calidad ofrece, si bien presenta algunos inconvenientes como son el costo de mano de obra y el tiempo requerido para realizar la captura. El mapa a digitalizar se fija al tablero digitalizador y sobre éste se pasa un cursor electrónico que va capturando las coordenadas por donde pasa el cursor. Hay básicamente dos tipos de cursor: el cursor plano con múltiples botones y el lápiz óptico.

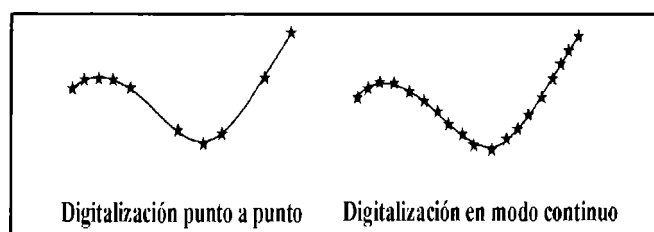


**Figura 2.1:** Tablero de Digitalización

El cursor más recomendable en la digitalización de cartografía es el cursor plano, ya que resulta más preciso que el lápiz, el cual se utiliza en sistemas CAD en los que la precisión la proporciona el sistema interno de coordenadas, tal como puntos de apoyo en una plantilla o entidades gráficas ya existentes y sobre las que se apoya el diseño o dibujo. Entre los cursores de tipo plano los más recomendables son los que tienen muchos botones programables, ya que con éstos es posible realizar múltiples operaciones directamente desde el cursor sin tener que acudir al teclado.

Las entidades que se digitalizan con tablero son las líneas, los puntos y los textos. Para la digitalización de líneas, existen dos técnicas; una, la digitalización en chorro de coordenadas continuas, para lo cual se establece la distancia mínima que debe haber entre dos grabaciones de coordenadas y el cursor automáticamente va grabando vértices de una línea a medida que el operador avanza con el cursor. Este sistema aunque parece el más cómodo para el operador, no es muy recomendable ya que graba muchos vértices superfluos, sobre todo en aquellos tramos que tienden a ser una recta. Las líneas digitalizadas con este sistema deben filtrarse posteriormente para eliminar todos los puntos superfluos, con el objetivo de definir las líneas con la menor cantidad de puntos y así optimizar la velocidad de proceso y minimizar el espacio de almacenamiento.

El otro sistema de digitalización de líneas es de punto a punto, en el cual es el operador el que decide qué puntos se graban pulsando uno de los botones del cursor plano. Con esta técnica, si el operador es cuidadoso se definen las líneas de forma óptima. Sin embargo una filtración posterior de las líneas será casi siempre necesario. En la digitalización punto a punto deben introducirse más puntos en aquellos tramos con más curvatura.



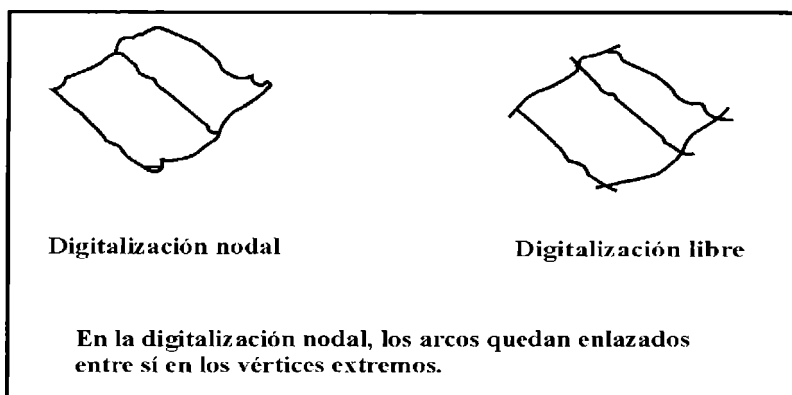
**Figura 2.2 :** Digitalización punto a punto y en modo continuo

La digitalización de cadenas de segmentos o arcos que van a formar parte de las entidades topológicas debe ser muy cuidadosa, para garantizar la continuidad entre unos arcos y otros. Esta continuidad se puede garantizar en el momento de la digitalización a través de la digitalización nodal, o posteriormente, con procesos de limpieza que garanticen todos los cierres de polígonos.

La técnica más recomendable para arcos que van a formar parte de redes de polígonos o sistemas lineales es la digitalización nodal, que consiste en que los puntos final e inicial de un arco tomen la coordenada de un punto previamente digitalizado.

Si se quiere digitalizar un arco, debe seleccionarse un punto en el que acaba otro arco o un vértice de otro arco existente, de manera que el nuevo arco se inicia en la coordenada que tiene almacenada en el arco anterior y no en la coordenada que se pulso al digitalizar. Si en un determinado punto y dentro del radio de búsqueda (distancia nodal) no existe un punto nodal, el programa de digitalización rechazará la coordenada y el operador tendrá la opción de buscar o introducir otro punto nodal con un botón del cursor.

La digitalización nodal resulta más lenta que la digitalización libre, si bien la calidad de digitalización es muy superior y resulta imprescindible cuando el sistema que se tiene instalado no cuenta con un potente sistema de limpieza de la digitalización (corrección automática y semiautomática de datos gráficos).



**Figura 2.2 :** Digitalización Nodal y digitalización libre de arcos

En la digitalización asistida o nodal es posible especificar la longitud mínima de un segmento de manera tal que el sistema rechace todos los segmentos de longitud menor al especificado.

Otro tipo de digitalización asistida es la que se realiza con una plantilla de puntos, de manera que el sistema genera una malla de puntos nodales a intervalos constantes en el sentido de las coordenadas  $x$ ,  $y$ , con lo que cualquier punto que se introduzca tendrá que pertenecer a esta plantilla. Esta técnica es muy útil cuando se digitalizan planos de arquitectura o esquemas de circuitos electrónicos en los que es importante mantener unas determinadas relaciones entre los puntos, así como garantizar los ángulos rectos de las líneas. Esta técnica no suele utilizarse en los sistemas GIS.

#### **2.4.2 Digitalización con un sistema posicionador global o local**

Un sistema de posicionamiento global (Global Positioning System, GPS) utiliza como referencia la posición de las estrellas y unos veinte satélites que orbitan la Tierra. Estos sistemas de posicionamiento global tienen el inconveniente de no ser muy precisos por lo que deberán ser rectificadas con correcciones locales. Este sistema esta basado en la triangulación, midiendo la distancia del punto en el que se mide la posición a tres o más satélites simultáneamente.

## **2.5 Funciones de captura**

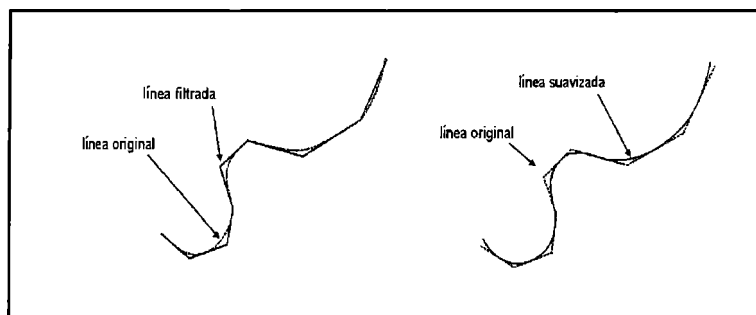
### **2.5.1 Funciones de digitalización**

Generalmente se conoce como digitalización a la captura de datos gráficos en técnica vectorial, si bien también es digitalización el proceso de captura en técnica raster, e incluso la captura de datos alfanuméricos.

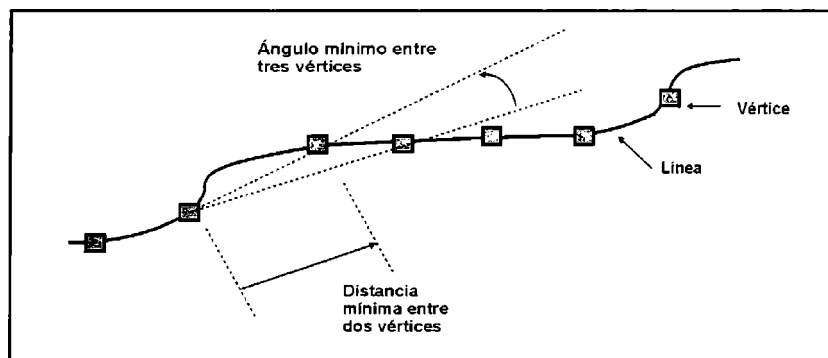
## 2.5.2 Funciones de filtrado de líneas

Las funciones de filtrado de líneas tiene como misión optimizar el número de vértices de las líneas para que sean más fáciles de procesar y ocupen menos espacio, aproximándose lo mejor posible a su forma original.

Se trata de eliminar algunos vértices cuando están muy próximos entre sí, o eliminar vértices intermedios en tramos rectos. Algunas funciones permiten incrementar vértices para que las curvas presenten una forma más suave. No debe confundirse esta técnica de representación de las líneas con el ajuste de curvas (splines), consistente en representar los arcos entre varios vértices mediante la trayectoria de una curva dada por su ecuación. El filtrado afecta los vértices que se almacenan en cada línea, mientras que la técnica de ajuste no afecta los vértices de la línea, sino que su representación se presenta más suave para seguir la trayectoria dada por una ecuación.



**Figura 2.3 :** En el filtrado se eliminan vértices del arco; en la suavización, se representa un arco por una ecuación que pase por un conjunto de vértices.



**Figura 2.4 :** Parámetros en el filtrado de arcos



La figura 2.4 muestra los parámetros de ajuste en la técnica de filtrado de arcos. En determinados sistemas, estas funciones de filtrado se asocian a otras funciones, tales como la de creación de topología, intersección de polígonos, etc., realizando una limpieza de vértices una vez que el proceso principal esté realizado. Todas las líneas que van a formar parte de un GIS deben ser filtradas para que éste se pueda gestionar más fácilmente. Es necesario establecer un parámetro de filtrado para cada escala de referencia en la que se utilizan los arcos. Aunque los mapas se pueden representar a diferentes escalas, siempre hay una escala de referencia que marca la calidad de los datos, como la escala de los mapas originales o la escala que más se utiliza para su presentación por plotter.

Es inútil mantener vértices más próximos que 0.2 mm medidos sobre papel a la escala de referencia, ya que esta distancia es el límite de discriminación de la vista sin instrumentación de ampliación, y suele ser el límite del grosor de las líneas en los plotters de pluma. Dependiendo del uso que se le dé al mapa, se definirá la distancia mínima entre vértices; para ello, se puede manejar el criterio de los 0.2 milímetros.

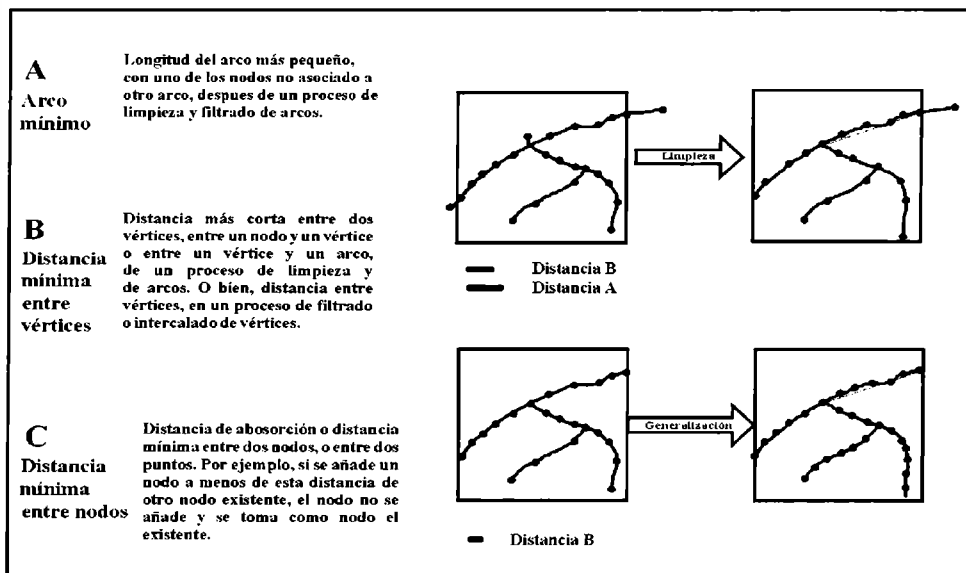


Figura 2.5 : Parámetros que permiten gestionar la distribución de vértices y nodos.

En una digitalización poco precisa, por ejemplo para visualizar en la pantalla o en forma de esquema, se podría tomar como criterio que no existan vértices a menos de 1 mm.

En la figura de arriba se recogen los diferentes parámetros de ajuste que permiten mantener una buena distribución de vértices y nodos.

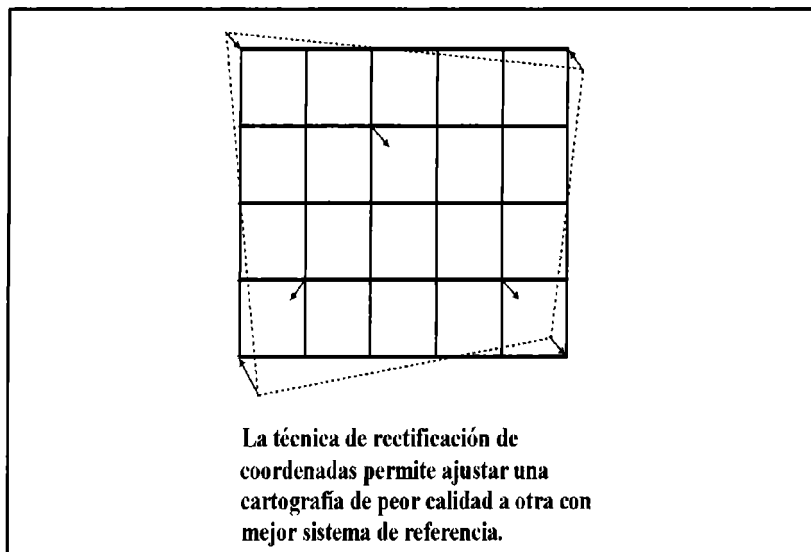
### **2.5.3 Funciones de transformación de coordenadas**

Para referenciar geográficamente las entidades de un GIS, se puede utilizar coordenadas geográficas globales con altitud, latitud o proyecciones planas con coordenadas cartesianas  $x$  e  $y$ .

Lo más sencillo es utilizar un sistema cartesiano como el definido por la proyección Mercator (UTM), que fue establecido en 1936 por la unión geodésica y geofísica internacional. Esta proyección divide la esfera terrestre en 60 planos o usos de proyección.

Las funciones de transformación de coordenadas son muy importantes en la explotación de los GIS, debido a que éstos se suelen alimentar de cartografía digitalizada procedente de muy diversas fuentes, las cuales no siempre tienen una calidad tal que posibilite que coincidan plenamente en el sistema de coordenadas utilizado por nuestro GIS.

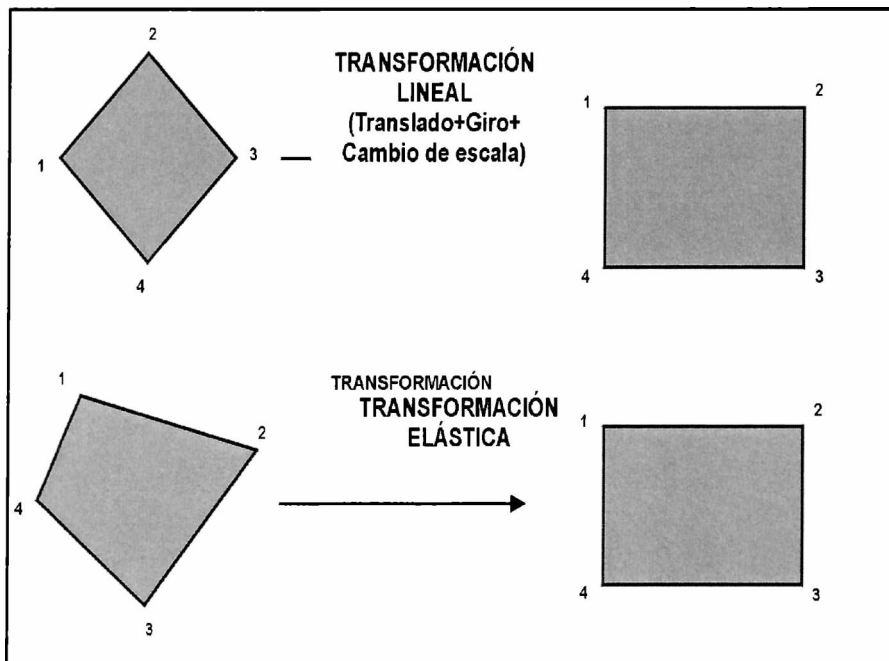
Para referenciar toda la cartografía a un mismo sistema de coordenadas, se puede utilizar uno de los mapas de mayor calidad, en cuanto a la precisión de las coordenadas y transformar los demás mapas mediante técnicas de ajuste, por ejemplo la técnica de rubber sheeting que se muestra en la Figura 2.6.



**Figura 2.6 :** Técnica de transformación elástica de coordenadas

Para ello se buscan puntos comunes entre el mapa tomado como patrón y en el mapa a transformar, por ejemplo se buscan cruces de carreteras, esquinas de edificios, etc; que estén representados en ambos mapas y se establecen los vectores de transformación entre cada pareja de puntos homónimos. Deberán seleccionarse al menos tres puntos que estén lo más alejados entre sí y que cubran lo mejor posible todo el mapa a transformar. De ser posible, deben seleccionarse cinco o más puntos, para que la transformación sea más eficaz.

Existen diversos algoritmos de transformación que van desde una simple transformación lineal, que comprende una translación, un giro y un cambio de escala, hasta transformaciones no lineales; es decir, con deformación geométrica.



**Figura 2.7 :** Transformación lineal y no lineal

La figura 2.7 muestra la diferencia entre una transformación lineal y una no lineal. Sin duda, siempre será preferible una transformación lineal, siempre que la calidad de la cartografía sea alta. Frecuentemente hay que recurrir a las transformaciones no lineales, ya que existen múltiples deformaciones de la cartografía, tanto en soporte físico como en papel, por las dilataciones, como en el digital, por inadecuada digitalización o por transformaciones entre sistemas poco precisas.

La digitalización se suele realizar en coordenadas de papel; es decir, asignando a la esquina inferior derecha de la hoja del mapa la coordenada  $x = 0$  e  $y = 0$ , transformándose posteriormente el archivo de digitalización a las coordenadas reales. Con ello se aprovecha mejor la precisión de los programas de digitalización, que suelen trabajar en simple precisión, es decir con pocos decimales, frente a los GIS, que deben trabajar en doble precisión.

## 2.6 El modelo de datos vectorial

El modelo de datos vectorial provee la ubicación precisa en el espacio de las entidades gráficas tales como puntos, líneas y polígonos.

Este modelo asume que las coordenadas son matemáticamente exactas. De hecho, el nivel de precisión está limitado por el número de bits usados para representar un valor simple, aunque esto es una resolución más precisa comparada con el tamaño de celdas generalmente usado en los sistemas raster.

La localización de las entidades gráficas en la superficie de la Tierra son representadas en un mapa usando un sistema de coordenada  $x,y$  denominado Sistema de Coordenadas Cartesianas y representados por medio de puntos, líneas y áreas. El modelo vectorial representa un punto como un simple par de coordenadas  $x,y$ ; una línea como una serie de coordenadas  $x,y$  y un área como un lazo cerrado de pares de coordenadas  $x,y$  que define el contorno del área (Un área delimitada por un lazo cerrado de segmentos rectos se denomina Polígono).

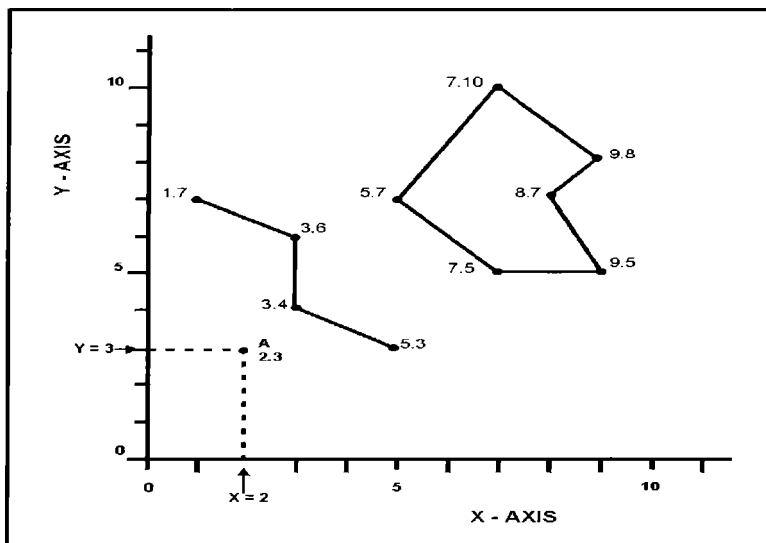


Figura 2.8 : Representación de Puntos, Líneas y Polígonos como coordenadas  $x y$

En el gráfico 2.8 las coordenadas son unidades arbitrarias. Sin embargo en un GIS las posiciones son usualmente almacenadas usando un sistema de coordenadas estándares como UTM o Latitud y Longitud.

Se han diseñado numerosos sistemas para almacenar las coordenadas de puntos, líneas y polígonos. Los primeros sistemas fueron hechos para satisfacer la necesidad de generar mapas automáticamente, donde el principal objetivo era almacenar la posición de los puntos, líneas y polígonos como así también las instrucciones para poder dibujarlos, tales como el espesor, el color de una línea, etc. En estos sistemas, la característica geográfica (lagos, caminos, estados, etc.) que estos elementos representaban no eran tenidos muy en cuenta.

Luego, los sistemas fueron evolucionando para permitir el almacenamiento de los datos geográficos como así también el reconocimiento de las características gráficas que dichos elementos representaban. Sin embargo, la información no era almacenada de una forma organizada, un ejemplo de esto es el modelo de datos vectorial denominado Modelo Spaghetti.

### 2.6.1 Modelo de Datos Spaghetti

En este modelo, el mapa en papel es traducido línea a línea en una lista de coordenadas  $x,y$ . Un punto es almacenado como un simple par de coordenadas  $x,y$  y una línea como una lista de coordenadas  $x,y$ . El área se representa por medio de un polígono y se guarda como un lazo cerrado de coordenadas  $x,y$  que definen su contorno. El borde en común entre dos polígonos adyacentes debe ser almacenado dos veces, uno para cada polígono. Un archivo de datos espaciales construido de esta manera es una colección de lista de coordenadas sin estructura inherente, de aquí el término modelo Spaghetti. La estructura es muy simple y fácil de entender. El modelo de datos es realmente el mapa expresado en coordenada cartesianas. El archivo de datos de coordenadas  $x,y$  es la estructura de datos, esto es, la forma en la cual los datos espaciales son almacenados en la computadora. Aunque son guardadas todas las representaciones espaciales, la relación entre

estos no es tenida en cuenta. Por ejemplo, no se guarda información a cerca de las características gráficas adyacentes a cada polígono. Esta información debe ser generada buscando dichas características en el archivo de datos y calculando si son o no adyacentes.

El modelo Spaghetti es muy ineficiente para la mayoría de los tipos de análisis espaciales puesto que cualquier relación espacial debe ser calculada. Sin embargo, es un modelo eficiente para reproducción de mapas puesto que la información que no tiene sentido a la hora del ploteo, como las relaciones espaciales, no se almacena.

### 2.6.2 Modelo Topológico

El modelo topológico es el método más usado para almacenar relaciones espaciales en un GIS. La topología es el método matemático para definir relaciones espaciales. La figura 2.9 ilustra un mapa y su tabla de datos asociada. Esta forma particular de modelo topológico es denominada modelo de datos arco-nodo. La entidad básica es el arco, una serie de puntos que comienzan y terminan con un nodo. Un nodo es un punto de intersección donde se encuentran dos o más arcos. Un nodo puede aparecer también al final de un arco suelto, por ejemplo un arco que no está conectado con otro arco como en el caso de una calle sin salida. Los nodos solitarios, no conectados a los arcos, representan puntos. Un polígono esta compuesto de una cadena cerrada de arcos que representan su contorno.

En la figura 2.9, la topología es guardada en tres tablas de datos, una para cada tipo de elemento espacial, y las coordenadas son almacenadas en una cuarta tabla. En un GIS, los polígonos y los puntos son almacenados a menudo en una capa de datos y las líneas en otra capa separada. En la figura los puntos, líneas y polígonos están en la misma capa de datos.

La tabla de topología de polígonos muestra los arcos que forman el contorno de cada polígono. Por ejemplo, el polígono A esta rodeado por los arcos a1, a3, y a5 (por convención los arcos que forman un polígono están definidos en sentido de las agujas del reloj). Dentro de los polígonos puede haber islas. El polígono C es una isla dentro del polígono B. Esto se indica en la lista de arcos para el polígono B poniendo un 0 delante de

la lista de arcos que forman la isla. En este caso, hay un solo arco en el polígono C. El punto en el polígono B es tratado también como un polígono, polígono D el cual está compuesto por un solo arco a6. Un punto puede ser considerado como un polígono sin área. En este ejemplo, es considerado también como un nodo aislado y como un arco formado por un solo punto (detalles cómo estos pueden ser tratados de manera diferente dependiendo del software usado). Para completar las definiciones espaciales debe existir una manera de representar el área que se encuentra fuera de los límites del mapa. Esta área es designada como polígono E, para el cual no se le definen arcos.

En la tabla de topología de nodos cada nodo es definido por los arcos a los cuales pertenece. Por ejemplo el nodo N1 es un punto final para los arcos a1, a3 y a4. El nodo N5 es un punto simple que también está definido como arco a6 y como polígono D. La tabla de topología de arcos define la relación de los nodos y los polígonos con los arcos. Los puntos finales se distinguen designando un nodo como inicio y el otro como fin. Por ejemplo el arco a5 comienza con el nodo N3 y termina con el nodo N2. Moviéndose desde N3 a N2, el polígono a la izquierda es el polígono A y el polígono a la derecha es el B. Con solo la topología (por ejemplo las tres tablas de topología) se puede realizar un análisis de la posición relativa de los elementos. Por ejemplo, se pueden encontrar todos los polígonos adyacentes al polígono B realizando una búsqueda en la tabla de topología de arcos. Todo polígono que aparezca en la tabla junto a B es adyacente a éste porque tienen un arco en común. Por ejemplo el polígono A y B se encuentran juntos en la fila correspondiente al arco a5, por lo tanto el polígono A es adyacente al polígono B.

Las tablas de topología se pueden usar para encontrar todas las representaciones gráficas contenidas dentro de un polígono realizando una búsqueda en la tabla de topología de polígonos de una lista de arcos que contengan un 0. Los arcos que se encuentran después del 0 se buscan en la tabla de topología de arcos para identificar los elementos. Se ve que el polígono B contiene dos representaciones gráficas, una definida por el arco a6 y otra por el arco a7. Por intermedio de las tablas de topología de arcos y nodos se ve que el arco a6 es un simple punto (tiene el mismo polígono a derecha e izquierda y tiene un solo nodo). El arco a7 es un polígono isla (tiene un polígono diferente a su derecha y a su izquierda y un nodo). Se pueden procesar consultas espaciales de este tipo de manera más rápida usando



las tablas de topología que realizando cálculos desde los datos coordenados (modelo spaghetti).

Para relacionar las características gráficas de un mapa con posiciones del mundo real se necesitan las coordenadas x,y. Estas se almacenan en la tabla "arcos-datos coordenados". Cada arco se representa por uno o más segmentos rectos definidos por una serie de coordenadas. Cuanto más compleja sea la figura, se necesitarán más coordenadas para representarla como una serie de segmentos rectos. Se puede obtener las coordenadas de los nodos de esta tabla referenciando a la tabla de topología de arcos en la cual se identifican los números de los nodos para los puntos de inicio y fin.

Los atributos de las características gráficas se almacenan comúnmente en un modelo relacional en la cual un dato contiene un código de identificación de la entidad espacial. Esto se ilustra en la figura 2.9

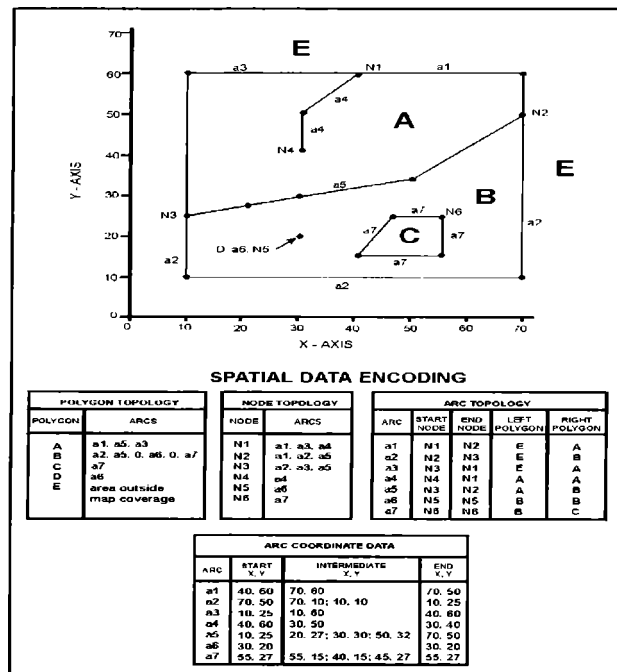


Figura 2.9 : El modelo de datos topológico

# **Capítulo III**

## **Funciones de Análisis y Localización de Errores**

### **3.1 Funciones de Localización de Errores**

La característica más específica del GIS es la posibilidad de generar relaciones topológicas entre los objetos, las cuales le permiten realizar una serie de análisis espaciales de gran interés en la gestión de datos geográficos. La organización de la topología es compleja, y además debe estar libre de errores, tanto gráficos como de codificación. Una vez que se han incorporado las capas de arcos al GIS, bien sea por digitalización o por importación de otro sistema, la primera labor es filtrar las líneas, y una vez que están optimizadas en cuanto a la cantidad de vértices, se procede a la creación de la topología. Para ello, todos los arcos que van a formar parte de un mapa temático deben intersectarse, con objeto de que no existan arcos que se crucen (modificación). Una vez que no hay arcos que se crucen, es decir, están todos partidos en las intersecciones, el proceso siguiente es localizar los nodos de unión entre arcos y los extremos de arco que no tienen otro arco al que unirse; una vez eliminados o resueltos todos los arcos sueltos se puede proceder a la definición de la topología de polígonos. Todo este proceso, que se describe más adelante, debe ser controlado a través de los sistemas de investigación de errores, el cual muestra gráficamente o mediante listado las entidades que tienen errores. A continuación se muestran las posibles operaciones para corrección de errores en los diferentes tipos de topología:

Es importante que el GIS permita la localización de estos errores ya que facilita la creación de topología y la corrección de errores, por ejemplo el que exista un polígono con dos etiquetas puede estar indicando que falta digitalizar un arco. Generalmente, estos sistemas de alarma de errores se activan en el proceso de creación de topología y no permiten la creación de la topología hasta que no esté totalmente libre de errores.

### 3.1.1 Funciones de Borrado Selectivo

Las funciones de borrado selectivo de entidades gráficas son muy útiles en los procesos de limpieza y creación de topología, así como en el proceso de análisis y simulación con objeto de simplificar o unir conjuntos de entidades. A continuación se muestran algunas de las funciones de borrado selectivo más comunes en los GIS.

#### *Operaciones sobre nodos.*

- *Borrado de nodos que no tienen ningún arco (Delete Dangling Nodes).*
- *Unión de nodos próximos (Node Clustering).*

#### *Operaciones sobre topología de arcos.*

- *Intersección entre todos los arcos de la topología (Nodify).*
- *Arco con un extremo sin conectar a otro arco (Extend Undershoots).*
- *Eliminación de arcos demasiado pequeños (Drop Short Links).*
- *Eliminación de arcos duplicados (Drop Duplicated Links).*

#### *Operaciones sobre topología de polígonos.*

- *Polígono sin centroide*
- *Polígono con más de un centroide*
- *Polígono de área nula*
- *Polígono con demasiados arcos*

#### 3.1.1.1 Borrado de nodos que no tienen ningún arco (Delete Dangling Nodes)

Este proceso consiste en la eliminación de todos los nodos que no pertenezcan a ningún arco. Inicialmente no deberían existir nodos aislados porque en una topología inicial

de arcos todos los nodos pertenecen a algún arco. Es común que se produzca este escenario luego de realizada la modificación de la topología y la posterior aplicación de operaciones de eliminación de arcos, con lo cual aquellos nodos pertenecientes a los arcos eliminados quedan aislados

### **3.1.1.2 Unión de nodos próximos (*Node Clustering*)**

El proceso consiste en el análisis de todos los posibles grupos de nodos muy próximos entre sí (cluster) donde, debido al nivel de detalles (escala) con la cual estamos trabajando podemos tomar un único nodo que represente al conjunto. Por consiguiente este proceso contribuye a la simplificación de la topología.

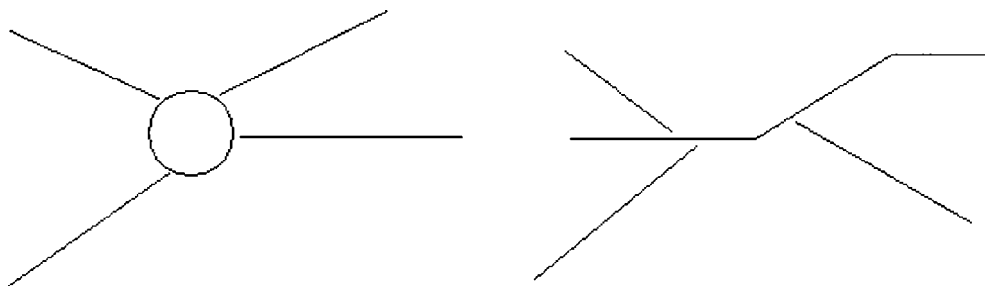
Para efectuar el proceso se debe dar un parámetro denominado tolerancia que indica el radio de un círculo alrededor del nodo más central del cluster denominado pivote. Claramente para unificar todos los nodos de cluster en el pivote, el círculo definido por la tolerancia debe tener un radio levemente mayor a la máxima distancia entre cualquier nodo y el pivote.

### **3.1.1.3 Intersección entre todos los arcos de la topología (*Nodify*)**

A este proceso se lo conoce como modificación. Es la primer operación que se debe realizar sobre una topología de arcos ya que nos permite la eliminación de todas las intersecciones entre los arcos, convirtiendo los puntos de intersección en nodos. Esta operación produce un aumento considerable en la cantidad de arcos; por otro lado, es sumamente necesaria, ya que todas las operaciones posteriores de corrección de errores se basan en el hecho de la no existencia de arcos intersectados.

### 3.1.1.4 Arco con un extremo sin conectar a otro arco (Extend Undershoots)

Este error se comete comunmente por imprecisión en la digitalización o en la conversión de datos escaneados, especialmente cuando no se utiliza digitalización nodal. Esta anomalía se produce cuando se utiliza la técnica de digitalización por defecto. En la Figura 3.1 se muestran dos ejemplos encontrados a menudo en mapas de rutas.



**Figura 3.1:** Ejemplos típicos en mapas de rutas

Cuando la intención del digitalizador es hacer que un arco (denominado **a1**) llegue a tocar a otro (denominado **a2**) pero se queda corto en el trazo se producen estas clases de errores.

Para solucionar este tipo de errores se debe extender el arco **a1** hasta que toque el arco **a2**, generándose un nuevo nodo de intersección. Esta función solo se aplicará en los casos en que la distancia entre el nodo final de **a1** y el arco **a2** sea menor al umbral de tolerancia ingresado.

### 3.1.1.5 Eliminación de arcos demasiado pequeños (Drop Short Links).

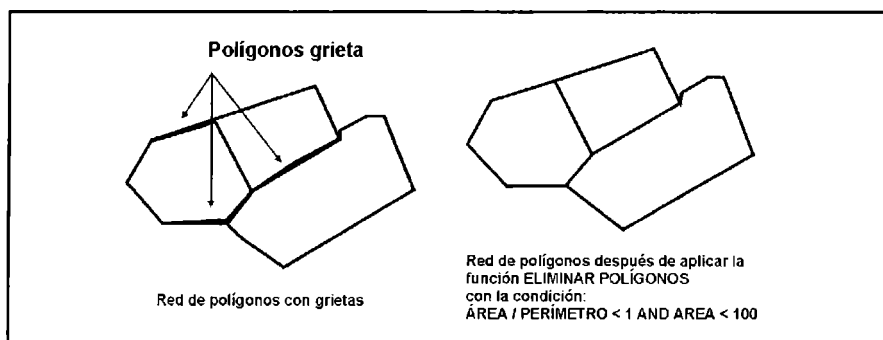
Debido a que la técnica más común de digitalización es la digitalización por exceso, al realizar el proceso de nodificación queda un gran número de arcos de tamaño muy pequeño producto del sobrante entre las intersecciones de arcos que se cruzan. También puede ocurrir que al digitalizar aparezcan pequeños arcos motivado por algún error

humano. El proceso a efectuar consiste en recorrer todos los arcos de la topología y, tomando un umbral de tolerancia (ídem a los casos anteriores), eliminar todos aquellos arcos cuya longitud sea menor a dicho umbral.

Como regla general el umbral de tolerancia debe ser levemente más pequeño que el largo del arco más corto que se quiere mantener.

### 3.1.1.6 Eliminación de polígonos de área muy pequeña

En los procesos de superposición de redes de polígonos se suelen crear multitud de polígonos de áreas muy pequeñas que no tienen ningún sentido. En la Figura 3.2 se muestra como se crean estos polígonos y como con las funciones de eliminación selectiva de entidades se suelen eliminar. En algunas ocasiones es recomendable utilizar el criterio de eliminar los polígonos cuyo perímetro dividido por el área sea mayor que una determinada cantidad, este criterio eliminará los polígonos muy largos y estrechos. Ambos criterios se pueden combinar



**Figura 3.2 :** Técnica de eliminación de polígonos grieta

### 3.1.1.7 Unión de polígonos o eliminación de bordes comunes

La unión de polígonos es una función muy útil en los procesos de análisis espacial con polígonos; generalmente, la técnica de trabajo en el análisis, por ejemplo, territorial

consiste en superponer varios temas, como pendiente media del terreno, tipo de vegetación, precipitación media, etc., por lo que se obtienen polígonos en los que estos factores son constantes; es decir, se ha dividido el territorio en áreas uniformes (unidad de diagnóstico). Si establecemos unos criterios de ordenación, por ejemplo, para repoblaciones, las distintas áreas obtenidas resultan por unión de las unidades de diagnóstico.

Para ello, debemos asignar a los polígonos un atributo que indique la nueva clase agregada y proceder a eliminar los arcos que a ambos lados los polígonos tienen el mismo valor en este campo de clase agregada. La Figura 3.3. muestra la técnica de eliminación de polígonos por la eliminación de un borde común.

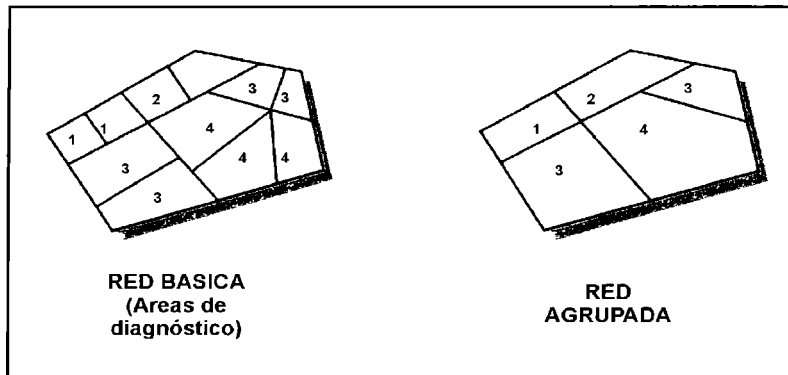


Figura 3.3 : Proceso de agrupamiento de polígonos

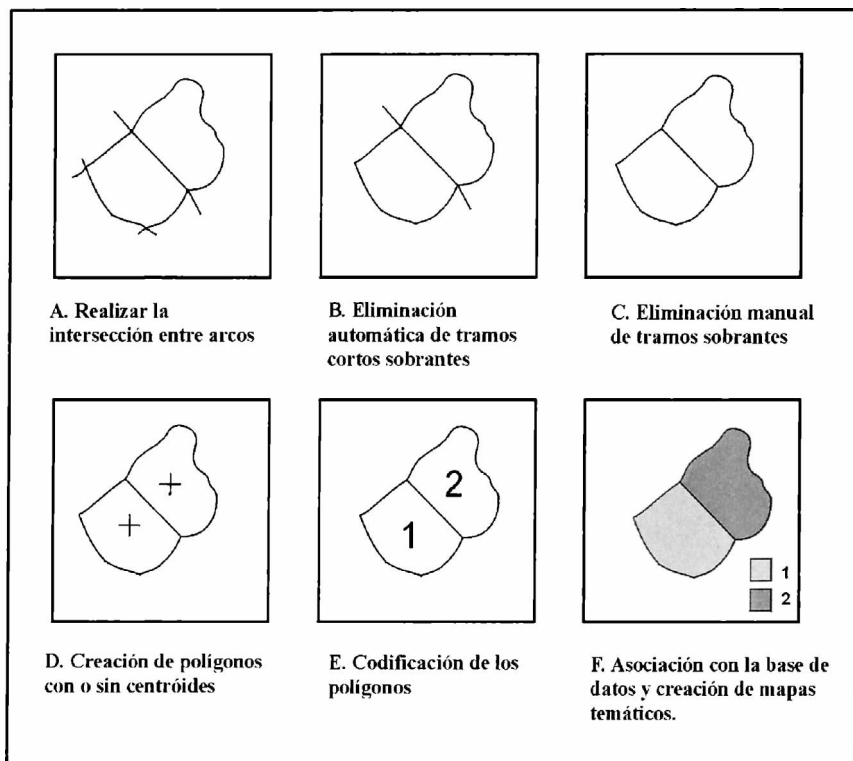
### 3.1.2 Funciones de Creación de Topologías

Se dice que un conjunto de entidades forma una topología cuando existe una o varias relaciones entre cada una y las que la rodean. Estas relaciones pueden ser, por ejemplo, la que guardan los arcos de los polígonos que forman una red, de tal forma que se garantice la continuidad de la misma.

Las funciones creadoras de topología son muy importantes para la organización del GIS, sobre todo en la fase de captura de datos.



Algunos sistemas GIS tienen funciones creadoras de topología muy potentes, y esto los caracteriza frente a otros sistemas que las tienen menos desarrolladas, o incluso que carecen de ellas, con lo que deberán ser apoyados en la captura de datos con estos sistemas creadores de topología.



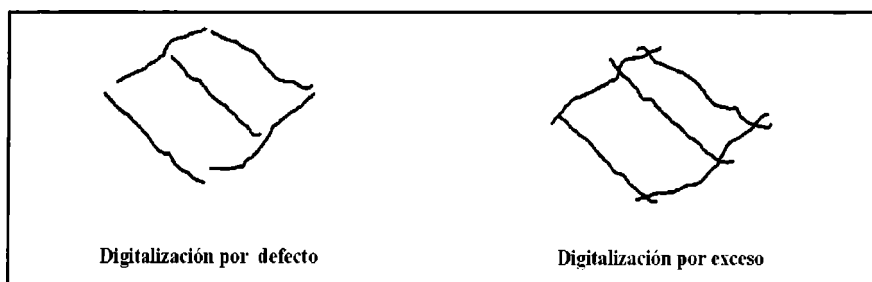
**Figura 3.4 :** Fases de corrección de errores y creación de la topología de polígonos

Las funciones de creación de topología suelen estar asociadas a las funciones de limpieza y corrección de errores geométricos.

En la figura 3.4 se recogen las fases de creación de la topología de polígonos. Una vez que todos los arcos están cortados entre sí, se procede a la creación de nodos en todos los extremos de los arcos; para que los arcos que confluyen en un punto tomen un mismo nodo, deberá especificarse un parámetro de tolerancia o umbral, de tal manera que si la distancia entre dos nodos es menor que este parámetro se unan los nodos en uno solo. En algunos sistemas, este parámetro se utiliza en el proceso de creación de topología para

realizar un filtrado de líneas, de manera que se elimine igualmente un vértice si está a menos distancia que este parámetro de otro vértice anterior. Sin embargo es preferible que ambos procesos se realicen por separado.

En este proceso de limpieza y generación de topología se puede establecer otro parámetro de tolerancia que permite eliminar los excedentes de un arco en una intersección. Este parámetro no afecta al filtrado de líneas, por lo que puede ser mayor que el anterior. Esta es la razón por la que en la digitalización vectorial deberá siempre procurarse que los arcos se crucen entre sí; es decir, es mejor digitalizar por exceso que por defecto, ya que los arcos que no llegan deberán ser corregidos a mano. En la Figura 3.5 se muestra este detalle de digitalización.



**Figura 3.5 :** En la digitalización manual, es preferible la digitalización por exceso que la digitalización por defecto

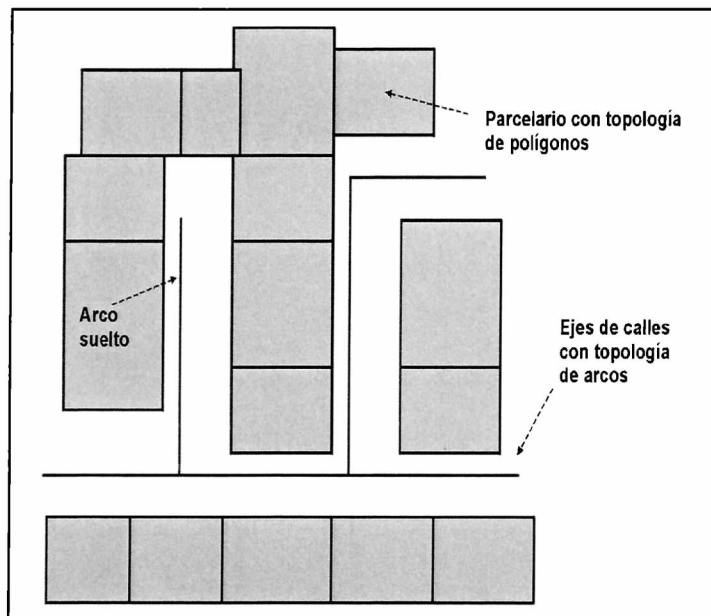
Si en el proceso de limpieza queda algún arco sin estar conectado a otro, se deberá proceder a su corrección, por ejemplo, aumentando los parámetros de ajuste; si la deformación que se produce no es aceptable, se deberá corregir manualmente cada uno de los casos, bien añadiendo un nuevo arco o quitando el que estaba suelto. Estos procesos deberán ser ensayados previamente y volcar el resultado sobre otro archivo, de forma que si la deformación que se ha producido no es aceptable, deberán reducirse los parámetros de ajuste y procesar nuevamente el archivo original.

Una vez que todos los arcos están conectados por ambos extremos, es posible proceder a la creación de la topología, para lo cual suele haber otra función que, una vez

comprobado que no hay errores de intersección y arcos sueltos, proceda a la clasificación de arcos y nodos.

Una vez que el proceso de creación de topología se ha completado con éxito, se crea un registro de base de datos para cada polígono y en la mayoría de los sistemas se crea una entidad puntual situada en el interior del polígono (centroide o etiqueta), que es la que permite el acceso a la base de datos. En los sistemas más evolucionados, tales como los orientados a objetos, este centroide es opcional, y se utiliza cualquier coordenada interior del polígono para acceder a la base de datos.

En la creación de topología de arcos, el proceso es similar, salvo que se permiten arcos sueltos por un extremo, por ejemplo, para representar una calle sin salida.



**Figura 3.6:** Una topología de arcos permite arcos sueltos, por ejemplo una calle sin salida

La creación de topología de puntos se limita a eliminar algún punto que esté muy próximo a otro anterior y a crear los registros de base de datos para cada punto.

# **Capítulo IV**

## **Implementación**

## 4.1 Descripción del desarrollo

Para poder realizar la aplicación se estudiaron previamente los diferentes métodos de digitalización y los posibles errores humanos que se cometen en el mismo. Luego se analizaron los diferentes tipos de formatos de almacenamiento de datos de los sistemas GIS y se eligió el formato shapefile de ESRI por ser el más ampliamente difundido y el más utilizado. El paso siguiente consistió en encontrar un modelo de datos que nos permita representar el shapefile y que facilite el manejo de las operaciones de corrección.

El modelo de datos a elegir debía tomar en cuenta los siguientes puntos:

- *Representar exactamente el contenido del shapefile.*
- *Facilitar el manejo del historial de aplicaciones.*
- *Mantener un registro de las operaciones efectuadas.*
- *Simplificar la visualización en pantalla.*
- *Posibilitar la realización de algoritmos matemáticos sobre el mismo.*
- *Utilizar metodología de análisis y diseño orientados a objetos.*

Se comenzó con el módulo de lectura e interpretación del shapefile y se buscó la mejor forma de almacenar en memoria esta información. El primer inconveniente que encontramos es que el formato shapefile contiene datos en representación big endian y little endian. Estas representaciones se refieren a cuáles son los bytes más significativos en un tipo de datos multibyte. En una arquitectura big-endian, los bytes de más a la izquierda (aquellos con la dirección de memoria más baja) son los más significativos. En una arquitectura little-endian los bytes de más a la derecha son los más significativos. Por ejemplo, consideremos el número  $1025 (2^{10} + 1)$  almacenado en un entero de 4 bytes.

00000000 00000000 00000100 00000001

<b>Dirección</b>	<b>Representación en Big Endian de 1025</b>	<b>Representación en Little Endian de 1025</b>
00	00000000	00000001
01	00000000	00000100
02	00000100	00000000
03	00000001	00000000

Muchos mainframes, particularmente IBM, usan arquitectura big endian. La mayoría de las computadoras modernas, incluyendo las PCs, usan el sistema little endian. El PowePC soporta bi-endian ya que entiende ambos sistemas.

El ejemplo de arriba muestra solo big y little endian a nivel de bytes. Pero también se puede usar a nivel de bits.

Continuando con la descripción del desarrollo, había que tomar dos caminos al momento de la elección del modelo de datos: uno que refleje exactamente el formato del shapefile manteniendo a lo largo de los procesos la estructura de la polilínea y sus partes, y otra que mantuviese los datos originales pero sin respetar la estructura original al 100%. Si optábamos por la primera elección encontrábamos que no podíamos mantener la estructura original del shapefile luego de realizar las operaciones de corrección, por lo cual no tenía sentido utilizar un modelo de datos demasiado complicado si no se iba a poder mantener la estructura a lo largo de todos los procesos de corrección.

El modelo que se eligió es simple: se optó por almacenar cada arco de cada parte de cada polilínea como un arco independiente, perdiendo la relación original que mantenía este arco dentro del shapefile de entrada. Si bien se puede pensar que se está perdiendo la estructura original, esto no es cierto porque hay que recordar que el proceso de depuración de errores se hace antes del proceso de asociación de información al archivo.

El modelo de datos elegido, y el cual nos pareció que podía facilitar todos los procesos de depuración es el siguiente:

Se eligió mapear el shapefile de entrada a una estructura de grafo. Cada línea de cada parte de cada polilínea es una arista del grafo, y los puntos extremos de cada línea involucrada son los nodos del grafo. Los nodos mantienen las coordenadas geográficas,

mientras que los arcos solamente contienen referencia a sus nodos extremos (nodo inicial y final).

Aunque esto se vea como una representación muy simplificada de la estructura original, tiene muchas ventajas, como por ejemplo:

- *Se facilitan mucho los procesos de eliminación de arcos y nodos, solo basta con eliminarlos de sus respectivos conjuntos.*
- *Cualquier operación que involucre el movimiento de un nodo extiende o acorta automáticamente la longitud de cualquier arco que contenga a dicho nodo.*
- *El proceso de visualización se simplifica mucho porque consiste solo en dibujar el grafo.*
- *El modelo de datos no se pierde a lo largo de cada aplicación de operaciones, debido a que se toma como entrada un grafo, se aplica la operación, y se obtiene otro grafo como resultado.*

Otro problema que nos encontramos fue la transformación de las coordenadas físicas (ver Apéndice C) almacenadas en el archivo (formato tipo flotante) a las coordenadas de pantalla que son de tipo entero (píxeles). Si bien a simple vista parece algo trivial, no lo es, porque al tener coordenadas fuente de tipo real se tiene mayor precisión que lo que se tiene con álgebra entera, por lo cual se requieren una serie de transformaciones y scaling para que la pérdida de precisión en la conversión de real a entero sea la menor posible.

## 4.2 Implementación del Modelo de Datos

A la hora de plasmar el modelo de datos en una implementación con un lenguaje de programación, se optó por utilizar la metodología orientada a objetos, dado que ésta nos permite una mejor representación de los datos y las operaciones efectuadas sobre los mismos.

El lenguaje de programación elegido, luego de ensayos preliminares en Java, fue C++ dado que el tipo de equipamiento que poseíamos no era el adecuado para ejecutar Java de manera performante.

Veamos un poco más en detalle como fue implementada la estructura de datos, describiendo las clases principales.

**Clase LINK:**

Representa la información de un arco en un grafo. Solamente contiene referencias a los nodos de la clase NODO que representan sus puntos extremos.

**Clase NODO:**

Un objeto de esta clase representa un nodo en el grafo. Contiene tanto coordenadas geográficas como lógicas y provee operaciones para obtener distancia entre nodos, etc.

**Clase GRAFO:**

Un grafo consiste esencialmente de dos conjuntos: uno de la clase LINK y otro de la clase NODO. Además provee todas las operaciones posibles de corrección de errores

**Clase SHAPEFILE:**

Un SHAPEFILE contiene un conjunto de objetos GRAFO, cada uno representando una instancia en el proceso de corrección. La primera instancia corresponde a la modelización del shapefile original y las subsecuentes a la aplicación de los diferentes procesos de corrección.

**Clase POLYLINE:**

Solamente se utiliza en el momento de la carga inicial del shapefile y provee operaciones para subdividir una polilínea y sus partes en el conjunto de líneas individuales que la constituyen.



## **4.3 Descripción Funcional de las operaciones**

Luego de la descripción de las clases vamos a detallar las operaciones que se utilizaron para la corrección automática y semiautomática de errores.

### **4.3.1 Carga del Shapefile en Memoria**

La carga desde el archivo shapefile es la primer operación a realizar en el proceso de corrección de errores. El formato de archivo utilizado es el formato shapefile de ESRI de polilíneas (tipo 3) (Apéndice B). Como primer paso se lee la cabecera (header) del cual se obtiene la versión, la longitud total (file length), el bounding box, el tipo de shapefile y el código de archivo (file code). Luego se realiza un proceso de iteración para leer cada una de las polilíneas que constituyen el archivo. Cada polilínea consta de un record header y de un cuerpo. Cada uno de los puntos que se leen de la polilínea se transforma en un nodo del grafo inicial y a partir de la información de la estructura de las partes se construye el conjunto de arcos del grafo. Aquí se puede observar la pérdida de estructura a la que se hizo referencia anteriormente. A pesar de ser un proceso absolutamente secuencial, gracias a la metodología orientada a objetos utilizada en la implementación, y a las ventajas del polimorfismo, obtuvimos un código sencillo y muy legible.

Resumiendo, el proceso de carga del shapefile en memoria consiste en tomar como entrada un archivo con las características mencionadas anteriormente, y construir un grafo inicial que representa la estructura del archivo y que sirve como punto de partida para la modificación y las sucesivas operaciones de detección y consiguiente corrección.

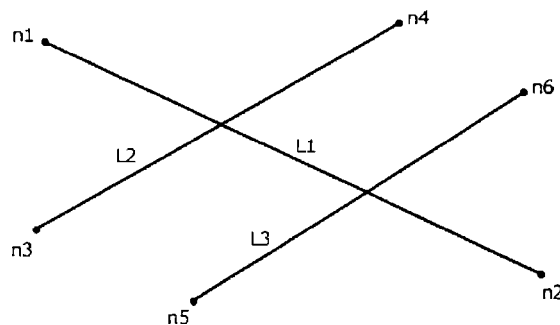
## 4.3.2 Descripción de las operaciones

### 4.3.2.1 Nodificación

Es la primera operación a aplicar y necesaria para poder realizar las operaciones subsiguientes. El proceso de nodificación consiste en encontrar todas las intersecciones entre aristas del grafo. Por cada intersección se genera un nuevo nodo y a su vez las aristas involucradas se reemplazan por las nuevas aristas que surgen de la intersección. El objetivo fundamental de este proceso es la obtención de un grafo cuyas aristas no se intercepten entre sí.

Para entender mejor el proceso utilizaremos un ejemplo simple pero ilustrativo.

Partimos de la siguiente situación:



**Figura 4.1:** Arcos antes de aplicar la nodificación

En la figura anterior vemos un caso previo a la aplicación del proceso de nodificación. El gráfico está compuesto por un conjunto de nodos  $N = \{n1, n2, n3, n4, n5, n6\}$  y un conjunto de aristas  $A = \{a1, a2, a3\}$ , donde:

$$a1 = (n1, n2)$$

$$a2 = (n3, n4)$$

$$a3 = (n5, n6)$$

Para encontrar todas las intersecciones de los elementos del conjunto A, se utiliza la ecuación de la recta entre todos los pares de aristas de A. En el ejemplo sería: **a1** con **a2**, **a1** con **a3** y **a2** con **a3**

Generalizando, para un conjunto A de cardinalidad N, la cantidad total de intersecciones a realizar es:

$$C(N,2) = N! / ((2!) * (N-2)!) \quad (\text{Combinación de } N \text{ tomados de } 2)$$

Como vemos, este proceso es simple pero para un conjunto cuya cardinalidad es muy elevada, el número de cálculos a realizar aumenta considerablemente; sin embargo, existen formas de optimizar este proceso que comentaremos más adelante.

Para llevar un registro de la cantidad de partes en que se divide una arista original asociamos una lista con cada arista, donde iremos almacenando la información de los nodos que se generen producto de la intersección de esta arista con todas las otras que la intercepten. El criterio de ordenación de los nodos en esta lista se define de la siguiente manera:

*Sea  $L(a)$  la lista asociada a la arista  $a$ , entonces*

*$L(a) = \{n_1, n_2, \dots, n_k\}$  está ordenada si para todo  $i, j$  tales que  $1 \leq i < j \leq k$ ,  $1 \leq i < j \leq k$ ,  $i < j$  se verifica que  $d(n_i, n_1) \leq d(n_j, n_1)$ , siendo  $d(n_i, n_j)$  la distancia entre los nodos.*

Inicialmente la lista contiene solo los nodos extremos. Para nuestro caso esto sería:

$$L(a_1) = [n_1, n_2]$$

$$L(a_2) = [n_3, n_4]$$

$$L(a_3) = [n_5, n_6]$$

La intersección de  $a_1$  con  $a_2$  generara el nuevo nodo  $i_1$ . De esta manera el conjunto de nodos N pasa a ser  $\{n_1, n_2, n_3, n_4, n_5, n_6, i_1\}$ .

Debido a que el nodo  $i_1$  se genera por la intersección de  $a_1$  y  $a_2$ , y por lo descrito anteriormente, debemos agregar  $i_1$  a la lista de nodos de ambas aristas.

Tomando en cuenta el criterio de ordenación especificado anteriormente, las respectivas listas quedan:

$$L(a_1) = [n_1, i_1, n_2]$$

$$L(a_2) = [n_3, i_1, n_4]$$

$$L(a_3) = [n_5, n_6]$$

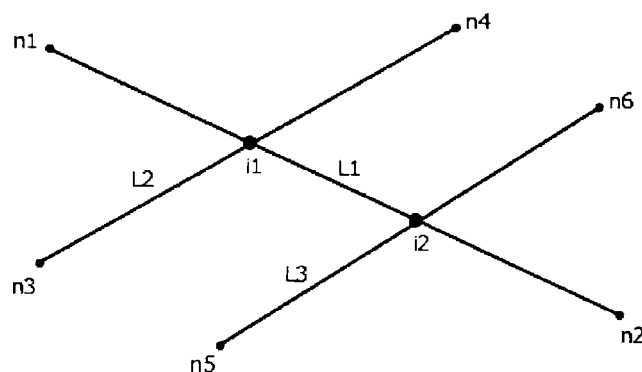
De igual manera la intersección de  $a_1$  con  $a_3$  genera el nodo  $i_2$ , por lo cual el nuevo conjunto de nodos es  $N = \{n_1, n_2, n_3, n_4, n_5, n_6, i_1, i_2\}$  y las listas quedan;

$$L(a_1) = [n_1, i_1, i_2, n_2]$$

$$L(a_2) = [n_3, i_1, n_4]$$

$$L(a_3) = [n_5, i_2, n_6]$$

$a_2$  no se intercepta con  $a_3$ , por lo tanto no se generan nuevos nodos.



**Figura 4.2:** Arcos modificados

Una vez finalizado el proceso de intersección y armado de listas de los nodos asociados a cada arista original nos encontramos con la siguiente situación:

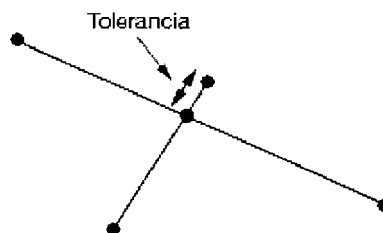
Podemos apreciar en la ventana de la derecha de la Figura 4.3 como aumenta considerablemente el número de nodos y arcos luego de aplicada la función “Nodify”.

#### 4.3.2.2 Eliminación de arcos demasiado pequeños (Drop Short Links).

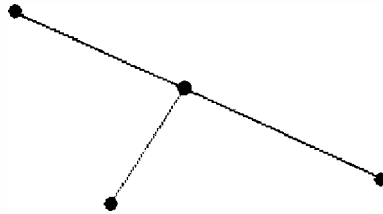
Este proceso consiste en la eliminación de aquellos arcos del grafo cuya longitud es menor o igual a un parámetro de tolerancia ingresado por el usuario.

Al seleccionar esta opción se abre una caja de diálogo donde se ingresa el factor de tolerancia (umbral) especificado en unidades de medida físicas. Una vez ingresado el parámetro se puede realizar una vista preliminar en un color diferente de cuales son los links que serán eliminados. Si se confirma la operación dichos arcos serán efectivamente borrados. De lo contrario se pueden ir probando diferentes umbrales de tolerancia hasta obtener el resultado deseado.

La implementación de este proceso consiste en iterar sobre el conjunto de arcos del grafo e ir calculando la longitud de los mismos. Aquellos arcos cuya longitud sea menor o igual al umbral de tolerancia se marcan como arcos “dirty”, de manera tal de poder identificarlos con otro color cuando se efectúa la vista preliminar. Esta misma marca sirve para, en caso de confirmación de la operación, saber cuáles son los nodos a eliminar.

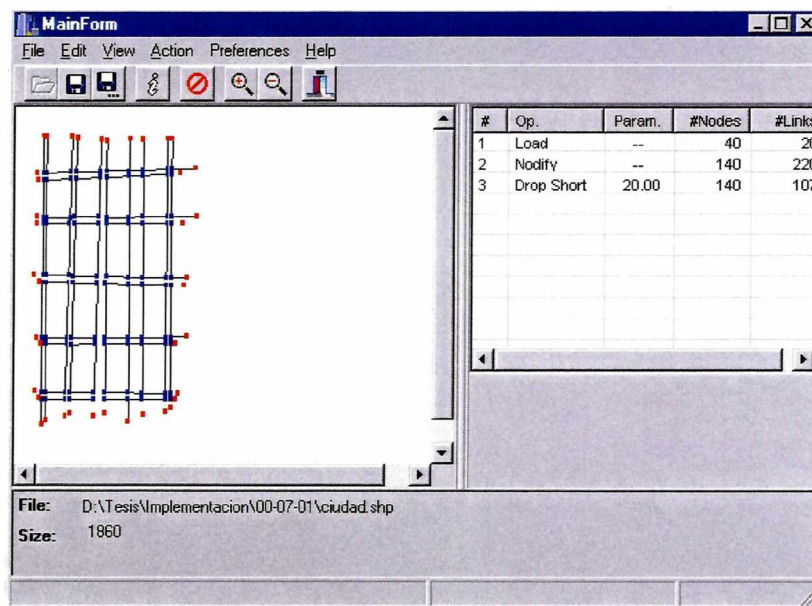


**Figura 4.4:** Arcos antes de aplicar la operación. Se nota que la tolerancia es levemente mayor al segmento a cortar.



**Figura 4.5:** Arcos después de aplicar la operación

En la siguiente figura vemos el resultado de aplicar la función de borrado de arcos pequeños con un parámetro de tolerancia de 20. Se puede observar la aparición de nodos huérfanos (Dangling Nodes), los mismos podrán ser eliminados mediante la aplicación de la función “Delete Dangling Nodes”.



**Figura 4.6:** Ejemplo de aplicación de la función de eliminación de arcos pequeños

#### 4.3.2.3 Borrado de nodos que no tienen ningún arco (Delete Dangling Nodes)

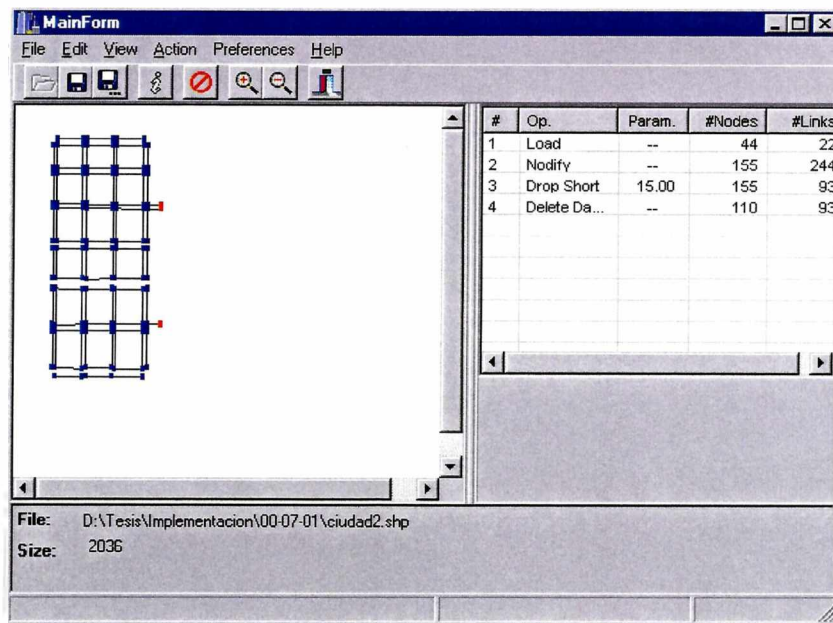
Una situación común que se presenta luego de la aplicación de una operación de eliminación de arcos es la aparición de nodos “huérfanos”. Estos nodos que eran puntos

extremos de los arcos eliminados, y luego del borrado de dichos arcos, quedaron sueltos, es decir, no pertenecen a ningún arco.

En nuestra implementación los nodos producto de las intersecciones (modificación) se visualizan en un color y los nodos generados en el proceso de carga del shapefile en otro. Debido a que en un objeto de la clase nodo no estamos guardando una referencia de a cuales arcos pertenece este nodo, resolvimos esta situación de la siguiente manera:

*Sea  $N$  el conjunto de todos los nodos del grafo y sea  $U$  el conjunto de todos aquellos nodos que son puntos extremos de al menos un arco ( $U \subseteq N$ ), se verifica que el conjunto  $D = N - U$  ( $D$  es el complemento de  $U$  con respecto a  $N$ ) es el conjunto de los nodos que no pertenecen a ningún arco.*

*Para obtener  $U$  iteramos sobre el conjunto de arcos del grafo, y para cada arco insertamos sus puntos extremos en  $U$ . Luego realizamos la resta de conjuntos y obtenemos el conjunto  $D$  que es el conjunto de los nodos a eliminar.*



**Figura 4.7:** Aplicación de la función “Delete Dangling Nodes”

En la figura anterior podemos observar como se reduce la cantidad de nodos (se borran los nodos sueltos) después de aplicar la función “Delete Dangling Nodes”. Es muy común usar esta función inmediatamente después de haber usado “Drop Short Links”.

#### 4.3.2.4 Unión de nodos próximos (Node Clustering)

La implementación de este proceso se dividió en dos partes, debido a la necesidad de intervención por parte del usuario para seleccionar cuales clusters (grupos de nodos próximos) unificar y cuales no, por lo que esta operación se considera semiautomática.

La primera etapa del proceso consiste en la determinación de todos los posibles clusters del grafo, y la segunda parte requiere de intervención manual para determinar cuales clusters se unifican y cuáles no.

Para determinar todos los clusters del grafo se decidió incluir un atributo al nodo, el cual identifica cuál es el estado del nodo relativo al clustering. Los valores posibles de este atributo son:

- **FREE:** Indica que el nodo no pertenece a ningún cluster.
- **HEAD:** El nodo es el centro del cluster (esto significa que ante la realización de esta operación sobre el cluster, todos los nodos restantes del cluster se unificarán en este).
- **SNAP:** Forma parte de un cluster pero no es head (será unificado en el head del cluster al cual pertenece).

El algoritmo de determinación de clusters se podría resumir básicamente en los siguientes pasos:

1. Marcar todos los nodos del grafo como *FREE*.
2. Tomar siguiente nodo  $n$  del grafo en estado *FREE*. Si no hay más pasar a 8.
3. Armar el conjunto  $C_n = \{n\}$ .



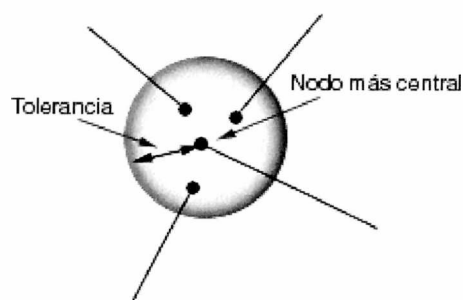
4. Agregar al conjunto  $C_n$  todos aquellos nodos  $m_i \diamond n$  en estado FREE tales que la distancia entre  $n$  y  $m_i$  sea menor o igual al umbral de tolerancia  $t$ .
5. Si el conjunto  $C_n = \{n\}$  entonces retornar al punto 2. (No hay ningún nodo  $m_i$  próximo a  $n$  dentro del umbral).
6. Marcar al nodo  $n$  como HEAD y a todos los nodos restantes de  $C_n$  como SNAP.
7. Retornar al punto 2.
8. Para cada conjunto  $C_n$  de cardinalidad mayor a 1 calcular el nodo central  $g_n$  (centroide) entre todos sus nodos.

El siguiente proceso consiste en iterar sobre los clusters, y unificar solo aquellos que el usuario seleccione.

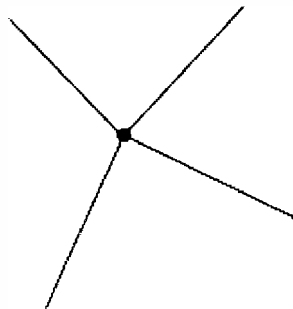
Sea  $C_n$  el cluster a unificar, el algoritmo puede resumirse de la siguiente manera:

1. Asignar al nodo  $n$  (HEAD de  $C_n$ ) las coordenadas de  $g_n$  (centroide de  $C_n$ ).
2. Tomar siguiente nodo  $m_i \diamond n$  (SNAP) del conjunto  $C_n$ . Si no hay más terminar.
3. Para cada arco del grafo que tenga como punto extremo a  $m_i$ , reemplazar  $m_i$  por  $n$ .
4. Eliminar  $m_i$  del conjunto de nodos del grafo.
5. Retornar al punto 2.

Como puede observarse, el nodo marcado como HEAD dentro de un cluster es el único nodo que sobrevive al proceso de unificación, dado que el resto de los nodos pertenecientes al cluster se unifican en éste y se borran del conjunto de nodos del grafo.

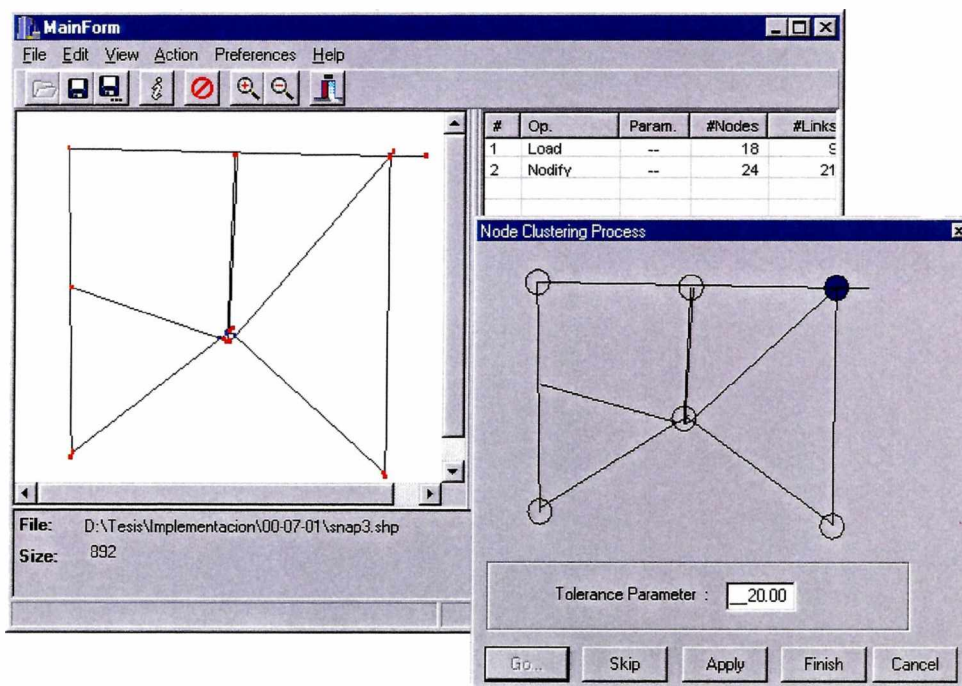


**Figura 4.8:** La circunferencia denota el entorno (cluster) de los nodos a unificar



**Figura 4.9:** Estos son los arcos después de aplicar snapping

En la figura 4.10 se observa como trabaja la operación de unión de nodos próximos.



**Figura 4.10:** Proceso para corregir nodos próximos

La aplicación permite, mediante una vista preliminar, seleccionar aquellos clusters que el operador desea unificar. El operador tiene que ir iterando por todos los clusters (nodos próximos), generados de acuerdo al parámetro de tolerancia previamente ingresado, y presionando el botón “Apply” si desea unificar, “Skip” si desea pasar por alto el cluster (no hace nada) o presionando “Finish” para terminar con el proceso de corrección. Queda

claro que este proceso es semiautomático ya que depende de una persona que decide que hacer en cada paso de iteración.

#### 4.3.2.5 Arco con un extremo sin conectar a otro arco (*Extend Undershoots*)

Como en el caso del snapping de nodos próximos el proceso se divide en dos etapas, la primera etapa del proceso consiste en la determinación de todos los posibles arcos a extender del grafo, y la segunda parte requiere de intervención manual para determinar cuáles arcos se extenderán y cuáles no. Al igual que el snapping esta operación es semiautomática.

La primera etapa del proceso consiste en la determinación de todos los posibles arcos del grafo, y la segunda parte requiere de intervención manual para determinar cuáles arcos se extenderán.

Para determinar todos los arcos a extender del grafo se decidió incluir un atributo al arco, el cuál identifica cual es el estado del arco relativo a si es candidato a extender o no, y se lo marca como *LINK\_TO\_EXTEND* marcando al arco como un potencial link a extender.

El algoritmo de determinación de arcos a extender se podría resumir básicamente en los siguientes pasos:

1. Tomar el siguiente arco  $l$  del grafo. Si no hay más arcos pasar al punto 3.

2. Para cada arco  $l' \in l$  hacer'

**Si  $\text{distancia}(l', l, n) \leq t$  (tolerancia) entonces**

*Guardo el nodo  $n$  (nodo extremo de  $l'$  más cercano al punto de intersección entre  $l$  y  $l'$ ) en  $C_n$  (Conjunto de nodos de arcos a extender)*

3. Retornar al punto 1

El siguiente proceso consiste en iterar sobre los nodos de arcos desde los cuales se puede extender, y extender solo aquellos que el usuario seleccione

Sea  $n$  el nodo extremo de un arco  $l$  a extender, el algoritmo puede resumirse de la siguiente manera:

1. **Si** el nodo extremo a extender es nodo inicial **entonces**

*Cambiar el valor del nodo inicial por el punto de intersección entre  $l$  y  $l'$*

**sino** *Cambiar el valor del nodo final por el punto de intersección entre  $l$  y  $l'$*

2. **Guardar** los nuevos links  $l_1$  y  $l_2$ , generados producto de la intersección entre  $l$  y  $l'$ , en el conjunto de arcos del grafo.

3. **Borrar** el arco destino  $l'$  de la lista de arcos del grafo.

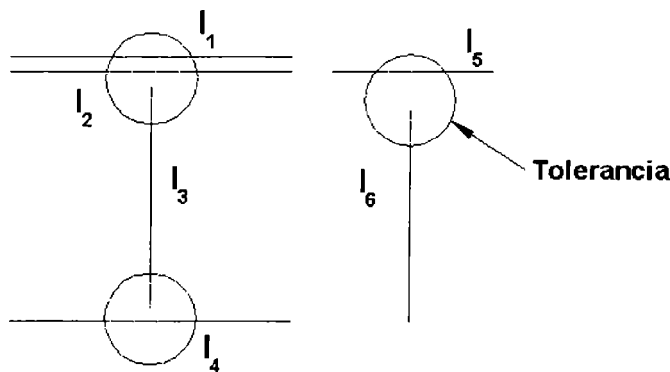


Figura 4.11: Arcos antes de aplicar la operación

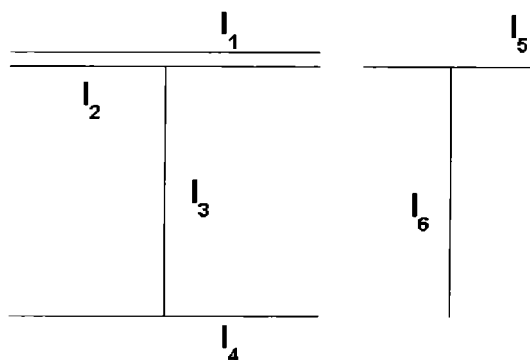


Figura 4.12: Arcos después de aplicar la operación

En la Figura 4.13 vemos como trabaja nuestra aplicación, para corregir, cuando existen arcos con un extremo sin conectar a otros arcos. Puede observarse la similitud en el modo de operación con la función de unión de nodos próximos, donde el operador decide

cuales arcos quiere extender según el umbral de tolerancia ingresado. La operatoria para la corrección es la misma que en “Node Clustering”

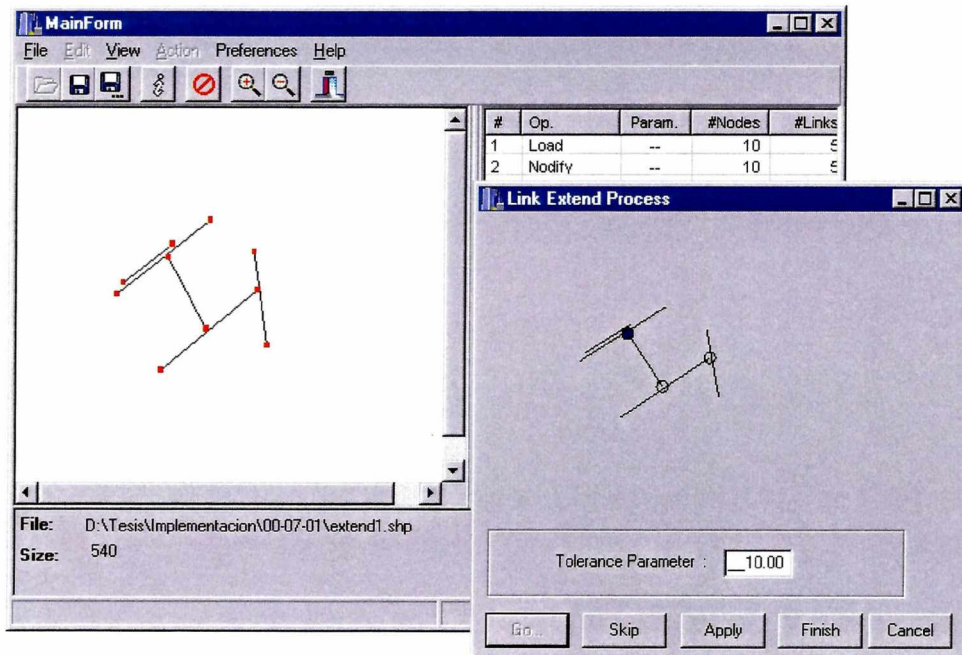


Figura 4.13: Aplicación de la función “Extended Undershoot”

#### 4.3.2.6 Eliminación de arcos duplicados (Delete duplicated links)

El proceso de eliminación de arcos es 100% automático, es decir no tiene interacción por parte del usuario, y se basa en eliminar las posibles superposiciones en donde un arco “tapa” a otro. Esta anomalía se debe a posibles errores en la digitalización o en el proceso de carga en el shapefile.

Para determinar la duplicación primero se tiene que probar que los arcos o vectores sean colineales para lo cual se procede de la siguiente manera:

Sean  $l_1$  y  $l_2$  los dos links, donde  $l_1$  tiene como puntos extremos  $n_1, n_2$ ; y  $l_2$  tiene como puntos extremos  $n_3, n_4$  entonces decimos que  $l_1$  y  $l_2$  son colineales sii

$$\frac{(n_2 - n_1) \times (n_4 - n_3)}{\| (n_2 - n_1) \times (n_4 - n_3) \|} = 0 \quad \text{y} \quad \frac{(n_3 - n_1) \times (n_2 - n_1)}{\| (n_3 - n_1) \times (n_2 - n_1) \|} = 0$$

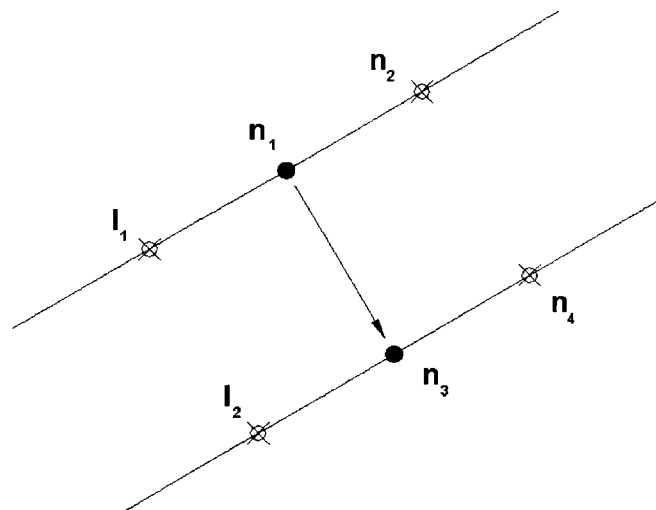
es decir el módulo del producto vectorial entre las diferencias entre los puntos  $n_2$  y  $n_1$  y la diferencia entre los puntos  $n_4$  y  $n_3$  debe ser 0 y el modulo del producto vectorial entre las diferencias entre los puntos  $n_3$  y  $n_1$  y la diferencia entre los puntos  $n_2$  y  $n_1$  debe también ser igual a 0

Una vez que probamos que los segmentos son colineales nos resta comprobar que estén superpuestos, para lo cual probamos que se cumpla la siguiente desigualdad:

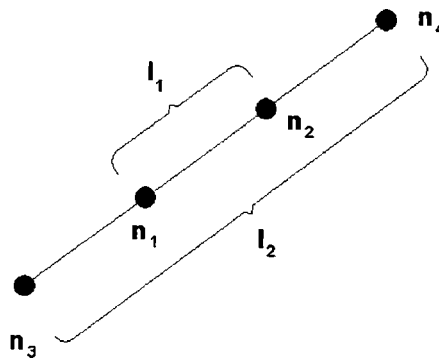
$$(n_4 - n_2) \bullet (n_3 - n_1) \leq 0$$

es decir el producto escalar de las diferencia entre los puntos  $n_4$  y  $n_2$  y la diferencia entre los puntos  $n_3$  y  $n_1$  debe ser menor que 0.

Con esta ultima prueba podemos afirmar que los dos segmentos  $l_1$  y  $l_2$  se superponen, es decir un segmento esta incluido en el otro



**Figura 4.14:** Arcos paralelos pero no superpuestos



**Figura 4.15:** Arcos superpuestos

En la Figura 4.14 se muestra dos links paralelos pero que no se superponen. Para comprobar esto basta con probar que  $(n_3 - n_1) \times (n_2 - n_1) / \neq 0$ , es decir que  $n_3 \notin l_1$

En la Figura 4.15 se observa la superposición de  $l_2$  sobre  $l_1$ . En este caso el algoritmo eliminará el link  $l_1$ .

El algoritmo de determinación de arcos incluidos en otros se podría resumir básicamente en los siguientes pasos:

1. Tomar el siguiente arco  $l$  del grafo. Si no hay más arcos para procesar pasar al punto 4
2. Para cada arco  $l' \supseteq l$  hacer'
  - Si  $l \subseteq l'$  entonces Guardar el arco  $l$  en  $C_d$  (Set de Arcos a eliminar). Pasar al pto 3.
3. Retornar al punto 1
4. Eliminar todos los arcos del conjunto  $C_d$

# **Conclusión**



Durante el transcurso del trabajo logramos comprender la utilidad real de los sistemas GIS y por qué cada vez es más frecuente su utilización.

Para que un sistema GIS sea realmente potente es fundamental que la información geográfica sea lo más precisa posible. Debido a que esta información se suministra a partir de procesos de captura de datos que en la mayoría de los casos son manuales, y por lo tanto inherentemente imprecisos, es de suma importancia contar con un conjunto adecuado de funciones de detección y corrección de errores que permitan una depuración.

Los sistemas más sofisticados cuentan con este tipo de funciones, pero los sistemas más económicos y de más fácil acceso en su mayoría carecen o cuentan con un conjunto muy limitado de herramientas de corrección.

El hecho de que los sistemas menos sofisticados no cuenten con dichas herramientas no es casual, ya que a lo largo del desarrollo del trabajo observamos que estas funciones tienen un costo computacional muy alto.

Para nuestro trabajo tomamos como punto de partida un sistema GIS sin ningún tipo de mecanismo de detección y corrección de errores. A partir de ello estudiamos los errores más frecuentes y propusimos soluciones a los mismos.

Un punto a considerar es que los datos geográficos no están representados mediante una base de datos (relacional u orientada a objetos), y por lo tanto carecen de cualquier herramienta standard para su fácil manipulación (como podría ser SQL, OQL, etc.). Por el contrario, estos datos están representados mediante archivos con formato propietario, por lo cual debimos crear un framework para poder trabajarlos y aplicar las funciones antes mencionadas.

Optamos por la metodología orientada a objetos ya que ésta nos brindaba mayor flexibilidad que otros paradigmas. En el momento de decidir el lenguaje a utilizar para la implementación elegimos C++ por ser orientado a objetos, con buena performance y, debido a que necesitábamos utilizar mayormente elementos gráficos, este lenguaje nos permitía gran flexibilidad para acceder a las librerías gráficas del sistema operativo Windows NT.

Es importante aclarar que nuestra atención se enfocó en buscar soluciones, aún cuando las mismas no fueran las óptimas.

Este trabajo puede considerarse como una presentación inicial al tema de detección y corrección de errores posterior a la captura de datos en los sistemas GIS.

Como en toda presentación preliminar a un tema, se puso más énfasis en detallar las causas y posibles soluciones de los errores cometidos que en buscar los algoritmos óptimos y las modelizaciones más eficientes.

A continuación detallamos una serie de sugerencias para futuros trabajos de grado que profundicen el tema:

- *Utilizar un método de modificación inteligente. Podría dividirse el área de modificación en regiones y analizar las intersecciones por región, con lo cual se acotarían muchísimo los tiempos ya que sería mucho más simple detectar líneas que no se intersectan.*
- *Diseñar una aplicación que utilice concurrencia para el análisis de procesos con gran carga computacional.*
- *Estudiar como mejoraría la performance la división en zonas para todos los métodos de correcciones de error vistos en este trabajo.*
- *Estudiar posibles modelos de datos alternativos que faciliten la representación de un shapefile.*
- *Analizar la posibilidad de mapear el contenido de un shapefile a una base de datos relacional u orientada a objetos y de esta manera estandarizar los procesos de consulta y análisis geográfico.*

El proceso de análisis de detección y corrección de errores en los sistemas GIS no es simple. Hay muchos tipos diferentes de errores y cada uno merece ser estudiado en profundidad. Es nuestro deseo que este trabajo pueda considerarse como un punto de partida para estudios más avanzados sobre el tema. Esperamos que sea de utilidad para aquellas personas que se interesen en el tema.

# **Apéndice A**

## **Preliminares Matemáticos**

## A.1 Concepto de línea.

Una línea es intrínsecamente un objeto en el espacio unidimensional, pero pueden aparecer en espacios de dimensiones mayores. Una línea infinita que pasa a través de los puntos **A** y **B** se denota como **AB**. Una línea se denomina a veces *línea directa* o *línea recta* para enfatizar el hecho de que no tiene curvas a lo largo de su longitud.

Consideremos las líneas en el plano (espacio bidimensional). La línea con *x-intercept* **a** y con *y-intercept* **b** se puede expresar en forma de intersección como:

$$\frac{x}{a} + \frac{y}{b} = 1.$$

La línea que pasa a través de  $(x_1, y_1)$  cuya pendiente es **m** se puede expresar en formato punto-pendiente como:

$$y - y_1 = m(x - x_1).$$

La línea con *y-intercept* **b** y pendiente es **m** se puede expresar en formato pendiente-intersección como:

$$y = mx + b.$$

La línea que pasa a través de los puntos  $(x_1, y_1)$  y  $(x_2, y_2)$  se puede expresar en formato de dos puntos como:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1).$$

Otras formas de representación de líneas en el plano son:

$$a(x - x_1) + b(y - y_1) = 0$$

$$ax + by + c = 0$$

$$\begin{vmatrix} x & y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = 0.$$

Una línea en el plano también puede representarse como un vector. El vector a lo largo de la línea

$$ax + by = 0$$

esta dado por:

$$t \begin{bmatrix} -b \\ a \end{bmatrix},$$

con  $t \in \mathcal{R}$ . Similarmente, vectores de la forma

$$t \begin{bmatrix} a \\ b \end{bmatrix}$$

son perpendiculares a la línea. Tres puntos yacen sobre una línea si:

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = 0.$$

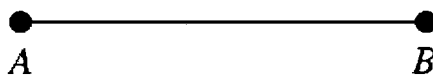
El ángulo entre las líneas

$$\begin{aligned} A_1x + B_1y + C_1 &= 0 \\ A_2x + B_2y + C_2 &= 0 \end{aligned}$$

es

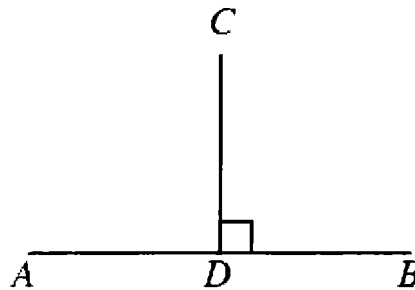
$$\tan \theta = \frac{A_1B_2 - A_2B_1}{A_1A_2 + B_1B_2}.$$

## A.2 Concepto de segmento de línea.



Se denomina segmento de línea a un intervalo cerrado correspondiente a una porción finita de una línea infinita. Los segmentos de línea generalmente son etiquetados en sus puntos extremos, digamos A y B, y entonces se escribe AB.

### A.3 Concepto de perpendicularidad.



Dos líneas, vectores, planos, etc. son perpendiculares si se intersectan formando un ángulo recto. En  $\mathcal{R}^n$  dos vectores **A** y **B** son perpendiculares si su producto escalar

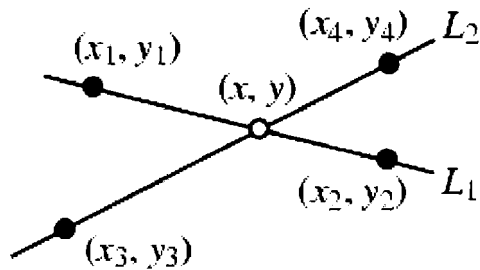
$$\mathbf{A} \cdot \mathbf{B} = 0.$$

En  $\mathcal{R}^2$ , una línea con pendiente  $m_2 = -1 / m_1$  es perpendicular a una línea con pendiente  $m_1$ . Los objetos perpendiculares son a veces denominados *ortogonales*.

En la figura de arriba, el segmento AB es perpendicular al segmento CD. Esta relación es comúnmente denotada por un cuadrado pequeño en el vértice donde los objetos perpendiculares se intersectan, como se muestra arriba, y se denota como

$$AB \perp CD$$

### A.4 Definición de intersección de líneas.



La intersección de dos líneas \$L\_1\$ y \$L\_2\$ en el espacio bidimensional, con \$L\_1\$ conteniendo los puntos \$(x\_1, y\_1)\$ y \$(x\_2, y\_2)\$ y \$L\_2\$ conteniendo los puntos \$(x\_3, y\_3)\$ y \$(x\_4, y\_4)\$ está dada por:

$$x = \frac{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} \begin{vmatrix} x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{vmatrix} - \begin{vmatrix} x_1 & y_1 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \begin{vmatrix} x_2 & y_2 & 1 \\ x_4 & y_4 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix}}$$

$$y = \frac{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} \begin{vmatrix} x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{vmatrix} - \begin{vmatrix} x_1 & y_1 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \begin{vmatrix} x_2 & y_2 & 1 \\ x_4 & y_4 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix}}$$

# **Apéndice B**

## **Descripción del formato shapefile.**



## B.1 Por qué utilizar shapefiles?

Un shapefile almacena geometría no topología y atributos de información para los gráficos espaciales de un conjunto de datos. La geometría para un gráfico se almacena como un conjunto de coordenadas vectoriales.

Debido al hecho de que los shapefiles no tienen la carga de procesamiento de una estructura topológica tienen ventajas sobre otros formatos de datos, tales como visualización más rápida de los dibujos y habilidad de edición. Los shapefiles permiten manejar gráficos simples que se superponen o que no son contiguos.

Además requieren menos espacio en disco y son más fáciles de leer y escribir.

Los diferentes formatos de shapefile son puntos, líneas y áreas. Un gráfico de área se representa como un lazo cerrado (polígonos doblemente digitalizados). Los atributos se mantienen en un archivo tipo dBase. Cada registro de atributo tiene una relación uno a uno con el registro del gráfico asociado.

## B.2 Métodos de creación de shapefiles.

Existen cuatro métodos generales para crear shapefiles:

- **Exportación:** Utilizando productos como ARC/INFO, PC ARC/INFO, ArcView GIS o BusinessMap se pueden exportar cualquier fuente de datos al formato shapefile
- **Digitalización:** Utilizando las herramientas disponibles en ArcView GIS se pueden crear directamente.
- **Programación:** Utilizando Avenue (ArcView GIS), MapObjects o Simple Macro Language (SML) (PC ARC/INFO) se pueden crear shapefiles desde un programa.
- **Escritura directa:** Utilizando las especificaciones de shapefile que provee ESRI Inc. se puede escribir un programa que genere un archivo con el formato correcto.

## B.3 Descripción técnica.

Un shapefile consiste de: un archivo principal, un archivo índice y una tabla dBase. El archivo principal es un archivo de acceso directo y de longitud de registro variable en el cual cada registro describe una figura con una lista de sus vértices. En el archivo índice cada registro contiene el desplazamiento del correspondiente registro del archivo principal tomando como referencia el principio del archivo principal. La tabla dBase contiene los atributos de las figuras, con un registro por figura. La relación uno a uno entre los atributos y la figura se basa en el número de registro.

Los registro de atributo en el archivo dBase deben estar en el mismo orden que los registros en el archivo principal.

### B.3.1 Convención de nombres.

Todos los archivos adhieren al formato de nombres de DOS (8.3). El archivo principal, el índice y el dBase tienen el mismo prefijo. El sufijo para el archivo principal es *shp*, el sufijo para el archivo índice es *shx* y el sufijo para el archivo dBase es *dbf*.

### B.3.2 Tipos numéricos

Un shapefile almacena números enteros y en doble precisión.

- **Enteros:** enteros de 32 bit con signo (4 bytes).
- **Dobles:** IEEE nro en punto flotante con precisión doble con signo 64 bits (8 bytes).

### B.3.3 Organización del archivo principal

El archivo principal (*shp*) contiene un encabezado de archivo de longitud fija seguido por registros de longitud variable. Cada registro de longitud variable consta de un encabezado de registro (longitud fija) seguido por un contenido de registro (longitud variable).

Encabezado del Archivo	
Encabezado del Registro	Contenido del Registro
Encabezado del Registro	Contenido del Registro
Encabezado del Registro	Contenido del Registro
Encabezado del Registro	Contenido del Registro
...	
...	
Encabezado del Registro	Contenido del Registro

**Figura B.1.:** Organización del archivo principal.

### B.3.4 División en categorías de un shapefile

#### *Relativo a los datos.*

- *Contenido del registro del archivo principal.*
- *Descripción de los campos del encabezado del archivo principal (shape type, bounding box, etc.)*

#### *Relativo al manejo del archivo.*

- *Longitudes de registros y del archivo.*
- *Desplazamiento de los registros, etc.*

Los enteros y los enteros de doble precisión que representan los campos de descripción de los datos en el encabezado del archivo y el contenido de los registros en el archivo principal están en little endian (PC o Intel). Los enteros y números de punto flotante de doble precisión que constituyen el resto del archivo se representan en big endian (Sun o Motorola).

### B.3.5 Encabezado del archivo principal.

El encabezado del archivo principal tiene una longitud de 100 bytes. La Tabla 1 muestra los datos que contiene el mismo con sus respectivas posiciones, valores, tipos y byte order de cada uno. En esta tabla la posición es relativa al comienzo del archivo.

Posición	Campo	Valor	Tipo	Byte Order
Byte 0	Código de Archivo	9994	Integer	Big
Byte 4	No Usado	0	Integer	Big
Byte 8	No Usado	0	Integer	Big
Byte 12	No Usado	0	Integer	Big
Byte 16	No Usado	0	Integer	Big
Byte 20	No Usado	0	Integer	Big
Byte 24	Largo del Archivo	Largo del Archivo	Integer	Big
Byte 28	Versión	1000	Integer	Little
Byte 32	Tipo de Shape	Tipo de Shape	Integer	Little
Byte 36	Rectángulo delimitador	Xmin	Double	Little
Byte 44	Rectángulo delimitador	Ymin	Double	Little
Byte 52	Rectángulo delimitador	Xmax	Double	Little
Byte 60	Rectángulo delimitador	Ymax	Double	Little
Byte 68*	Rectángulo delimitador	Zmin	Double	Little
Byte 76*	Rectángulo delimitador	Zmax	Double	Little
Byte 84*	Rectángulo delimitador	Mnin	Double	Little
Byte 92*	Rectángulo delimitador	Mmax	Double	Little

\* No Usado, con valor 0.0

**TABLA 1.** Descripción del Encabezado del Archivo Principal

El valor para el campo que especifica la longitud del archivo (*file length*) es el largo total del archivo medido en palabras de 16-bits (incluyendo las 50 palabras de 16-bits que constituyen el encabezado).

Todas las figuras no nulas en un shapefile deben ser del mismo tipo. Los valores para los tipos de shapefiles se muestran en la siguiente figura

Valor	Tipo de shapefile
0	Shapefile nulo.
1	Punto.
3	Polilinea.
5	Polígono.

**Figura B.2.: Valores y tipos de shapefile**

Actualmente los shapefiles están restringidos a contener el mismo tipo de figura. En el futuro los shapefiles podrían permitir contener más de un tipo de figura.

El Rectángulo delimitador en el encabezado del archivo principal almacena la región que delimita las figuras en el archivo: el rectángulo mínimo ortogonal a los ejes X e Y que contiene todas las figuras. Si el shapefile es vacío (es decir que no tiene registros) los valores para Xmin, Ymin, Xmax e Ymax no se especifican.

### B.3.6 Encabezado de registro.

El encabezado para cada registro almacena el número de registro y el largo del mismo. Los encabezados de registro tienen un largo fijo de 8 bytes. La tabla 2 muestra los campos en el encabezado del registro con su posición de byte, valor, tipo y byte order. La posición es relativa al comienzo del registro.

Posición	Campo	Valor	Tipo	Byte Order
Byte 0	Numero de Registro	Numero de Registro	Integer	Big
Byte 4	Largo de Contenido	Largo de Contenido	Integer	Big

**TABLA 2.** Descripción de los encabezados de registro del archivo principal

Los números de registro comienzan en 1.

El largo del contenido para un registro es la cantidad total de bytes que ocupa el registro medida en palabras de 16 bytes. Cada registro por lo tanto contribuye (4 + largo del contenido) con palabras de 16 bytes hasta el largo total del archivo (byte 24 del encabezado del archivo principal).

### B.3.7 Contenido de los registros del archivo principal

El contenido de cada registro del shapefile consiste de un tipo de figura seguido por los datos geométricos para dicha figura. El largo del contenido del registro depende del número de partes y vértices en la figura. Para cada tipo de figura describimos primero la figura y luego su mapeo del contenido en almacenamiento físico.

En las tablas siguientes la posición es relativa al comienzo del contenido del registro.

#### B.3.7.1 Shapefiles de tipo nulo

El tipo de figura 0 indica una figura nula, con ningún dato geométrico para la figura. Cada tipo de figura (punto, polilínea, etc) soporta figuras nulas. Frecuentemente las figuras nulas se utilizan para reservar lugar. Se usan durante la creación del shapefile y son reemplazadas con datos geométricos a medida que se van creando.

Posición	Campo	Valor	Tipo	Numero	Byte Order
Byte 0	Tipo de Shape	0	Integer	1	Little

**TABLA 3.** Contenido del registro de un Shapefiles de tipo nulo

### B.3.7.2 Shapefiles de puntos

Un punto consiste de un par de coordenadas de doble precisión en el orden X,Y.

#### Point

```
{
    Double X    // coordenada X
    Double Y    // coordenada Y
}
```

Posición	Campo	Valor	Tipo	Numero	Byte Order
Byte 0	Tipo de Shape	1	Integer	1	Little
Byte 4	X	X	Double	1	Little
Byte 12	Y	Y	Double	1	Little

**TABLA 4.** Contenido del Registro de Puntos

### B.3.7.3 Shapefiles de polilíneas.

Una polilínea es un conjunto ordenado de vértices que consiste de una o más partes. Una parte es una secuencia conectada de dos o más puntos. Las partes pueden llegar a estar o no conectadas entre ellas. Las partes pueden o no intersectarse unas con otras. Debido a que esta especificación no toma en cuenta el hecho de que existan puntos consecutivos con coordenadas idénticas, es responsabilidad del programa que lee los datos tomar en cuenta estos casos. El caso de una parte que no tiene puntos no es tomado en cuenta.

**Polilyne**

```

{
    Double[4]    Box           // rectángulo delimitador
    Integer     NumParts     // Numero de partes
    Integer     NumPoints    // Numero total de puntos
    Integer[NumParts] Parts  // Indice del primer punto en cada parte
    Point[NumPoints] Points  // Puntos para todas las partes
}

```

Posición	Campo	Valor	Tipo	Numero	Byte Order
Byte 0	Tipo de Shape	3	Integer	1	Little
Byte 4	Rectángulo	Rectángulo	Double	4	Little
Byte 36	NumPartes	NumPartes	Integer	1	Little
Byte 40	NumPuntos	NumPuntos	Integer	1	Little
Byte 44	Partes	Partes	Integer	NumPartes	Little
Byte X	Puntos	Puntos	Point	NumPuntos	Little

**TABLA 5.** Contenido del registro de Polilinea

A continuación describimos los campos para una polilinea.

- **Box:** es el rectángulo delimitador para la polilinea almacenado en el orden  $X_{min}$ ,  $Y_{min}$ ,  $X_{max}$ ,  $Y_{max}$ .
- **NumParts:** es el numero de partes de la polilinea.
- **NumPoints:** numero total de puntos para todas las partes.
- **Parts:** es un arreglo de tamaño  $NumParts$ . Almacena para cada polilinea el índice de su primer punto en el arreglo de puntos. Los índices del arreglo comienzan en cero.
- **Points:** es un arreglo de tamaño  $NumPoints$ . Los puntos para cada parte en la polilinea se almacenan de principio 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 de comienzo para cada parte. Dentro del arreglo de puntos no hay delimitador para las partes.*

## **B.4 Ventajas y Desventajas de los Shapefiles**

ESRI Inc. Introdujo es formato shapefile para proveer un entendimiento simple y efectivo de la información geoespacialmente diseminada. Hoy en día el Shapefile es el candidato a standard de facto para el intercambio de datos geoespaciales y aplicaciones GIS de escritorio.

El formato de estos archivos se encuentra totalmente publicado y es de libre difusión brindando de esta forma un formato de intercambio de datos geoespaciales no-propietario que beneficia algunas aplicaciones GIS, pero puede limitar a otras.

### **B.4.1 Puntos Favorables de los Shapefiles**

El Shapefile efectivamente implementa el intercambio de datos espaciales. Por otra parte, es un standard publicado y abierto que esta disponible gratuitamente a los desarrolladores de software GIS.

El Shapefile es lo que su nombre indica; un archivo que almacena figuras. Realmente no es solo un archivo, sino un set de tres o más archivos. El primer archivo es para las figuras, un segundo archivo es para el índice de los punteros a las figuras, y un tercero es para los atributos asociados.

Los ShapeFiles pueden ser puntos, líneas o polígonos, y los figuras del mismo tipo son almacenados en el mismo archivo. Debido a que cada polígono y línea del ShapeFile tienen un numero variable de segmentos de línea, la información de inicio de cada archivo es mantenida en el archivo de índices.

El atributo del tercer archivo es asociado con cada figura respecto a su posición correspondiente en el shapefile. La primer figura es asociada con el primer registro de atributos, la segunda con el segundo registro, y así siguiendo.

Los shapefiles proveen un formato común para almacenar, transferir y mostrar datos geoespaciales para productos GIS de escritorio tales como el Arcview de ESRI y Arc Explorer que es un visualizador freeware. En contraste a los históricos formatos propietarios, el shapefile es una especificación publicada abiertamente. Otros software GIS o desarrolladores de aplicaciones pueden usar la especificación para desarrollar herramientas que puedan leer o escribir shapefile.

Además el uso de Shapefiles por la comunidad GIS se esta convirtiendo en el estandar de facto usado por los vendedores de software GIS para importar y/o exportar datos geoespaciales.

#### **B.4.2 Puntos Desfavorables de los Shapefiles**

Sin embargo, el shapefile no logra todas las expectativas previstas para la funcionalidad de procesamiento de los GIS tradicionales en términos de rendering cartográfico, atributos relacionales, topología y extensibilidad.

##### **- No Rendering**

Cuando los Shapefiles se muestran en ArcView o ArcExplorer, se pueden usar símbolos coloreados para representar puntos, estilos de líneas coloreados para representar líneas, y patrones de color de relleno para los polígonos. Esta capacidad de rendering cartográfico es fundamental para crear y mostrar datos geoespaciales como un mapa. Cuando los usuarios intentan pasar el mapa y sus leyendas asociadas a alguien que solo usa formato Shapefile, las asignaciones del rendering cartográfico no son incluidas como parte de la definición estándar del Shapefile. A menos que los usuarios establezcan su propia convención para asignar rendering cartográfico como atributos del shape, estos no van a ser dejados ocultos. El resultado: Lo que ve se no es lo que se tiene.

## - Manejo de Base de Datos No Relacional

La relación uno-a-uno entre un shape y sus registros de atributos no son efectivos para representar el multinivel, relaciones uno-a-muchos que usan los datasets geospaciales para recursos naturales, manejo de facilidades, planeamiento y otras aplicaciones GIS. Si un Shapefile fuera asociado con muchos registros de atributos, intentar representar relaciones de datos complejas en una simple tabla no normalizada resultaría en un archivo plano extremadamente grande que sería imposible de manipular y procesar. Esta es la causa por la que la mayoría de los paquetes de software GIS que tienen un conjunto completo de funciones usan bases de datos relacionales para representar relaciones de atributos complejas.

## - No Topológico

No tiene importancia la forma en que los arcos, nodos y pseudotopologías son aplicados a Shapefiles. Cada polígono es una figura completamente limitada por segmentos de líneas. Cuando dos polígonos comparten el mismo lado, estos segmentos de líneas son duplicados. Cuando un polígono es editado o movido, no hay nada implícito en el formato Shapefile que asegure que los segmentos de líneas duplicados para los polígonos adyacentes con lados compartidos sean editados o movidos también. De hecho sin un especial cuidado y manejo adecuado, estos segmentos de líneas duplicadas quedarán desenganchadas, dejando de retener las características de lados compartidos del polígono alterado.

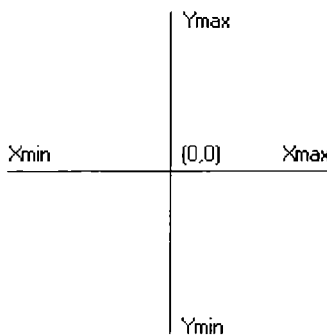
El problema se complica cuando los Shapefiles se intersectan o mezclan con otros Shapefiles durante análisis de superposición booleano, como una leve variación en la posición de los arcos y nodos puede crear “myriad sliver”, esto es, polígonos que son virtualmente indetectables durante inspecciones visuales trabajando con escalas de mapas, pero afectando seriamente el tiempo de procesamiento y exactitud durante cálculos espaciales.

### **- Extensibilidad limitada**

La simplicidad de los formato Shapefile deja poco lugar para crear un formato de archivo flexible y auto descriptivo. El intercambio de datos estándar intenta ser lo más flexible y abierto posible maximizando la interoperabilidad y asegurando adaptarse a requerimientos futuros. Las transferencias de datos auto descriptivas son extensibles porque ellas están preparadas para cambios, permitiendo nuevas descripciones de sus estructuras internas. Esto es particularmente real en la falta de soporte para transferir metadatos que describen los datos geoespaciales y sus características.

# **Apéndice C**

## **Espacio de Coordenadas y Transformaciones.**



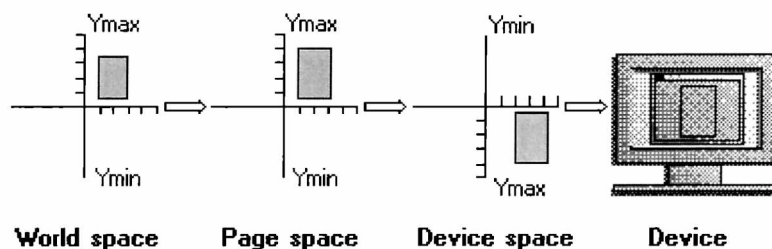
Windows soporta cuatro espacios de coordenadas, los cuales se describen en la siguiente tabla:

Espacio de coordenadas	Descripción
<b>World</b>	Se utiliza opcionalmente como el espacio de coordenadas inicial para las transformaciones gráficas. Permite escalado, rotación, traslación, deformación y reflexión. El espacio de coordenadas world tiene un tamaño de $2^{32}$ unidades de alto por $2^{32}$ unidades de ancho.
<b>Página</b>	Se utiliza como el siguiente espacio luego del world. En este espacio se setea el modo de mapeo. El espacio de página tiene las mismas dimensiones que el world.
<b>Dispositivo</b>	Utilizado después del espacio de página. Solamente permite traslación, lo cual asegura que el origen del espacio de dispositivo mapea en una ubicación apropiada dentro del espacio de página. Mide $2^{27}$ por $2^{27}$ .
<b>Físico</b>	El espacio final para las transformaciones gráficas. Usualmente se refiere al área de la ventana del cliente en la aplicación; sin embargo puede incluir también el escritorio completo, una ventana completa (incluyendo bordes y menús) o una página de una impresora o un plotter. Las dimensiones físicas varían dependiendo de las dimensiones del monitor, impresora o plotter.

El *espacio de página* trabaja conjuntamente con el *espacio de dispositivo* para proveer a las aplicaciones unidades de medida independientes del dispositivo, tales como milímetros o pulgadas.

Para mostrar la salida en un dispositivo físico (monitor por ejemplo), Windows copia (o mapea) una región rectangular de un espacio de coordenadas en el siguiente utilizando una transformación, hasta que la salida aparece enteramente en el dispositivo físico. El mapeo comienza a nivel del *espacio world* (o *espacio del mundo*) si la aplicación llama la función API de Windows **SetWorldTransform**; de otra manera, el mapeo comienza en el espacio de página. A medida que el sistema mapea cada punto dentro de la región rectangular de un espacio a otro, aplica un algoritmo denominado transformación.

Aunque una transformación afecta un objeto como un todo, se aplica a cada punto y a cada línea del objeto.



### C.1.1.1 Transformaciones del espacio world al espacio de página

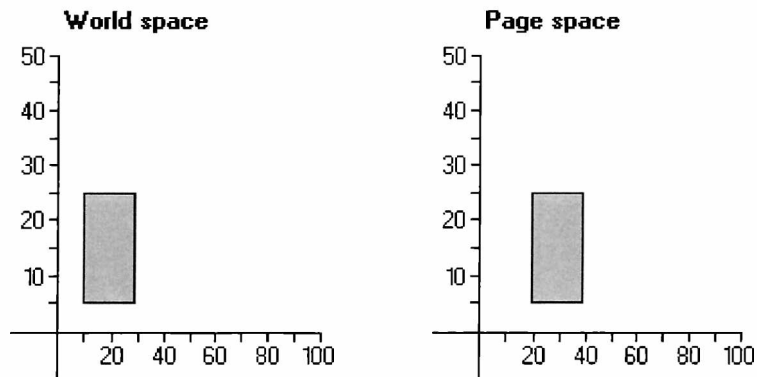
Las transformaciones del espacio world en el espacio de página soportan traslación y escalado. Además, soportan rotación, deformación y reflexión.

#### C.1.1.1.1 Traslación

Algunas aplicaciones trasladan objetos dibujados en el área del cliente ejecutando la función **SetWorldTransform** para setear la transformación apropiada entre el espacio world y el espacio de página. La función **SetWorldTransform** recibe un puntero a una estructura **XFORM** que contiene los valores adecuados. Los miembros **eDx** y **eDy** de la

estructura **XFORM** especifican las componentes de traslación horizontal y vertical respectivamente.

Cuando ocurre la traslación, cada punto en un objeto se corre verticalmente, horizontalmente o ambos en una cantidad especificada. La siguiente figura muestra un rectángulo de 20x20 unidades que se traslada a la derecha en 10 unidades.

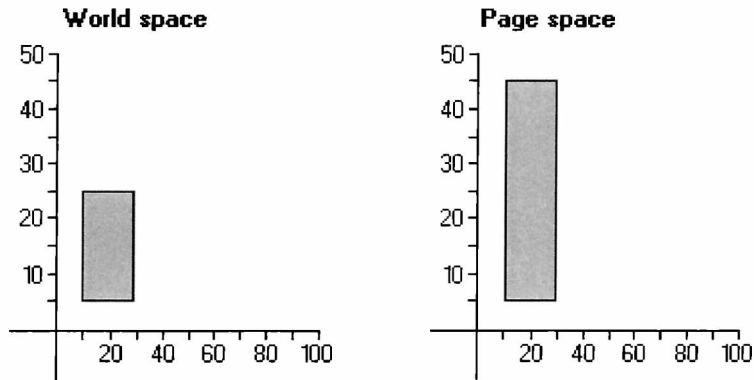


#### C.1.1.1.2 Scaling

Muchas aplicaciones CAD y de dibujo proveen herramientas que permiten escalar la salida creada por el usuario. Las aplicaciones que proveen scaling (o zoom) llaman a la función **SetWorldTransform** para setear la apropiada transformación entre el espacio world y el espacio de página. Esta función recibe un puntero a una estructura **XFORM** (idem anterior) en cuyos componentes **eM11** y **eM22** se especifican los factores de scaling horizontal y vertical respectivamente.

Cuando ocurre el scaling, las líneas horizontales y verticales que constituyen el gráfico son estiradas y/o comprimidas con respecto a los ejes x e y. La siguiente ilustración muestra un rectángulo de 20x20 unidades escalado verticalmente al doble de su altura original cuando se copia del espacio world al espacio de página.

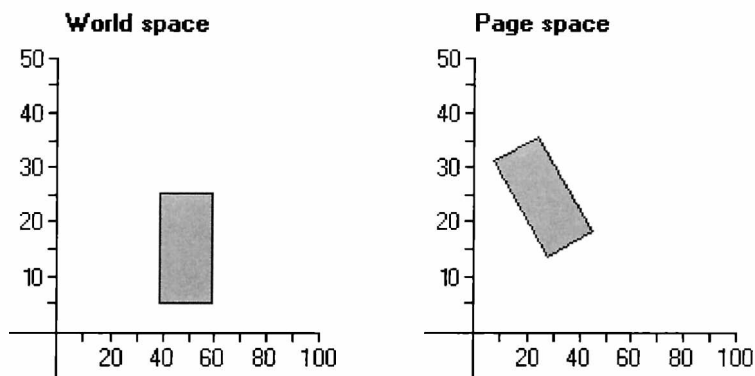




### C.1.1.1.3 Rotación

Las aplicaciones CAD también proveen herramientas para rotar objetos dibujados en el área del cliente. Las aplicaciones que incluyen capacidades de rotación invocan a la función **SetWorldTransform** (idem casos anteriores). Los componentes **eM11**, **eM12**, **eM21** y **eM22** de la estructura **XFORM** especifican el seno, coseno, seno negativo y coseno negativo de la rotación respectivamente.

Cuando ocurre la rotación, los puntos que constituyen el objeto se rotan con respecto al sistema origen de coordenadas. La siguiente figura muestra un rectángulo de 20x20 unidades rotado 30 grados.



### **C.1.1.2 Transformaciones combinadas del espacio world al espacio de página**

Las tres transformaciones anteriores pueden combinarse en una única matriz de 3x3 elementos. La función **CombineTransform** puede utilizarse para combinar dos transformaciones del espacio world al espacio de página.

# Bibliografía

1. Aronoff S.: *“Geografic Information Systems: A Managment Perspective”* WDL Publications. Ottawa. Canada. 1989.
2. Moldes Javier F. : *“Tecnología de los Sistemas de Información Geográfica”* , RA-MA Editorial. Madrid. España. 1995.
3. Barredo José : *“Sistemas de Información Geográfica y Evaluación Multicriterio en la Ordenación del Territorio”* , RA-MA Editorial. Madrid. España. 1996.
4. ESRI: *Tutorial Multimedial. ArcView 3.0* . ESRI. Redlands. EE.UU.1999.
5. Maguire David, Goodchild Michael, Rhind David: *Geographical Information System Volume I: Principles*. Logman Scientific & Technical. Londres. Gran Bretaña. 1991.
6. Montgomery Glenn, Schuch Harold. *GIS Data Conversion Handbook*. GIS World Books EE.UU. 1993
7. GIS World: *“The World Leading Integrated Spatial Technologies”* Publication Mayo 1998 VOL II No 5.
8. Autodesk: *“AutoCAD Map Release 3 User Guide”*. EE.UU. 1998.
9. ESRI White Paper Series. *Automation of Map. Generalization - the Cutting-Edge Technology.*. Mayo 1996.
10. ESRI White Paper Series. *GIS Approach to Digital Spatial Libraries*. Mayo 1994.
11. Douglas, David H. and Thomas K. Peucker. *Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature*, The Canadian Cartographer, Vol. 10, No. 2. 1973.
12. Visvalingam, M. and J.D. Whyatt. *The Douglas-Peucker Algorithm for Line Simplification: Re-evaluation through Visualization*. Computer Graphics Forum 9. 1990.
13. Goodchild Michael, *Error Propagation in Environmental Modelling with GIS*, Taylor & Francis. Abril 1998.
14. Andy Mitchell, *The ESRI Guide to Gis Analysis Volume I: Geographics Patterns & Relationships*, Julio 1999.

15. <http://www.esri.com>  
Sitio Oficial de ESRI.
16. <http://blaze.innovativegis.com/education/primer/nature.html>.  
The GIS Primer. An Introduction to Geographic Information Systems.
17. <http://pasture.ecn.purdue.edu/~tgis/cases.html>  
GIS @ Purdue. Case Studies. Learning by Example.
18. <http://mathworld.wolfram.com/Line-LineIntersection.html>  
Line to line intersection theory
19. <http://msdn.microsoft.com>  
MSDN library, Platform SDK, Windows GDI, Coordinate Spaces and Transformations
20. [http://www.gisca.adelaide.edu.au/~bbryan/lectures/analysis\\_vec](http://www.gisca.adelaide.edu.au/~bbryan/lectures/analysis_vec)  
Spatial Analysis with Vector GIS